A Review on using Python as a Preferred Programming Language for Beginners

Vineesh Cutting [1], Nehemiah Stephen[2]

[1,2]Assistant Professor, Department of Computer Science & I.T., V.I.A.E.T., Sam Higginbottom University of Agriculture, Technology and Sciences, Prayagraj, India.

---***---

Abstract - *The choice of programming language to execute any required work related to programming or any other relevant domain is the foremost task for any beginner in computers. Choosing a beginner-friendly programming language is very important as it acts as a catalyst in the path of becoming a programmer. Selecting a convoluted language could result in loss of interest in the field of programming. This paper highlights the features of Python, along with its applications, advantages, and disadvantages, etc. It also discusses in brief the object-oriented concepts that can be implemented easily using Python that may help reader to understand to opt Python as a beginner in programming.*

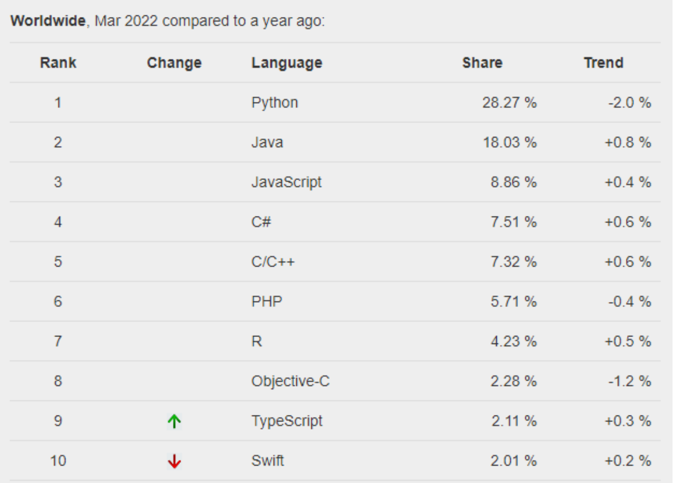
Key Words: Python; Programming Language; High-Level Language; Object-Oriented; Beginner.

1. INTRODUCTION

Python is s general purpose programming language that supports multiple programming paradigms. A high-level programming language that takes very few lines compared to other programming languages to accomplish a task. Python today is considered as one of the easiest programming language to begin with because of its super user-friendly coding style. The language comes with large number of built-in methods as a part of standard library. [1] The main features of Python are: it is simple and easy to learn, free to use and open source, high level programming language, platform independent, portability, dynamically typed, both procedure oriented and object oriented, interpreted, extensible, embedded, extensive library.

Guido van Rossum began working on Python in the late 1980s and first released it in 1991 as Python 0.9.0 [2]. It is dynamically-typed programming language which means the user doesn't have to declare the data type for the values to be stored in the program. In 2000, Python 2.0 was released while Python 3.0 was released in 2008 that is not completely backward-compatible with earlier versions [2].

According to PYPI: PopularitY of Programming Language, 2022, Python ranked as the number one programming language [3].



Worldwide, Mar 2022 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	28.27 %	-2.0 %
2		Java	18.03 %	+0.8 %
3		JavaScript	8.86 %	+0.4 %
4		C#	7.51 %	+0.6 %
5		C/C++	7.32 %	+0.6 %
6		PHP	5.71 %	-0.4 %
7		R	4.23 %	+0.5 %
8		Objective-C	2.28 %	-1.2 %
9	↑	TypeScript	2.11 %	+0.3 %
10	↓	Swift	2.01 %	+0.2 %

Fig. 1: Programming Language Popularity, 2022

[*Source: https://pypl.github.io/PYPL.html*]

Worldwide, Python is the most popular language, Python grew the most in the last 5 years (12.1%) and Java lost the most (-4.5%) [4].

