PYTHON LANGUAGE: JEWEL OF COMPUTER PROGRAMMING

*Guneet Singh, **Aarti Singh

*Research Scholar, VIT Bhopal, M.P., India

**Asstt. Prof., Guru Nanak Girls College, Yamuna Nagar, Haryana, India

Abstract

Coding is the backbone of software industry, which in turn depends on programming languages. There are wide variety of programming languages with different features and capabilities. This work focuses on features and capabilities of Python programming language that make it a versatile language grabbing strong roots in software world these days. Budding programmers often get confused in choosing a language to gain proficiency in. This work compares features of Python with other existing prominent languages like C, C++ andJava.

Key words ; Python programming, High level languages, Web development, Object oriented programming languages

1. Introduction

Python is a general-purpose programming language [1] which falls into the 4thgeneration of programming languages[13]; thus, is a high-level programming language. With quite an elegant design philosophy it puts emphasis on code readability and a reduced number for lines of code as compared to other languages such as C, C++ and Java. It supports a wide array of programming styles, its four main paradigms are imperative, functional, procedural, and object-oriented. The language also allows programs to be written clearly on a small scale as well as on a larger scale. Because of the powerful tools that reflect the way programmers tend to think and the way they implement an algorithm; this language is not only good for teaching but also for real-world applications. Python is finding applicability in artificial intelligence-based applications as well as in web development because of its simplicity and codereusability.

Next section explores literature to find relevant works on Python.

2. Literature Review

Although Python is much popular among programmers, not much literature is available in the form of research papers,that is quite obvious since programming language is for learning and applying. Sharma et. Al [3] highlighted Python as programming language for the future, they illustrated various appealing features of Python. Karmore and Girhe in [14] emphasized that Pythonis used in vast number of applications due to the variousstandard libraries that come along with it and its capacity to integrate with other languages and use their features.K.R.Srinath in [2] illustratedthe reasons behind python being credited as the fastest growing programming language in the recent times.
This work contributes by illustrating features of Python with coding examples. Also this work compares three most popular programming languages i.e. C, JAVA and Python over most desired features.
Next section throws light on appealing features of Python that make it suitable for budding programmers.

3. Python: best choice for noviceprogrammers

When a person starts programming, it is always beneficial to choose a language capable of providing versatile features and wider applicability. Python is a language incorporating both these merits. This section outlines features [2] of this language which makes it first choice for naïve programmers:

- ***A reduced amount of language specifics andstrictness:***

The important thing for a person just learning to code is to understand the concept and not to focus on the language specifics such as always declaring variables in the beginning of a C program or memory management in C++ both of which cannot be avoided. This also puts the thinking of a programmer in a box because while programming the programmer has to think of the solution to the given problem at the same time, not violating any syntax rules or placement rules and other things specific to that language. However, Python promotes the flow in a person's thinking process; the language is quite flexible in a lot of ways.

Let us elaborate this feature with an example, in C if a programmer has written a program of 200 lines of code and now wants to add a new variable to the mentioned program, the programmer would have to go all the way at the top just to declare the variable along with its data type along with assigning the variable a value and then has to come all the way down to continue coding. While a person proficient in the language would not have any trouble doing so, one must look at this statement from the perspective of a naïve programmer who does not have that good grasp on the syntax as well as the on concepts of coding. Such a person would be puzzled and after some time, irritated. Python happens to get rid of this problem entirely. If the programmer is put in the exact same circumstances except C being the language of coding is swapped with Python in this scenario, the results would be contrasting. Since, Python allows the programmer to declare the variable anywhere in the program, the programmer now, does not have to go all the way to the top of the code just to declare a variable, the interpreter takes care of the rest. Moreover, the programmer does not even have to declare the variable's data type. Which makes it a lot less cumbersome than it would have been in C. The syntax of declaring a variable in both of languages is givenbelow:

Figure 1. Syntax for declaring a variable in C

```
variable_name = [value];
```

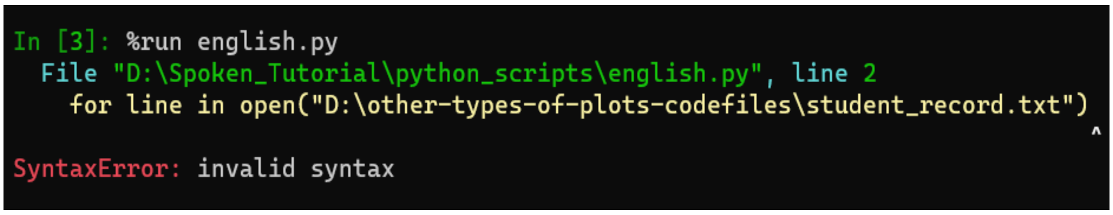
Figure 2. Syntax for declaring a variable in Python

```
[storage_class] data_typevariable_name = [value];
```

