

# Why Python is Most Famous

Swapnil Raj<sup>1</sup>, and Mrinal Paliwal<sup>2</sup>

<sup>1,2</sup> SOEIT, Sanskriti University, Mathura, Uttar Pradesh, India

Correspondence should be addressed to Swapnil Raj; [swapnil.cse@sanskriti.edu.in](mailto:swapnil.cse@sanskriti.edu.in)

Copyright © 2021 Swapnil Raj et al. This is an open-access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**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**- Application, Dynamic, Data, Python, Programming.

## I. INTRODUCTION

Python is an interpreter high general computer languages. Its design concept increases code accessibility by using a lot of indentation. Its programming capabilities and object-oriented methodology are designed to help programmers write concise, logical code for small and big applications. Python is dynamic typed and garbage-collected. Structured (particularly procedural) computing, object-oriented computing, and function coding are among the programming paradigms it supports. It is commonly described to as a "battery packs contained" languages due to its vast standard library. Python was created in the late 1980s by Guido van Rossum as a successor for the ABC computer languages, and Py 0.9.0 was released in 1991. [1]. Python 2.0 were launched in the years 2000, and it included new features such as list comprehension skills and a trash collecting method based on references tracking. In 2008, Python 3.0 was released, and it was a major change to the technology that were not backward friendly. Python 2 was discontinued in 2020, with python 2.7.18 being the last release.

Python is 1 of the more widely used computer technologies. Python was designed in the late 1980s by Guido van Rossum of the Center Wiskunde & Informatica

(CWI) in the Holland as a successor for the ABC computer languages, which was inspired by SETL and could handle exceptions and connect with the Amoeba operating system. It became effective in December 1989. [2]. Van Rossum was the project's only programmer till June 12, 2018, where he declared his "permanent sabbatical" from his job as Python's Beneficent Dictator for Life, a nickname bestowed by the Py society to honor his long-term commitment as the project's primary judgment. In January 2019, active Py core developers elected a 5 "Steering Council" to supervise the project. As of 2021, the membership of this council are Barry Warsaw, Brett Canon, Carol Eager, Tom Wohar, and Pablo Galindo Salgado. On November 16, 2000, Python 2.0 was released, featuring a number of notable new capabilities, include a trash collectors that identifies cycle and Multilingual support. Perl 3.0 was published on December 3, 2008. It was a major language overhaul that wasn't completely backwards compatible. Many of the most important functionality has been backported to Both 2.6 and 2.7. Python 3 includes the 2to3 tool, which automates (at least in part) the traduction of Python 2 script to Python 3. The edge deadline for Tensorflow was initially set for 2015, however were subsequently moved out to 2020 owing to worries that migrating a large body of existing work to Python 3 would be challenging. It would no longer be updated with security patches or other improvements [3]. Just Python 3.6 and later are supported, since Python 2 is no longer supported. Because of security flaws in all version of Django, which might lead to arbitrary code administration and web cache poison, Python 3.9.2 and 3.8.8 were rushed out. Python is a high-level programming language that is dynamically semantic, interpreted, and object-oriented. Because of its elevated built-in data structures, dynamic typing, and dynamic binding, it's perfect for Rapid Applications Deployment and as a programming or glue language for integrating existing elements. Python's simple, easy-to-understand syntax encourages readability, which reduces program maintenance costs. Python supports modules and packages, which encourages software flexibility and code sharing. The Py language and its extensive standard library are available in code or binaries format for all main platforms and are open to use and share [4].

Python is famous amongst developers since it allows them to work more efficiently. Because there is no compiling step, the modify cycle is exceedingly fast [4]. Python

programs are easy to troubleshoot: a bug or bad input will never cause a segment failure. Instead, if the translator discovers a flaw, it raises an exception. If the program fails to detect the error, the translator shows a stack trace. A source level debugger allows you to inspect regional and globally variables, execute arbitrary expressions, create baselines, walk throughout the program one word at a moment, and so on [5]. Python is used to create the debugger, exhibiting Python's introspective capabilities. Adding a few print instructions to the source code, on the other hand, is usually the quickest way to diagnose a software: the fast modify cycles making this simple approach incredibly effective [6].

