Scientific Journal of Impact Factor (SJIF): 5.71

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 6, Issue 10, October -2019

PYTHON CURRENT TREND APPLICATIONS- AN OVERVIEW

POPULAR WEB DEVELOPMENT FRAMEWORKS IN PYTHON

A.L.Sayeth Saabith [1], MMM.Fareez [2], T.Vinothraj[3]

[1]*Centre for Information Communication Technology*
Faculty of Science, Eastern University, Vantharoomulai, Sri Lanka

[2]*Finance Department*
Eastern University, Vantharoomulai, Sri Lanka

[3]*Centre for Information Communication Technology*
Faculty of Science, Eastern University, Vantharoomulai, Sri Lanka

Abstract —*Python is a powerful high-level, interpreted, interactive, and object-oriented scripting language created by Guido Van Rossum in late 1980's. Python is a very suitable language for the beginner level programmers and supports the development of a wide range of applications from simple text processing to www browsers to games developments. One of the biggest reasons for Python's rapid growth is the simplicity of its syntax. The language reads almost like plain English, making it easy to write complex programs. In this paper we first analyze you to Python programming language popularity and features. Moreover, this paper specifying applications areas where python can be applied and specially analyzing web application frameworks which are using in Python programming language.*

Keywords-*Python, frameworks, object-oriented scripting language, web application*

I. INTRODUCTION

Programming languages have been around for ages, and every decade sees the launch of a new language sweeping developers off their feet. Python is considered as one of the most popular and in-demand programming language. A recent Stack Overflow survey showed that Python has taken over languages such as Java, C, C++ and has made its way to the top [8]. The following reasons that are the Python programming language is becoming so popular [1,2].

- **Simple and Easy to learn:** Python which is extremely simple and easy so the python easy to read and easy to learn, since it is closely resembles in English language.
- **Supportive community:** Python has been around for three decades, which has been plenty of time for a developed, supportive community to grow up around the language. From official documentation to YouTube tutorials, Python learners of all ages and skill levels can find the support they need to improve their knowledge of the language.
- **Web Development:** Web programming with python provides a lot of choices as python has an array of frameworks for developing website. There are so many frameworks available in Python such as Django, flask, pylon and so on. Python is majorly used in the web development.
- **Use in big data and machine learning:** Big data and machine learning are two of the hottest trends in computer science right now, helping enterprises transform their workflows and processes. Python is the language in which much of this research and development takes place. As the second most popular tool for analytics and data science, Python powers countless data processing workloads in organizations around the world. Meanwhile, Python libraries such as OpenCV for computer vision and TensorFlow for neural networks are used in thousands of machine learning projects every day.
- **Efficiency:** Python represents a different programming paradigm than older languages such as Java and C++. However, this "Pythonic" way of doing things often permits developers to get more done with less work–often in just a few lines of code. What's more, the versatility of Python allows you to use the language across a variety of environments, from web development and mobile development to desktop applications and hardware programming.

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Small Talk, and Unix shell and other scripting languages. Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL). Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress. Python 1.0 was released in November 1994. In 2000, Python 2.0 was released. Python 2.7.11 is the latest edition of Python 2. Meanwhile, Python 3.0 was released in 2008. Python 3 is not backward compatible with Python 2. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules so that "There should be one -- and preferably only one -- obvious way to do it." Python 3.7.4 is the latest version of Python 3[5]. Tech Giants like Cisco, IBM, Mozilla, Google, Quora, Hewlett-Packard, Dropbox, and Qualcomm are using this language owing to its simplicity

and elegance. Most developers prefer Python over the plethora of programming languages out there because of its emphasis on readability and efficiency.

