

# An Overview of Python to Learn Programming

*Dr Arun Kumar Marandi, Assistant Professor*

*Department of Computer & IT, ARKA JAIN University, Jamshedpur, Jharkhand, India*

*Email Id- dr.arun@arkajainuniversity.ac.in*

**ABSTRACT:** *We now have a plethora of programming languages to meet our requirements, but the most pressing issue is how to teach programming to beginners. We recommend Python for this job in this article since it is a programming language with a well-organized syntax and strong capabilities for solving any problem. Furthermore, it is extremely similar to basic math reasoning. In most top institutions, Python is selected as the main programming language for freshmen. Python makes it simple to write code. In this article, we provide several computer code samples developed in Java, C++, and Python, as well as a comparison of the three languages. To begin, this article discusses the benefits of Python over C++ and Java. The results of a comparison of brief program codes written in three distinct languages are then shown, followed by a discussion of how pupils comprehend programming. Finally, the experimental findings of students' programming course success are shown.*

**KEYWORDS:** *Dynamically, Development, Python, Interface, Processing.*

## 1. INTRODUCTION

Because it offers strong tools that mirror the way people conceive and create code, the Python programming language is best suited as a first language to learn for novice programmers. It also reduces the number of additional keywords required to create syntactically valid code[1]. This method seems to be more effective than teaching the C++ or Java languages, which contain a lot of words and components that are more connected to the language's details than to the implementation of an algorithm[2]. Additionally, teachers at over a dozen institutions have used it to teach basic programming to computer science students, including MIT, UC Berkeley, UC Davis, Sonoma State University, the University of Washington, the University of Waterloo, Luther College, and Swarthmore College[3].

Learning at least one programming language is essential for a computer scientist nowadays, since many breakthroughs and technologies rely on a comprehensive knowledge of computers, operating systems, software APIs, and certain hardware peripherals. All of those are developed by programmers who think in a certain manner. To acquire such mindset, one must learn one of the programming languages and get a certification in software development. It is critical for anybody beginning to learn programming to focus on programming principles rather than language details, since the latter may vary across programming languages. However, Python is the most advanced programming language. So the learner doesn't have to worry about memory management, which is required in C++, or class structure, which is required in Java, or variable types and declarations, which are required in virtually every programming language[4].

Python is a dynamically in Nationalized high-level, easy-to-learn programming language. Python is a computer language that may be used for a variety of purposes. Python's creation started as a hobby for its creator[5]. He rolled them because he wanted to come up with his own language. Use beautiful and easy-to-read indentations instead of indentations. Curly braces to be secured describes the code blocks that came with curly braces to be secured. There was a purpose in the statements, but it was not embraced love as it should have been. Machine learning and artificial intelligence are relatively new languages in contrast to other languages[6]. Python's intelligence has lately received a lot of attention[7].

Python was created in the late 1980s by Guido van Rossum at Central Wiskunde & Informatica (CWI) in the Netherlands and implemented in December 1989. Python was seen as the ABC language's successor, capable of handling exceptions and interfacing with other Amoeba operating system systems. The name Python was inspired by Guido's fondness for the television program Monty Python's Flying Circus. Python is an interpreted

and dynamically typed programming language, which means programmers don't have to define or compile variable data types, and they may get instant feedback by utilizing the interactive command-line rather of waiting for the whole program to finish. Since Python version 2.1, the Python Software Foundation (PSF) has owned the intellectual property rights to the language. Python is the most popular programming language today. Python's popularity in data science is one of the main reasons behind its rise. Python is used to build YouTube, Google, Instagram, Reddit, Spotify, Dropbox, Quora, and other digital applications. IBM, Disney, NASA, Instagram, Spotify, Amazon, Survey Monkey, and Facebook are just a few of the businesses that utilize Python[8].