### A. Python 2 VS Python 3

When a new version of a programming language is released, it usually supports the features and syntax of the previous version, making it simpler for projects to transition to the newer version. However, when it comes to Python, the two versions, Python 2 and Python 3, are vastly different. The following is a list of changes between Python 2 and Python 3:

- Print is a statement in Python 2 that can be used as print "something" to print a string to the console. Print, on the other hand, is a function in Python 3 that may be used like print("something") to print something to the console.
- Raw input () is a function in Python 2 that accepts human input. It returns a string that represents the value entered by the user. We'll need to utilize Python's int() method to convert it to an integer. Python 3 on the other hand, makes use of the input () method, which interprets the kind of input provided by the user automatically. Using basic methods (int (), str (), etc.), we may cast this value to any type.
- The implicit string type in Python 2 is ASCII, while the implicit string type in Python 3 is Unicode.
- The range () method from Python 2 is not available in Python 3. The xrange() method is a version of the range() function that returns an range object that operates in the same way as a Java iterator. The function range () produces a list; for example, range(0,3) gives 0, 1, and 2.
- In Python 3, there is also a minor change in the way exceptions are handled. It defines a keyword as one that must be utilized. We'll go over it in the Python programming tutorial's exception handling section.

### B. Advantages of Python

- Python is an elevated computer languages having grammar that is comparable to French. It is easy to read, comprehend, and write. This makes it easier to understand and understand the code. Python is quite easy to learn and comprehend, that is how most experts recommend it to newbies. As opposed to more prominent technologies such C/C++ and Java, you can perform the same task with less lines of code.
- Increased Productivity: Python is a powerful coding languages. Python's brevity helps programmers to focus on the problem at hand. Students did n't have to invest a lot of time understanding the syntax or

behavior of the coding languages. You create fewer code yet get more done.

- Languages Perception: Python is an interpretation languages, that means it executes the program word by line. It suspends the project's operation and reports the fault in the case of a mistake. Even if the code has a lot of flaws, Python only shows one of them [7].
- Typed Dynamically: Python has no notion what sort of variable we're working with until we run the function. It generates the information kind dynamically throughout operation. Variable and their data types do not need to be declared by the developer.
- Accessible and Free Apps: Python is distributed with an accessible license approved by the Open Source Initiative (OSI). As a consequence, you may use and distribute it for free. You could get the software, modify it, and even publish you own Language version. This is useful for businesses who want to tweak a certain behavior and develop their own version of it [8].
- Large Library Support: Python's core libraries is massive, and it provides virtually all of the methods you'll need. You won't have to depend on third-party libraries as a consequence. Even if you do, using a Py program management (pip) makes loading extra great apps from the Py program index a lot easier (PyPi). In all, there are around 200,000 packets.
- Convenience: To run the program on numerous computers, you must adapt your program in different technologies, like C/C++. This is not the situation with Python, though. It just has to be written once and then it could be utilized wherever.



Figure 1: Diagrammatic Representation of Advantages of Python [ASHISHGEORGE1811]

### C. Disadvantages Of Python

- Slow Motion: As previously said, Python is an interpretation and dynamic coded languages. Row program processing is infamous for its sluggishness. Python's sluggishness is owing to its fluid structure, which necessitates it doing more work while executing code. As a consequence, Python is not suggested for tasks that need high speed.
- Storage is ineffective: In order to provide developer simplicity, Python needs make a tradeoff. Python is a memory-intensive programming language. This might be a disadvantage if we chose resource minimization in our applications.

- **Why Python Is a Weak Language for Server-Side Programming:** Python is a famous languages for server-side programming. For the following reasons, Python is not utilized on the client side or in mobile applications. As contrasted to other technologies, Python consumes a lot of space and has a slow processing speed. Python database access software is straightforward and uncomplicated. When it comes to interacting with the database, though, it falls short. Python's database connection interface is simple and immature, in comparison to popular standards like JDBC and ODBC. Python is seldom used in business since it requires the smooth integration of complex legacy data.
- **Runtime Errors:** Since Py is a dynamic coded languages, the data type of a variable might change at any time. A variable storing an integer number might store a string in the future, leading in Execution Problems.