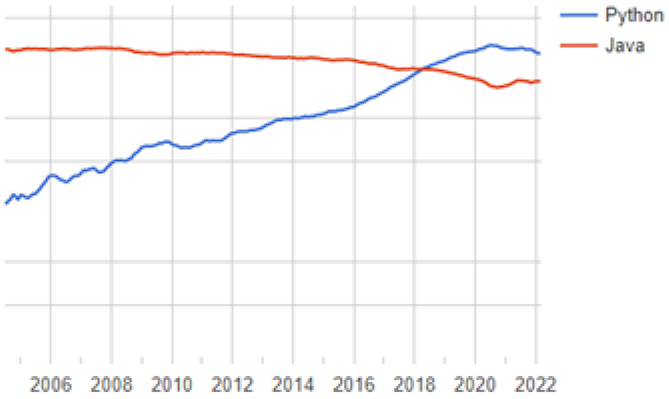
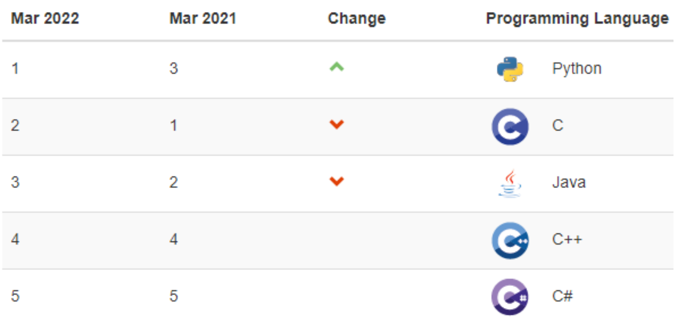


Fig. 2: Popularity Graph: Python vs Java, 2022

[source: https://pypl.github.io/PYPL.html]

As per the Tiobe Index also the Python ranked number one for Mar 2022 with a rating of 14.26 % with +3.95% gain compared to March 2021 during which Python occupied third position.



Mar 2022	Mar 2021	Change	Programming Language
1	3	▲	Python
2	1	▼	C
3	2	▼	Java
4	4		C++
5	5		C#

Fig. 3: TIOBE Programming Community Index, 2022

[*source: https://www.tiobe.com/tiobe-index]*

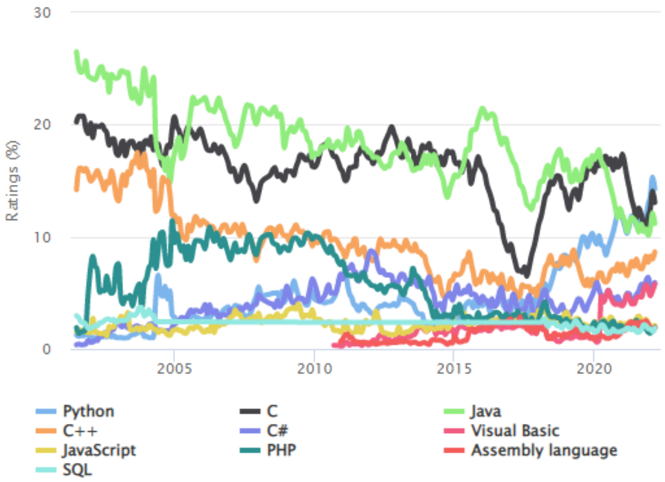


Fig. 4: Programming Language Ratings, 2022

[*source: https://www.tiobe.com/tiobe-index]*

According to StackOverflow Developer Survey, under most popular technologies, Python passed SQL to become our third most popular technology, and Node.JS moves to the sixth most popular technology [7].

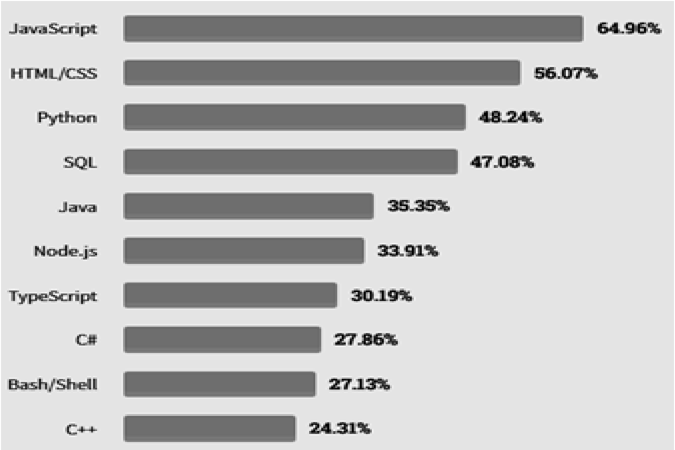


Fig. 5: Most Popular technologies, 2021

[*source: https://insights.stackoverflow.com/survey*]

There are over 40,000 jobs that are entirely for python experienced developers in major countries [3]. Job boards like Indeed and Naukri offer around 20,000 to 50,000 job listings for Python showing high career opportunities.