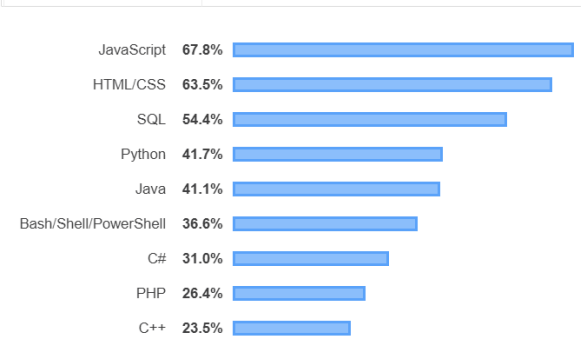

Figure 1:Most Popular Technology [8]

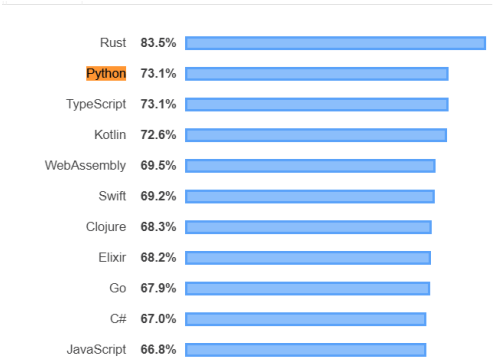

Figure 2: Most wanted language [8]

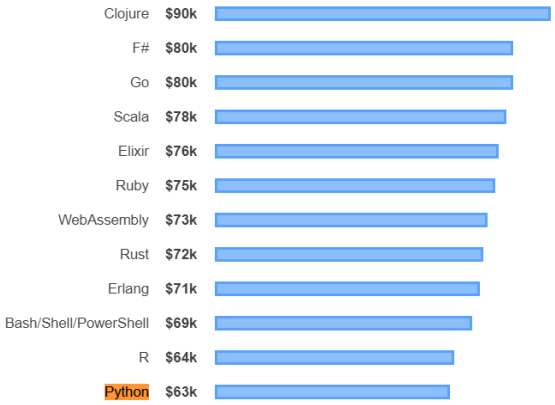

Figure 3: Highest Salary in worldwide [8]

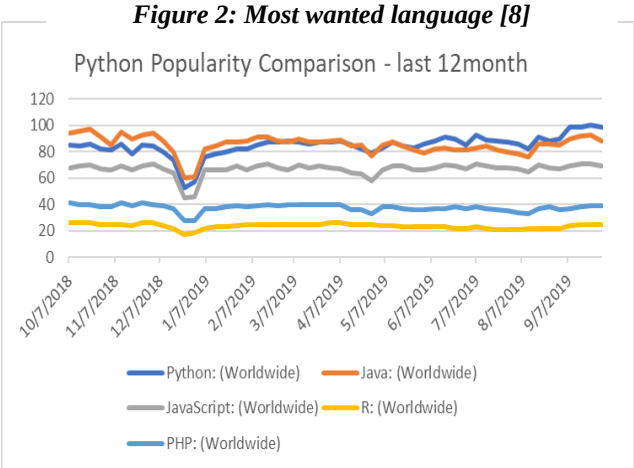

Figure 4: Python Popularity Comparison [9]

II. PYTHON PROGRAMMING FEATURES

- **Simple:** Compared to many other programming languages, coding in Python is like writing simple strict English sentences. In fact, one of its oft-touted strengths is how Python code appears like pseudo-code. It allows us to concentrate on the solution to the problem rather than the language itself.
- **Easy to Learn:** As we will see, Python has a gentler learning curve (compared to languages like C, Java, etc.) due to its simple syntax. Free and Open Source: Python and the majority of supporting libraries available are open source and generally come with flexible and open licenses. It is an example of a FLOSS (Free/Libré and Open Source Software). In layman terms, we can freely distribute copies of open source software, access its source code, make changes to it, and use it in new free programs.
- **High-level:** Python is a programming language with strong abstraction from the details of the underlying platform or the machine. In contrast to low-level programming languages, it uses natural language elements, is easier to use, automates significant areas of computing systems such as resource allocation. This simplifies the development process when compared to a lower-level language. When we write programs in Python, we never need to bother about the lower-level details such as managing the memory used by programs we write, etc.
- **Dynamically Typed:** Types of variables, objects, etc. in Python are generally inferred during runtime and not statically assigned/declared as in most of the other compiled languages such as C or Fortran.