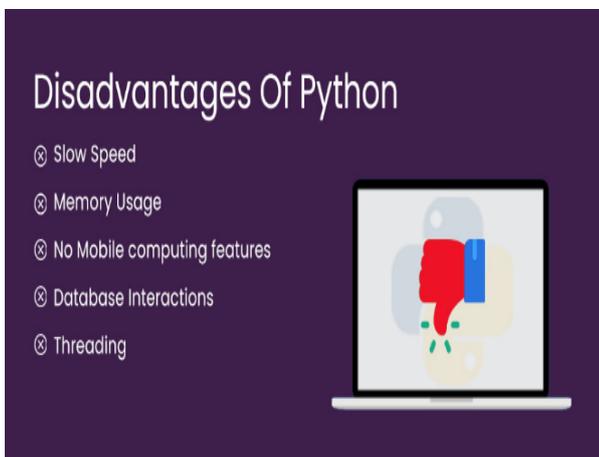


Figure 2: Diagrammatic Representation of Disadvantages of Python [MONOCUBED]

#### D. Application of Python

- **Web-based programs:** Python may be used to create web apps. It includes libraries for HTML and XML, JSON, email processing, request, BeautifulSoup, Feedparser, and other internet protocols. Instagram makes use of Django, a Python web framework. Python comes with a number of helpful frameworks, which are listed below:
  - i) Frameworks Django and Pyramid (Use for heavy applications)
  - ii) Bottle and flask (Micro-framework)
  - iii) Plone and Django are two popular content management systems (Advance Content management)
- **Desktop Graphical User Interface (GUI) Applications:** The Graphical User Interface (GUI) is a user interface that allows for easy interaction with any program. To create a user interface, Python includes the Tk GUI framework.
- **Application that runs on a console:** The command-line or shell is used to execute console-based programs. These are computer programs that are used to carry out instructions. This kind of program was more common in the previous generation of computers. Python is an excellent language for creating this kind of application. It is well-known for having REPL, or Read-Eval-Print Loop, which makes it the best language for command-line applications. Python has a number of free libraries and modules that aid in the development of command-line applications. To read and write, the required IO libraries are needed. It comes with built-in support for parsing arguments and generating console help text. There are additional advanced libraries that may be used to create standalone console applications.
- **Development of software:** Python is a programming language that is helpful in the procedure for creating software. It's a helper languages that may be utilized for things like administration and governance, testing, and more. Scorns are a method of developing power. Buildbot and Asf Gumps were used for automatic continual assembly and testing. Use Round or Trace for bug reporting and project management.
- **Numerical and Scientific:** This is the age of artificial intelligence, in which a computer can do tasks as well as a person can. The Python programming language is ideal for artificial intelligence and machine learning. It has a number of scientific and mathematical libraries that make doing complicated computations simple. Putting machine learning algorithms into practice requires a lot of arithmetic. Numpy, Pandas, Scipy, Scikit-learn, and other scientific and numerical Python libraries are available. If you know how to use Python, you'll be able to import libraries on top of the code.
- **Business Applications using Matplotlib:** Standard apps are not the same as business applications. Business applications include e-commerce and enterprise resource planning. This kind of program requires a lot of scalability and readability, which Python offers. Oddo is an example of a Python-based all-in-one application that provides a variety of commercial applications. The commercial application is built on the Tryton platform, which is provided by Python.
- **Applications that are audio or video-based:** Python is a versatile programming language that may be used to build multimedia applications. Tim Player, cplay, and other Python-based multimedia apps are examples. The following are a few multimedia libraries. Pyglet QT Phonon Gstreamer
- **3D CAD:** Engineering-related architecture is designed using CAD (Computer-aided design). It's used to create a three-dimensional depiction of a system component. The following capabilities in Python may be used to build a 3D CAD application.
- **Business Applications :** Python may be used to develop apps for usage inside a company or institution. Applications include OpenERP, Tryton, Picalo, and similar real systems
- **Image Processing Software:** Many libraries for working with images are available in Python. The picture may be modified to our specifications.