### 1.1 Advantages Of Python:

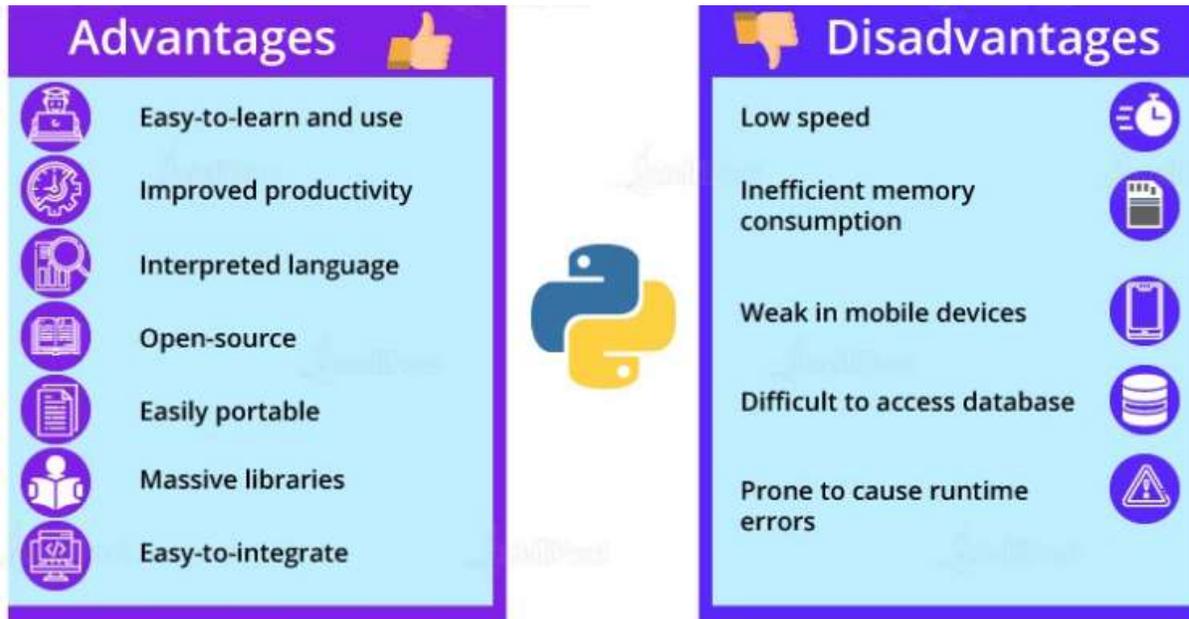
- *Simple to Read, Understand, and Write:* Python is a high-level programming language with syntax that is similar to English. This makes the code simpler to read and comprehend. Python is very simple to pick up and understand, which is why many people suggest it to newcomers. When compared to other popular languages like C/C++ and Java, you require less lines of code to accomplish the same job[7].
- *Productivity Gains:* Python is a highly useful programming language. Python's simplicity allows developers to concentrate on the issue at hand. They don't need to spend a lot of time learning the programming language's syntax or behavior. You write less code and accomplish more[9].
- *Language Interpretation:* Python is an interpreted language, which means that the code is executed line by line by Python. In the event of a mistake, it halts the program's execution and communicates the problem. Even if the program has many faults, Python only displays one. This facilitates debugging[10].
- *Typed Dynamically:* Until we execute the code, Python has no idea what kind of variable we're dealing with. During execution, it allocates the data type automatically. The programmer is not required to declare variables or their data types.
- *Open-Source and Free Software:* Python is released under an open-source license that has been authorized by the OSI. As a result, it is both free to use and share. You may get the source code, tweak it, and even release your own Python version. This is helpful for companies who wish to change a particular behavior and create their own version.
- *Support from Large Libraries:* Python's standard library is enormous, and it contains nearly all of the functions required for your job. As a result, you won't need to rely on third-party libraries. Even if you do, a Python package manager (pip) makes importing additional excellent programs from the Python package index much simpler (PyPi). There are approximately 200,000 packets in all.
- *Convenience:* You must modify your code in various languages, such as C/C++, to execute the application on multiple systems. With Python, however, this is not the case. You just have to write it once and it may be used everywhere.