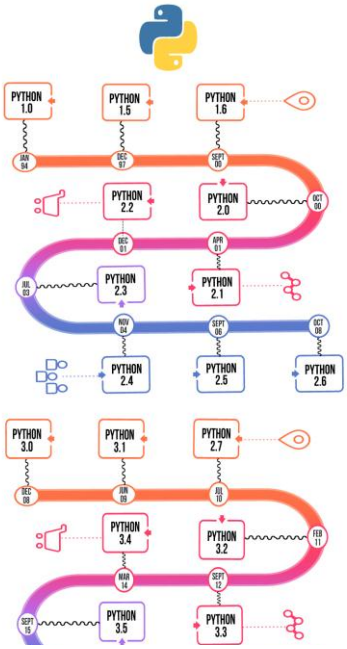

Figure 5: Python version history [5]

- **Portable/Platform Independent/Cross Platform:** Being open source and also with support across multiple platforms, Python can be ported to Windows, Linux and Mac OS. All Python programs can work on any of these

platforms without requiring any changes at all if we are careful in avoiding any platform-specific dependency. It is used in the running of powerful severs and also small devices like the Raspberry Pi3. In addition to the above-mentioned platforms, following are some of the other platforms where Python can be used

- FreeBSD OS
- Oracle Solaris OS
- AROS Research OS
- QNX OS
- BeOS
- z/OS
- VxWorks OS
- RISC OS

- **Interpreted:** A programming language can be broadly classified into two types viz. compiled or interpreted. A program written in a compiled language like C or C++ requires the code to be converted from the original language (C, C++, etc.) to a machine-readable language (like binary code i.e. 0 and 1) that is understood by a computer using a compiler with various flags and options. This compiled program is then fed to a computer memory to run it. Python, on other hand, does not require compilation to machine language. We directly run the program from the source code. Internally, Python converts the source code into an intermediate form known as byte code and then translates this into the native language of the underlying machine. We need not worry about proper linking and the loading into memory. This also enables Python to be much more portable, since we can run the same program onto another platform and it works just fine! The 'CPython' implementation is an interpreter of the language that translates Python code at runtime to executable byte code.
- **Multiparadigm:** Python supports various programming and implementation paradigms, such as Object Oriented, Functional, or Procedural programming.
- **Extensible:** If we need some piece of code to run fast, we can write that part of the code in C or C++ and then use it via our Python program. Conversely, Python code can embed in a C/C++ program to give it scripting capabilities.
- **Extensive Libraries:** The Python Standard Library [6] is huge and, it offers a wide range of facilities. It contains built-in modules written in C that provides access to system functionality such as I/O operations as well as modules written in Python that provide standardized solutions for many problems that occur in everyday programming. Some of these modules are listed below
  - Text Processing Modules
  - Data Types
  - Numeric and Mathematical Modules
  - Files and Directory Modules
  - Cryptographic Modules
  - Generic Operating System Modules
  - Networking Modules
  - Internet Protocols and Support Modules
  - Multimedia Services
  - Graphical User Interfaces with Tk
  - Debugging and Profiling
  - Software Development, Packaging and Distribution

  In addition to the Python Standard Library, we have various other third-party libraries which can be accessed from Python Package Index [7].
- **Garbage Collection:** Python takes care of memory allocation and deallocation on its own. In other words, a programmer does not have to manage memory allocation and need not have to reallocate and deallocate memory before constructing variables and objects. Additionally, Python provides Garbage Collector [6] interface to handle garbage collection.

III. CURRENT TREND APPLICATION IN PYTHON

Python is used by the novice programmer as well as by the highly skilled professional developer. It is being used in academia, at web companies, in large corporations and financial institutions. Python is known for its general-purpose nature that makes it applicable in almost each domain of software development. Python as a whole can be used in any sphere of development [1,2,3].

- **Python Applications in Web Frameworks and Applications:** Python has been used to create a variety of web-frameworks including CherryPy, Django, TurboGears, Bottle, Flask etc. These frameworks provide standard libraries and modules which simplify tasks related to content management, interaction with database and interfacing with different internet protocols such as HTTP, SMTP, XML-RPC, FTP and POP. Plone, a content management system; ERP5, an open source ERP which is used in aerospace, apparel and banking; *Odoo* – a consolidated suite of business applications; and Google App engine are a few of the popular web applications based on Python.