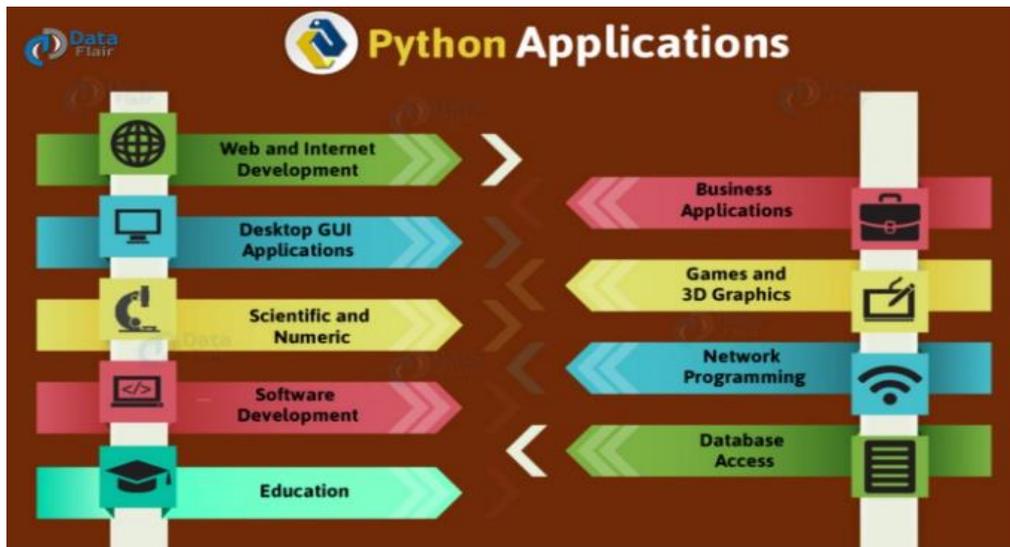


Figure 3: Diagrammatic Representation of Application of Python [MEDIUM]

## II. LITERATURE REVIEW

F. Pedregosa [9] Propose that Scikit-learn is a Py package that integrates a broad variety of cutting-edge computer training methods for recognized and unstructured issues on a moderate. This program concentrates on utilizing an overall elevated vocabulary to bring computer intelligence to non-specialists. Ease of use, speed, clarity, and API consistency are all prioritized. It has few requirements and is released via the BSD license, making it suitable for usage in either academia or business environments. <http://scikit-learn.sourceforge.net/> contains source code, binaries, and documentation.

M. Landy [10] Propose that Charlie Python's Flying Clown was a blend of the campy humor and the analytical, and it were probably of the more inventive comedy shows to ever run on television. Many of the norms of situation comedy, such as the completely formed and logical screenplay, story closure, recognizable character, and the etiquette connected with presenting, have been abandoned on the program. Its unusual global appeal defies the stereotype that comedy is a provincial and exclusive genre, and the show's cult fame continues to this day in the United Kingdom and the United Kingdom, thanks to repeats, videos, DVDs, and frequent performances by the movie's now-iconic actors. The majority of published assessments of Michael Python's Fly Circus concentrate entirely on the Pythons' members, history of the routines, or the evolution of later Monty Python ventures, leaving a scarcity of academic and context research on the television program itself. Marcia Landy's article is one of the few that examines the Flying Circus in its historical context, assessing the show's effect on British television in the 1960s and 1970s as much as English culture impact on the show's famous content. Landy examines not just why the series' complicated type of humor was significant, yet also how it were so highly accepted, noting the Pythons' eclectic mix of hilarious material, which includes the anarchic portrayal of sensuality, satire of church and class, and a criticism of tv as a media. The Flying Circus mocked both the high brow and the intellectual, tackling a variety

of previously taboo subjects, and here Higgins dismantles the movie's risks and reaction. This ebook would attract to cinema and media specialists, popular culture aficionados, and Python Python fans alike since it is both useful and amusing. Detroit Technological College Press owns the copyright to this work. All intellectual property rights are retained.