- ***Easier to debug aprogram:***

There can only be one thing more heart breaking than writing 2000 lines of code spending a significant amount of time, all to find an error in the program and that is not knowing exactly where the error has occurred; thus, having to sift through those 2000 lines of code looking for that problem. Although not at this big scale, this problem is faced by almost every programmer. Now, one might suggest the idea of using a debugging console. While the idea has nothing wrong with it, it does not work for the most part in real life, the problem being that not all languages have a user-friendly debugging console. Python solves this problem by clearly stating the line of occurrence of error as well as pointing its location using the '^' sign as a pointer in the line below it. After this the interpreter clearly mentions the kind of error that has occurred in the program. A visual representation of this point can be found below:

Figure 3. Error Diagnosis in Python



```
In [3]: %run english.py
  File "D:\Spoken_Tutorial\python_scripts\english.py", line 2
    for line in open("D:\other-types-of-plots-codefiles\student_record.txt")
                                                                            ^
SyntaxError: invalid syntax
```

In this case, the line of code lacks the use of a colon sign (:) at the end

It is much easier to locate error in Python than in C, where editor sometimes points to line below occurrence of error.

- ***Names of functions are very simple tounderstand:***

From the perspective of a programmer who has just started to learn coding, it is already quite hard to understand the logics and other language specifics and on top of that some languages use function names which do not make sense to a "newbie's" brain. For example, in C, if the programmer wants to take an input from the user, the programmer has to use the function called 'scanf' which stands for 'scan formatted' but most of the people new to the language are not familiar would be confused when they are coding and might commit a mistake of writing 'scan' instead of 'scanf' and thus find an error. The same applies to Java, where the function is 'scanner'. While in Python the function for taking input is just 'input'; the user thus does not have to remember any other name to perform this task. The code for taking the input of age of the user in the aforementioned languages is provided below:

```
int age;

scanf("%d", &age);
```

Figure 4. Code for taking input from the user in C

Figure 5. Code for taking input from the user in Python

```
age = input("Please enter your age")
```

Figure 6. Code for taking input from the user in Java

- ***Programs in Python tend to be relatively concise as can be seen in the examples mentionedabove.***

```
importjava.util.Scanner;
class MyClass {
public static void main(String[] args) { Scanner
myObj = new Scanner(System.in);
            }
}
```

- **Python can be used in both teaching as well in real-worldprogramming:**

Most of the programming languages tend to be good in either teaching or in real-world programming. The languages supporting the former are relatively easier to understand usually do not have as powerful tools as needed in real-world programming. Thus, programming in these languages tends to take more time and is more complex. The languages used in real- world programming possess a very powerful set of tools but are not as easy to understand. So, naïve programmer interested in programming had to initially learn a language which had no real-world implications and then continue to learn a language which could be used in real- world programming. Python has been able to break this pattern since it is easier to understand and has a powerful set of tools. The implication being that it is useful in both teaching and real-world programming. Along with that, the language itself is quite versatile as it can be used by a wide spectrum of programmers ranging from data scientists to website and even game developers.

- ***Python is extremelyportable:***

It can be used on almost all the operating systems that exist today. Some of the examples might include Windows, Mac OS, Linux and UNIX. Moreover, the programs made in this language are executable across different platforms with a minimal to no amount of additional changesnecessary.

All above features are all that a programmer desires from a programming language, thus a naïve programmer learning Python will not regret for doing so in near future. Next section elaborates areas of applicability ofPython.

4. Applicability of Python in RealWorld

With all its appealing features, Python achieves the merit of being a versatile programming language. Given below are prominent areas of applicability [3] of Python in present world:

- ***WebDevelopment:***

When it comes to web development, Python needs to be one's go-to tool. That's because Python offers numerous options for web development. For instance, 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.

- ***GameDevelopment:***

As we mentioned earlier, 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.

- ***Scientific and NumericApplications:***

Thanks to its massive library base, 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 inPython.

Some of the most useful Python packages for scientific and numeric computation include:

- Get our free whitepaper!
- Data Science inHealthcare
- the next biggestthing
- Download Now
- SciPy (scientific numericlibrary)
- Pandas (data analyticslibrary)
- IPython (commandshell)
- Numeric Python (fundamental numericpackage)
- Natural Language Toolkit (Mathematical And textanalysis)

- ***Artificial Intelligence and MachineLearning:***

Artificial Intelligence (AI) and Machine Language (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 skillset 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. Some of the best Python packages for AI and ML are:

- SciPy for advancedcomputing
- Pandas for general-purpose dataanalysis
- Seaborn for datavisualization
- Keras, TensorFlow, and Scikit-learn forML
- NumPy for high-performance scientific computing and dataanalysis

Apart from these libraries, there are also other Python-based libraries like NLTK, Caffee, PyTorch, and Accord.NET, that are useful for AI and ML projects.