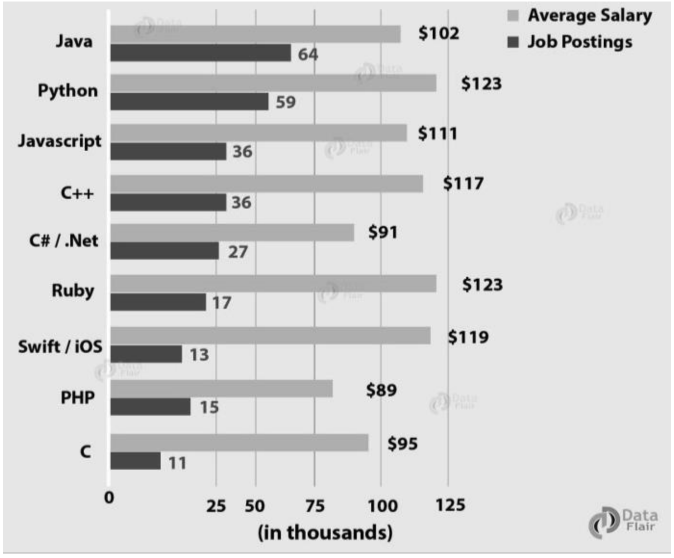


Fig. 6: Trending Programming Languages Salary

[*source: https://data-flair.training*]

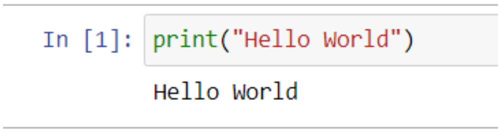
2. PYTHON PROGRAMMING LANGUAGE

Python Software Foundation (PSF) is a non-profitable organization entrenched as the intellectual owner of Python since version 2.1 [5]. Python has become the fastest-growing language. The popularity of Python in data science is one of the main reasons for the hike of Python [6].

2.1 Features of Python

A. Easy to Code

Unlike C, C++, C# and Java, the syntax of Python language is very straight forward and developer-friendly i.e., basic code can easily be learned and written in couple or hours or days. Python is one of the easiest to learn.



```
In [1]: print("Hello World")

        Hello World
```

Fig. 7: Hello World Python Program

B. Open Source and Free

Being an open-source technology, Python has a great community of coder, data scientists, etc. to contribute to Python's development process along with large set of libraries for different purposes. There are plenty of online forums where people discuss and share the python knowledge with other programmers and carry out conversations that cover different topics related to the language. Some of the popular Python Forums are: Reddit, StackOverflow, Source. Python, DigiForum, etc. For Windows, Mac or Linux user can download the latest version of Python for free.

C. GUI Support

There are large number of GUI frameworks available in Python. Majority of software now a days come with well-designed user-friendly GUI to make results more visual. Some examples of GUI frameworks in Python are: TkInter, PyQt, PySimpleGUI, etc.

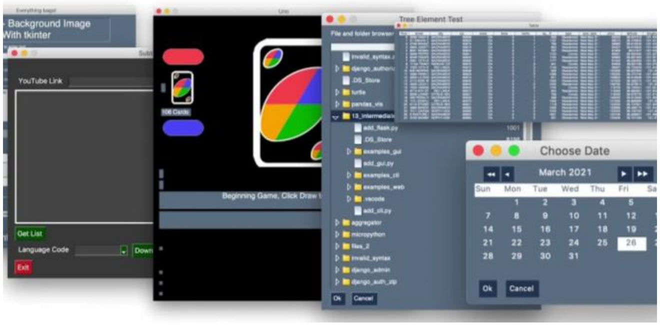


Fig. 8: Demo GUI Application

[*source: https://www.youtube.com/RealPython*]

D. Highly Portable

Python is compatible with wide variety of hardware platforms. Python code can execute on Linux, MacOs, Microsoft Windows, etc. Coder can run same code across various platforms seamlessly until any strict system-dependent code is included.