J. Glover [6] Propose that The usage of Python to construct sound signals analysis apps is discussed in this study. There are outlines of the Python programming languages, NumPy, SciPy, and Matplotlib, that combined create a sophisticated science computer environment. After that, we'll explain how SciPy was used to construct two audio computing frameworks, as well as how Python may be utilized in conjunction with the SndObj toolkit and Pure Information, two current ecosystems for music creation and signals analysis.

## III. DISCUSSION

Because Language lacks security features (such as public/private/protected), the program becomes simpler, shorter, more stringent, and easier to comprehend. Python is also incredibly dynamic, allowing fields and attributes to be built on the fly, something is not possible in JAVA or C++. Unlike C++, which has virtual and non-virtual functions, Django features and classes functions are polymorphic. Operator overloading provides Python objects more power since, unlike JAVA's constrained syntax, it may be utilized for any natural expression. Python coding underlining is important for program structure; as a result, every Django code is simple to read and understand. However, in COBOL or C++, everyone pupil tries to keep his or her code shorter by squeezing it onto one line or by writing each phrase with the identical indent, making the code harder to read and understand not only for the instructor but also for the student. Python's native libraries provide several fantastic algorithms. To do large computations, the pupil does not require to comprehend 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 created in C++ for a particular purpose. However, for newcomers, it is not the most important factor to consider. Because Python has its own style of storing and using variables, the developer no longer has to declare 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 vital; this may be explained in courses, but it is simpler to pass over the information kinds knowledge in practice. Python also includes an adaptable and straightforward collection of terms and instructions that aid students in learning coding.

#### IV. CONCLUSION

Finally, we'd want to emphasize the importance of a basic grasp of computing for a computing researcher. Furthermore, sophisticated computer technologies seem to be difficult to master due to their intricacies. Pupils' comprehension and willingness to pursue computing and computing science are imprinted by their first programming languages. As a result, it is essential to use a precise and appropriate technique in the language acquisition process. Today, several of the world's top institutions either utilize Python to teach programming basics to its pupils or design and educate their own basic compilers that pupils can understand. As a consequence of their employment of Python, we may conclude that pupils have a good understanding of programmers. They have a command line program that enables them to quickly test their hypotheses. Furthermore, the translator has a very useful assistance function that constantly informs you of the architecture of certain classes. Pupils like the basic turtle toolkit because it has an easy-to-use interface for drawing on canvases. This aids kids in comprehending basic ideas such as cycles and circumstances. However, the main outcome is dependent on the students' motivation to study coding and computer science. In the experiment part, we examined the outcomes of mid-terms marks from a study given in the Java languages to the outcomes of the identical course offered in Django. We saw a 16 percent boost in outcomes.

#### REFERENCES

- [1]. Kadiyala A, Kumar A. Applications of Python to evaluate environmental data science problems. *Environmental Progress and Sustainable Energy*. 2017.
- [2]. Ulloa R. Kivy: Interactive Applications in Python. *cs\_python\_in*. 2013.
- [3]. Zakka K. A Complete Guide to K-Nearest-Neighbors with Applications in Python and R. Kevin Zakka's Blog. 2016.
- [4]. Panggabean H, Tobing CA. Computational Linguistics Application Using Python Programming. *IOSR J Humanit Soc Sci Ver II*. 2015;
- [5]. Schmidt AG, Weisz G, French M. Evaluating rapid application development with python for heterogeneous processor-based FPGAs. In: *Proceedings - IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines, FCCM 2017*. 2017.
- [6]. Glover J, Lazzarini V, Timoney J. Python for Audio Signal Processing. *Int J Speech Technol*. 2017;
- [7]. Oxoli D, Zurbarán M, Shaji S, Muthusamy A. Hotspot analysis: a first prototype Python plugin enabling exploratory spatial data analysis into QGIS. *PEERJ Prepr*.

- 2016;
- [8]. Wagner M, Llorca G, Mercadal E, Giménez J, Labarta J. Performance Analysis of Parallel Python Applications. In: *Procedia Computer Science*. 2017.
- [9]. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. *J Mach Learn Res*. 2011;
- [10]. Landy M. *Monty Python's flying circus*. Monty Python's Flying Circus. 2005.