### 1.2 Disadvantages Of Python:

- *Slow Motion:* Python is an interpreted and dynamically typed language, as we said before. Code execution that is carried out line by line is notoriously sluggish. Python's sluggish performance is due to its dynamic nature, which requires it to do more work when running code. As a result, Python isn't recommended for projects where performance is critical.
- *Memory is inefficient:* Python must make a compromise in order to offer developer simplicity. The Python programming language consumes a lot of RAM. When we choose memory optimization in our apps, this may be a drawback.
- *Mobile Computing Is Weak:* Python is a popular language for server-side development. Python is not used on the client-side or in mobile apps for the following reasons. Python uses a lot of memory and has a sluggish processing speed when compared to other languages.
- *Access to Databases:* Python programming is simple and stress-free. However, when we interact with the database, it falls short. In contrast to popular technologies like JDBC and ODBC, Python's database

access layer is basic and undeveloped. Python is seldom utilized in businesses since they need seamless interaction of complicated legacy data.

- *Errors in the Runtime* : Because Python is a dynamically typed language, a variable's data type may change at any moment. In the future, a variable holding an integer number may store a string, resulting in Runtime Errors.



**Figure 1: Diagrammatic Representation of Advantages and disadvantages of python [INTELLIPAAT]**

### 1.3 Application of python

- *Web design and development*: We think you're all aware of the term "web development." It is one of Python's most well-known applications. Python is one of the most popular programming languages for web development since it comes with a broad range of frameworks and Content Management Systems (CMS) designed to make the life of a web developer easier. Flask, Django, Pyramid, and Bottle are popular web development frameworks, whereas Django CMS, Plone CMS, and Wagtail are well-known Content Management Systems. Python for web development also has a number of additional advantages, including security, scalability, and ease of use throughout the development process. Python also has out-of-the-box support for a variety of web protocols, including HTML, XML, commonly used e-mail protocols, and FTP. Python also offers one of the most comprehensive libraries for improving the functionality of web applications while also making them simpler to develop.
- *A graphical user interface for the desktop*: Don't allow the preceding Python apps on this list convince you that Python isn't suitable for desktop GUI development. Of course, there are instances when a headless interface is appropriate, but we all know that certain projects need a graphical user interface, right? Python, on the other hand, provides lots of possibilities for developers to create a completely functioning GUI for such applications. Python's understandable syntax and modular programming style are essential for building extremely fast and responsive GUIs while making the whole development process a breeze. PyQt, Tkinter, Python GTK+, wxWidgets, and Kivy are only a few of the numerous tools available for GUI creation in Python, despite the large list.
- *Image Processing*: The importance of image (pre)processing technologies has increased as a result of the growing usage of Machine Learning, Deep Learning, and Neural Networks. To meet this need, Python provides a number of modules that make many of a Data Scientist's early preparation chores considerably easier. OpenCV, Scikit-Image, and Python Imaging Library are some of the most popular

image processing Python libraries (PIL). GIMP, Corel PaintShop, Blender, and Houdini are some more popular image editing programs that utilize Python.

- *Text Processing*: Text processing is one of Python's most popular applications. Text Processing is strongly linked to Natural Language Processing for the uninitiated, but let's not go into that just yet. Text Processing enables you to manage large amounts of text while maintaining the freedom to organize it as you see fit. You're right if you're thinking about sorting lines, extracting text, reformatting paragraphs, and so on. What else may Text Processing be used for? You can do a lot more than that using Python's text processing features.
- *Applications for Business*: In many respects, business applications vary significantly from consumer software. To begin with, they provide a limited range of features rather than a large number of options. Second, unlike consumer apps, which are designed with a broad purpose in mind, the user group they target is extremely close-knit, typically an organization. The greatest thing about Python is that it is ideal for developing high-performance bespoke solutions, whether they are for consumers or for businesses. Education and training programs are number eight on the list. One of the greatest things about Python is that it's an excellent starting place for newcomers wanting to get into contemporary programming. The reason for this is simple: Python's syntax is quite close to standard English, making it very simple to understand and work with. Python also offers a lower learning curve than other programming languages. Python is one of the finest choices for instructional programs all around the globe because of this feature. A simple Google search for "online Python courses" would likely turn up dozens of websites providing Python classes at different degrees of difficulty. Coursera, edX, Udemy, the Python Institute, and Harvard are among the most popular online educational sites, but there are others as well.