E. High-Level Language

Since Python is high-level programming language, the coder doesn't require to undertake much consideration of the underlying hardware architecture while writing the code.

F. Dynamically Typed

The data types for variable in python are not defined while writing the code instead Python automatically identifies the type and allocates memory accordingly thereby keeping the user free from data type mismatch errors.

G. Data Structures

Python comes with rich dynamic data structures such as lists, sets, tuple, dictionary, etc. as a part of standard installation therefore instantly ready to use.

H. Interpreted

Python code executes line by line without the need to be compiled first, which makes debugging easy and efficient for beginners compared to other compiled languages.

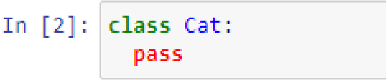
I. Language Interoperability

One of the excellent features of Python is versatility i.e., it supports the execution of code written in C, C++, Java, etc. User can call libraries and functions of languages like C, C++, etc. from Python using F2pyCtypes, Cython, etc [3].

J. Object Oriented

Python can be considered as hybrid language that can support fully object-oriented programming concepts as well as structured programming style. In fact, everything in Python is an object. OOP in Python helps defining real world entities in programming and describes what relationship is present between entities.

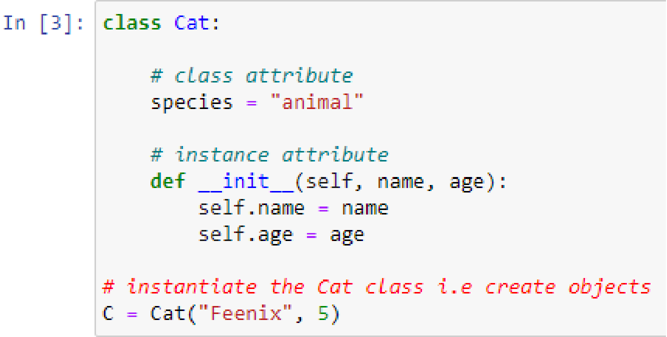
Class: User defined data structure used a blueprint to create an object.



```
In [2]: class Cat:
          pass
```

Fig. 9: Structure of Python Cat class

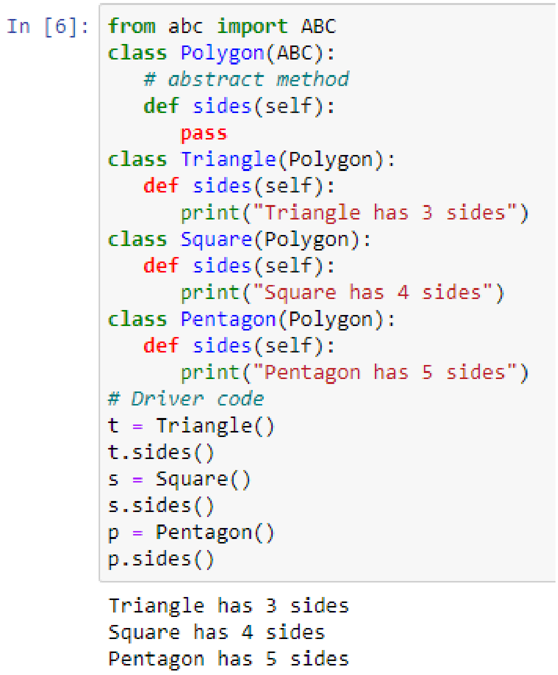
Object: An object is an instance of a class that has been allotted memory to execute. It contains the attributes and functions as per class definition. Syntax to create object is simple in Python.



```
In [3]: class Cat:

            # class attribute
            species = "animal"

            # instance attribute
            def __init__(self, name, age):
                self.name = name
                self.age = age

        # instantiate the Cat class i.e create objects
        C = Cat("Feenix", 5)
```

Fig. 10: Creating object of a class

Abstraction: The process of hiding internal details of implementation and displacing the necessary interface only.