- ***DesktopGUI:***

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).

- ***SoftwareDevelopment:***

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 for Software Developers to use Python as a support language for building control, testing, and management.

For instance, SCons [7] is designed explicitly for build control,Buildbot [8] and Apache Gump [9] allow for automated continuous compilation and testing, and Roundup [10] and Trac are great for bug tracking and project management.

Python also supports data analyzation and visualization, thereby further simplifying the process of creating custom solutions minus the extra effort and time investment.

- ***Enterprise-level/BusinessApplications:***

Enterprise-level software or business applications are strikingly different from standard applications; as in the former demands features like readability, extensibility, and scalability. Essentially, business applications are designed to fit the requirements of an organization rather than the needs of individualcustomers.

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. Furthermore, Python has other tools for business application development, like:

- **Odoo**: an all-in-one management software that forms a complete suite of enterprise management applications.
- **Tryton**: a three-tier, high-level, general-purpose application platform, is another fantastic tool for building businessapplications.

- ***Education programs and trainingcourses:***

If there's any beginner-friendly programming language, it is Python. This work has emphasized it earlier many times and here it is reiterated – Python has an extremely straightforward syntax that's similar to the English language. It has a short learning curve and hence, is an excellent choice for beginners. Python's easy learning curve and simplicity are the two main reasons that make it one of the most used programming languages in educational programs, both at beginner and advancedlevels.

However, Python is not just great as an introductory language – even professional developers and coders all around the world rely heavily on Python.

- ***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 [11]. All of these languages share numerous similarities with Python on grounds like object model, syntax, and indentation.

- ***OperatingSystems:***

Yes, 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) [4]. Usually, Python is combined with the C programming language to design and develop operatingsystems.

- ***Web ScrapingApplications:***

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 [12].

- ***Image Processing and Graphic DesignApplications:***

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.

After discussing characteristics of Python and its application areas, table given below provides a comparison between C, Java and Python as all these three are prominent programming languages and each has grabbed strong market in itstime.

Table1: Comparison of Python with Java and C Languages

Feature	**C**	**Java**	**Python**
Period of evolution andfame	1969-1979	1995-till date	Evolved in 1989, became popular after 2010
Programming style	Procedure oriented	Pure Object Oriented	Imperative, functional, object-oriented, and procedural
Ease of Programming (in terms of LOC, simplicity ofcode)	Low	Low	High
Machine Level programmingsupport	Supports but difficult to program	Doesn't support	Supports with ease
Web application development	Doesn't support	Supports	Supports with ease
Artificial Intelligence application development	No	Very cumbersome	Yes

Graphics Designing applications	Difficult to design	Difficult to design	Support with ease
Game Development support	Not much	Quite minimal	Supports to great extent

CONCLUSION

This work has thrown light on features and capabilities of Python programming language which has become a buzzword in coding world today. It has been compared with other existing prominent programming languages which used to be first choice for naïve programmers earlier. From feature comparison it is clear that Python offers versatile features which make it suitable for programming in diverse domains. All those features were not available in languages like C and Java simultaneously. Thusany novice programmer may select Python as a language of coding to begin with and it will help a long way in career.

REFERENCES

1. KalyaniAdawadkar "Python Programming – Applications and future". Published in International journal of Advanced Engineering and Research Development, Special Issue SIEICON-2017, April 2017, pp.1-4.
2. K. R. Srinath, "Python – Fastest Growing Programming Language". Published in International Research Journal of Engineering & Technology, Vol.4, Issue 12, December 2017, pp.354-357.
3. Akshansh Sharma, Firoj Khan, Deepak Sharma, Dr. Sunil Gupta May 2020 "Python: Programming Language of Future". Published in International Journal of Innovative Research in Technology (IJIRT), volume 6, issue 12, pp.115-118.
4. https://www.upgrad.com/blog/python-applications-in-real-world/
5. https://www.python.org/
6. https://spoken-tutorial.org/
7. https://pypi.org/project/SCons/
8. https://pypi.org/project/buildbot/
9. https://gump.apache.org/
10. https://pypi.org/project/roundup/
11. https://medium.com/@tellmegist/11-programming-languages-influenced-by-python-4cdab987cd4b
12. https://analyticsindiamag.com/top-7-python-web-scraping-tools-for-data-scientists/
13. Bogdanchikov, M.Zaparov, R. Suliyeb, 'Python to Learn Programming'. Published in Journal of Physics: Conference Series , 423(2013).
14. Preetee K. Karmore, Gaurishankar L. Girhe, 'Programming Language Python: A Review'. Published in International Journal of Advanced Research and Innovative Ideas in Education, Vol. 6, issue 2,2020, pp. 1634-1637.