## 2. DISCUSSION

Because Python lacks security characteristics (such as public/private/protected), the program becomes simpler, shorter, more stringent, and easier to comprehend. Python is also highly dynamic, allowing fields and attributes to be generated on the fly, which is not possible in JAVA or C++. Unlike C++, which has virtual and non-virtual functions, Python functions and class methods are polymorphic. In contrast to JAVA's limited syntax, operator overloading offers Python objects more power since it may be used for any natural expression. Python programming indentation is important for program structure; as a result, every Python program is simple to read and understand. In JAVA or C++, on the other hand, every student tries to make his or her program shorter by fitting it onto one line or by writing every statement with the same indentation, making the program harder to read and understand not just for the instructor but also for the student. Python's native libraries contain several fantastic algorithms. To perform large computations, the learner does not need to grasp lengthy arithmetic. In Python, it already exists as a native long data type. Any series of data may be sorted, found, sliced, and joined with the right tools. They may not always be successful, such as those developed in C++ for a particular purpose. However, for newcomers, it is not the most important factor to consider. Because Python has its own method of storing and using variables, the programmer no longer has to specify the kinds of variables; instead, the types of variables are determined by the data they hold. Of course, knowing how various kinds of variables are kept in memory is essential; this may be covered in lectures, but it is simpler to pass over the data types knowledge in practice. Python also offers an adaptable and straightforward collection of terms and instructions that aid students in learning programming.

## 3. CONCLUSION

Finally, we want to emphasize the importance of a basic knowledge of programming for a computer scientist. Furthermore, sophisticated programming languages seem to be difficult to master due to their complexity. Students' knowledge and motivation to study programming and computer science in general is imprinted by their first programming language. As a result, it is essential to use a precise and appropriate approach in the language acquisition process. Today, several of the world's top institutions either utilize Python to teach

programming basics to their students or create and teach their own simple compiler that students can understand. [2] As a consequence of their usage of Python, we may conclude that pupils have a good understanding of programming. They have a command line program that enables them to quickly test their ideas. Furthermore, the interpreter has a very useful assistance feature that constantly reminds you of the structure of particular classes. Students like the basic turtle library [3], which has a highly user-friendly interface for drawing on canvas. This aids children in comprehending basic concepts such as cycles and circumstances. However, the main outcome is dependent on the students' willingness to study programming and computer science. In the experiment part, we compared the outcomes of midterm marks from a course taught in the Java language to the results of the same course taught in Python. We saw a 16 percent improvement in outcomes.

#### REFERENCES

- [1] M. Troyer, M. Cox, J. Cox, and A. J. Cox, "Books [Two books reviewed]," *Comput. Sci. Eng.*, 2014.
- [2] C. Dierbach, "Python as a first programming language," *J. Comput. Sci. Coll.*, 2014.
- [3] S. Davies, *Beginning Programming with C++*. 1969.
- [4] D. Sarda, *Python for the busy java developer: The language, syntax, and ecosystem*. 2017.
- [5] G. Sayfan, *Mastering kubernetes automating container deployment and management*. 2017.
- [6] B. Burd, *Beginning Programming with java*. 2005.
- [7] A. Zeller, *Why Programs Fail, 2nd Edition A Guide to Systematic Debugging*. 2009.
- [8] M. A. Bramer *et al.*, *Ian plant*. 2009.
- [9] J. Glover, V. Lazzarini, and J. Timoney, "Python for Audio Signal Processing," *Int. J. Speech Technol.*, 2017.
- [10] P. Fayyad, U., Piatetsky-Shapiro, G., & Smyth, "Python for Scientific Computing Python Overview," *Comput. Sci. Eng.*, 2007.