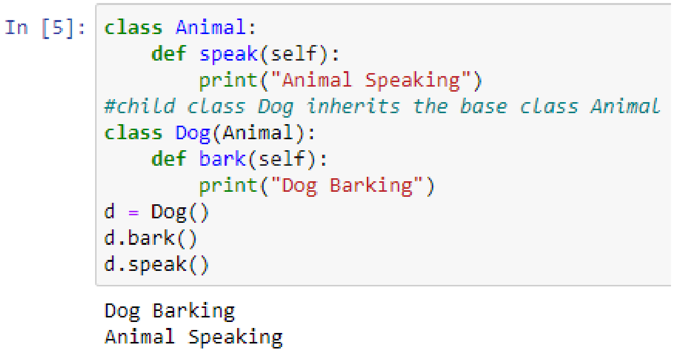


```
In [6]: from abc import ABC
        class Polygon(ABC):
           # abstract method
           def sides(self):
              pass
        class Triangle(Polygon):
           def sides(self):
              print("Triangle has 3 sides")
        class Square(Polygon):
           def sides(self):
              print("Square has 4 sides")
        class Pentagon(Polygon):
           def sides(self):
              print("Pentagon has 5 sides")
        # Driver code
        t = Triangle()
        t.sides()
        s = Square()
        s.sides()
        p = Pentagon()
        p.sides()

        Triangle has 3 sides
        Square has 4 sides
        Pentagon has 5 sides
```

Fig. 11: Abstract class Polygon with abstract method.

Encapsulation: Binding data and methods in a single unit, represented by class, represented as object.

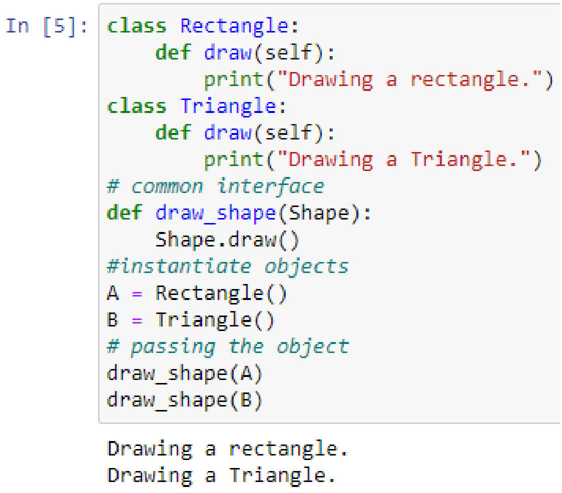
Inheritance: The process of passing properties of parent class to child class to enable the feature of re-usability in order to simply enhance the code in child class.



```
In [5]: class Animal:
            def speak(self):
                print("Animal Speaking")
        #child class Dog inherits the base class Animal
        class Dog(Animal):
            def bark(self):
                print("Dog Barking")
        d = Dog()
        d.bark()
        d.speak()

        Dog Barking
        Animal Speaking
```

Fig. 12: Inheritance of Animal attributes by Dog class

Polymorphism: Ability of an object to take many forms i.e., the same action can be performed in many different ways.



```
In [5]: class Rectangle:
            def draw(self):
                print("Drawing a rectangle.")
        class Triangle:
            def draw(self):
                print("Drawing a Triangle.")
        # common interface
        def draw_shape(Shape):
            Shape.draw()
        #instantiate objects
        A = Rectangle()
        B = Triangle()
        # passing the object
        draw_shape(A)
        draw_shape(B)

        Drawing a rectangle.
        Drawing a Triangle.
```

Fig. 13: Polymorphic function draw_shape

2.2 Applications of Python

A. Web Applications

Large number of frameworks are available on Python to develop web applications. For heavy applications Pyramid or Django can be used while for medium applications Flask or Bottle can be used. Some internet technologies like HTML, JSON, etc. can be easily handled by the libraries in Python. Cryptographic functions can also be implemented [8] for developing secure applications. [11] Both connection and connectionless protocols can be implemented using Python.

B. Scientific and Numeric

Artificial intelligence and machine learning are becoming part of everyday life, more and more applications are build having complex structures and algorithms that perform lot of mathematical and scientific computation to achieve the required task. Python comes with large numbers of libraries to the rescue, and because of its high-level data abstraction, libraries like, Numpy, Pandas, Scipy, Scikit-learn can be easily used to get the work done.

C. Software Development

Python is used to build control and management software using SCons. For automated continuous compilation and testing, Buildbot and Apache Gumpr can be used. For bug tracking and project management Roundup or Trac can be used.