- **Python Application in GUI-Based Desktop:** Python has simple syntax, modular architecture, rich text processing tools and the ability to work on multiple operating systems which make it a desirable choice for developing desktop-based applications. There are various GUI toolkits like wxPython, PyQt or PyGtk available which help developers create highly functional Graphical User Interface (GUI).
- **Python Application in Image Processing and Graphic Design:** Python has been used to make 2D imaging software such as Inkscape, GIMP, Paint Shop Pro and Scribus. Further, 3D animation packages, like Blender, 3ds Max, Cinema 4D, Houdini, Lightwave and Maya, also use Python in variable proportions.
- **Python Application in Scientific and Computational:** The higher speeds, productivity and availability of tools, such as Scientific Python and Numeric Python, have resulted in Python becoming an integral part of applications involved in computation and processing of scientific data. 3D modeling software, such as FreeCAD, and finite element method software, such as Abaqus, are coded in Python.
- **Python Application in Games and 3D graphics:** Python has various modules, libraries and platforms that support development of games. For example, PySoy is a 3D game engine supporting Python 3, and PyGame provides functionality and a library for game development. There have been numerous games built using Python including Civilization-IV, Disney's Toontown Online, Vega Strike etc.
- **Python Application in Software Development:** Python is often used as a support language for software developers, for build control and management, testing, and in many other ways. For instances, SCons for build control. Buildbot and Apache Gump for automated continuous compilation and testing. Roundup or Trac for bug tracking and project management.
- **Python Applications in Business:** Python is also a great choice to develop ERP and e-commerce systems such as Tryton – A three-tier, high-level general-purpose application platform, Odoo – A management software with a range of business applications. With that, it's an all-rounder and forms a complete suite of enterprise-management applications in-effect.
- **Python Applications in Language Development:** Python's design and module architecture has influenced development of numerous languages. Boo language uses an object model, syntax and indentation, similar to Python. Further, syntax of languages like Apple's Swift, CoffeeScript, Cobra, and OCaml all share similarity with Python.

IV. Python Web Development Frameworks

Frameworks are the collection of packages and modules that allows us to create web applications very easily without having to handle low-level activities like thread management or process management, and protocols management etc... With the help of the frameworks, we can build our application very effectively, easily, within a short span. Frameworks are like a gift for the developers because it makes the developer's life a lot easier and happier.

Since python is also used by the developer's quite a lot in the web development industry. because of its readability, maintainability in a convenient manner. when we are talking about web development we should need to focus on both front-end and back-end part of an application. In a web development environment, the backend framework consists of languages and certain tools used in server-side programming. python is using ridiculous while developing the backend part of an application. In python, for web development, there are three types of frameworks are available shown in following table.

Full Stack Frameworks	Non-Full Stack Frameworks	Asynchronous Framework
Full-Stack Frameworks are ones that help us with the full development stack from the user interface till the data store. It gives full support to developers including basic components like form generators, form validation, and template layouts.	Non-full stack frameworks are also called as Micro frameworks because it doesn't have many components like full stack frameworks. These are lightweight frameworks that don't offer additional functionalities and features, such as database abstraction layer, form validation, and specific tools and libraries. Developers using a microframework need to add a lot of code and additional requirements manually.	Gaining popularity recently, any asynchronous framework is a microframework that allows for handling a large set of concurrent connections. Typically, an asynchronous framework built for Python uses the programming language's async IO library.

4.1 POPULAR WEB DEVELOPMENT FRAMEWORKS ANALYSIS

Name	Key Features	URL	Type of Framework
Django	extremely fast Fully Loaded assure secure excessively scalable Incredibly versatile	https://www.djangoproject.com/	Full
Web2Py	Easy Setup Cross-Platform Database Connectivity Security	http://www.web2py.com/	Full
Turbo Gears	Integration with the MochiKit Javascript library. Command-line tools. It supports SQLObject and SQLAlchemy. It supports multiple databases.	https://www.turbogears.org/	Full
CubicWeb	It has good security workflow. It also supports for Resource Description Framework (RDF) and Web Ontology Language (OWL). It has Relational Query Language (RQL) to simplify the queries related to data.	https://www.cubicweb.org/	Full
Pyramid	Single-file applications URL generation Extensible configuration All-embracing templating and asset specifications Flexible authentication and authorization	https://trypyramid.com/	Full
Giotto	Automatic URL routing Database persistence with SQLAlchemy Extremely concise code Functional CRUD patterns Generic models and views Inbuilt cache with support for Memcache and Redis (Available API for extending support for other engines) Jinja2 for HTML templates (API available for supporting other template engines) Multiple pluggable controllers	https://pypi.org/project/giotto/	Full
Flask	Lightweight framework. Compatible with Google App Engine. Built-in development server and debugger.	https://palletsprojects.com/p/flask/	Micro
CherryPy	It has a powerful configuration and flexible plugin system. It can easily run on multiple HTTP servers at once. It has Built-in tools for caching, encoding, sessions, authentication, static content, and many more.	https://cherrypy.org/	Micro
Bottle	It allows users to access form data, file uploads, cookies, and other HTTP-related metadata in a much simpler way. It has a built-in HTTP server.	http://bottlepy.org/docs/dev/index.html	Micro
Tornado	Non-blocking HTTP client. It has a high-quality performance. It allows us to implement third-party	http://www.tornadoweb.org/en/latest/	Micro

	authentication and authorization schemes i.e (Google, Facebook, Twitter)		
Sanic	It runs on Python 3.5+. Sanic supports asynchronous request handlers, which makes it compatible with Python 3.5's async/await functions. This enhances its speed, offering non-blocking capabilities.	https://sanic.readthedocs.io/en/latest/	Micro
Falcon	An extensible, highly-optimized code base DRY request processing through middleware components and hooks Ease of access for headers and bodies via request and response classes Extra speed boost with Cython support Idiomatic HTTP error responses REST-inspired resource classes and URI templates offer intuitive routing Unit testing via WSGI helpers and mocks Upfront exception handling	https://falconframework.org/	Micro
Dash	Dash apps demand very little boilerplate code for getting started Error handling (Dash Deployment Server) High-degree of customization LDAP integration (Dash Deployment Server) Plugin support Simple interface for tying UI controls, including dropdowns, graphs, and sliders URL routing (Dash Deployment Server)	https://plot.ly/dash/	Micro
Growler	Easy to see program flow due to lack of required callbacks and proper try/except blocks Support for a multitude of open-source packages Use decorators for writing clean, reusable code Ziapp module allows zipping an entire application into a single executable file	http://www.growler.rocks/	Micro
AIOHTTP	Allows effectively building the views Middlewares support Pluggable routing Signals Supports both Client WebSockets and Server WebSockets without the Callback Hell	https://aiohttp.readthedocs.io/en/stable/	Asynchronous

V. CONCLUSION

In this paper, we discussed the python programming as a more suitable choice for beginners and professional expert. This paper elaborated the why this python more popular in this real-world era, features of this programming language such as fast, easy, powerful, portable, simple, and free open source language that supports other technologies, and discussed the applicable various domains of the python. This study has also specially analyzed python web development frameworks in three different types which are Full-stack, Micro, and Asynchronous.

REFERENCES

[1] K. R. Srinath, "Python – The Fastest Growing Programming Language," International Research Journal of Engineering and Technology (IRJET), vol. 4, Issue 12 pp. 354–357, December 2017.
[2] Kalyani Adawadkar, "Python Programming-Applications and Future," International Journal of Advance Engineering and Research Development(IJAERD), Special Issue SIEICON-2017, pp. 1–4, April -2017.
[3] QuantInsti, Python Basics With Illustrations from the Financial Markets, QuantInsti Quantitative Learning Pvt. Ltd.
[4] Pinky Sodhi, Naman Awasthi, Vishal Sharma, "Introduction to Machine Learning and Its Basic Application in Python", Proceedings of 10th International Conference on Digital Strategies for Organizational Success, January, 2019
[5] https://www.geeksforgeeks.org/history-of-python/
[6] https://www.python.org/doc/essays/blurb/
[7] https://docs.python.org/3/library/
[8] https://insights.stackoverflow.com/survey/2019
[9] https://trends.google.com/trends/