D. Image, Audio and Video Processing

One of the fields where Python progressing is Image Processing. OpenCV, SimpleITK, Pillow are some libraries to process and manipulate images. Python also has the ability to process audio, video data using Pyglet, QT Phonon, etc.

E. Education

Python is best fit for any new learner because of high readability and easy to use syntax. From introductory to advance level, python stands great for people may or may not having any solid knowledge of coding. With great community support and availability of learning resources, a person can easily learn python and start experimenting with the domain specific libraries within few days.

2.3 Advantages of Python

- Code length is too short compared to other languages.
- Supports multiple programming paradigm.
- Python has large community support.
- Comes with large number of libraries as a part of standard installer.
- Dynamic typing reduces stress on new coders as data types are handled automatically.
- Has vast third-party libraries for different work domains.

2.4 Disadvantages of Python

- Runs slower compared to Java, C, C++ being interpreted.
- Since its dynamically typed, therefore allows lesser control over data types.
- Less preferrable for GUI driven programs being interpreted.
- Large memory consumption makes python less favorable for mobile devices.
- Underdeveloped database access layers compared to JDBC (Java DataBase Connectivity) and ODBC (Open DataBase Connectivity).

2.5 Future of Python

Python is getting popular day by day due to its reduced complexity to solve big problems. With the rapid increase in digital data generated everyday by softwares, users, customer, sensors, etc. the future of Python seems bright and promising, as a language of scientific computing. More and more new Python libraries are introduced for Artificial Intelligence and Machine Learning in order to automate tasks in various work domains. Python is in great demand to tackle the needs of Data science [12]. With the release of updates every year, Python remains relevant having new features added regularly. In terms of search volume for anyone interested in learning Python, it has skyrocketed to the 1st place when compared to other languages [6].

3. CONCLUSIONS

After going through the information furnished in the paper, it can be concluded that Python is highly suitable for beginners as a first language to learn and model real world entities using object-oriented concepts. According to features mentioned, Python is powerful, portable, easy to learn, open-source, free to download and use. Support for multiple programming paradigm attracts coders of different style and technique to work on a common platform. Some latest applications are also mentioned in the paper projecting a clear view that the requirement of Python is increasing in different work domains.

From beginner to advance level of coding, python remains efficient in solving complex problems as well as getting the required work done without getting into much of the implementation complexity because of the high level of abstraction and availability of numerous standard and third-party libraries.

REFERENCES

[1] P. N. S. Jyothi snd R. Yamagant, “A Review on Python for Data Science, Machine Learning and IOT”, International Journal of Scientific & Engineering Research Volume 10, Issue 12, December-2019, 851 , ISSN 2229-5518.

[2] “Python (programming language)”, Wikipedia, The Free Encyclopedia: https://en.wikipedia.org/wiki/Python_(programming_language)

[3] D. Lakshminarayanan and S. Prabhakaran, “A Study On Python Programming Language”, Dogo Rangsang Research Journal, Vol-10 Issue-07 No. 12 July 2020, ISSN: 2347-7180.

[4] PYPL PopularitY of Programming Language: https://pypl.github.io/PYPL.html

[5] Insanudin, E. "Implementation of python source code comparison results with Java using bubble sort method." Journal of Physics: Conference Series. Vol. 1280. No. 3. IOP Publishing, 2019. doi:10.1088/1742-6596/1280/3/032027.

[6] K. R. Srinath, "Python–The Fastest Growing Programming Language." International Research Journal of Engineering and Technology (IRJET) 4.12 (2017): 354-357.

[7] 2021 Developer Survey, StackOverflow: https://insights.stackoverflow.com/survey/2021

[8] Vidya, S.,”Network security using python”. IJSRSET, Vol. 4, No. 4, print, (2018), ISSN: 2395-1990, Online ISSN: 2394-4099.

[9] "Applications for Python", Python: https://www.python.org/about/apps/.

[10] "Python Applications", JavaTPoint: https://www.javatpoint.com/python-applications

[11] "5 application areas of Python programming language", TechGigBreau : https://content.techgig.com/5-application-areas-of-python-programming language/articleshow/69162767.cms

[12] S. Ahmad, "Data Science & Analysis using Python": https://www.researchgate.net/publication/332539010 _Introduction_to_Data_Science_ Analysis_using_Python