# Review of Python Programming

Ms.N.D.Sonwane[1], Mr.S.P.Taley[2], Ms.P.K.Karmore[3]

*[1] Assistant Professor, Computer Science & Engineering, Maharashtra, India*
*[2] Assistant Professor, Computer Technology, SDMP, Maharashtra, India*
*[3] Assistant Professor, Computer Science & Engineering, DBACER, Maharashtra, India*

## ABSTRACT

*The Python programming language is introduced, and promoted as an excellent choice as a first programming language for anyone who wants to learn programming. We give a brief history and synopsis of what makes it so popular, followed by a series of code examples comparing it to Java. Finally, it is argued that the rise in popularity of the World Wide Web has drastically changed the nature of programming, and that Python is a better general-purpose introductory language than Java, C++, or other traditional high-level languages. Python is chosen as a primary programming language for freshmen in most of leading universities. Writing code in python is easy. In this paper we give some examples of program codes written in Java, C++ and Python language, and we make a comparison between them. Firstly, this paper proposes advantages of Python language in relation to C++ and JAVA. Then it shows the results of a comparison of short program codes written in three different languages, followed by a discussion on how students understand programming.*

**Keyword: -** *python, java,, and C++*

## 1. INTRODUCTION

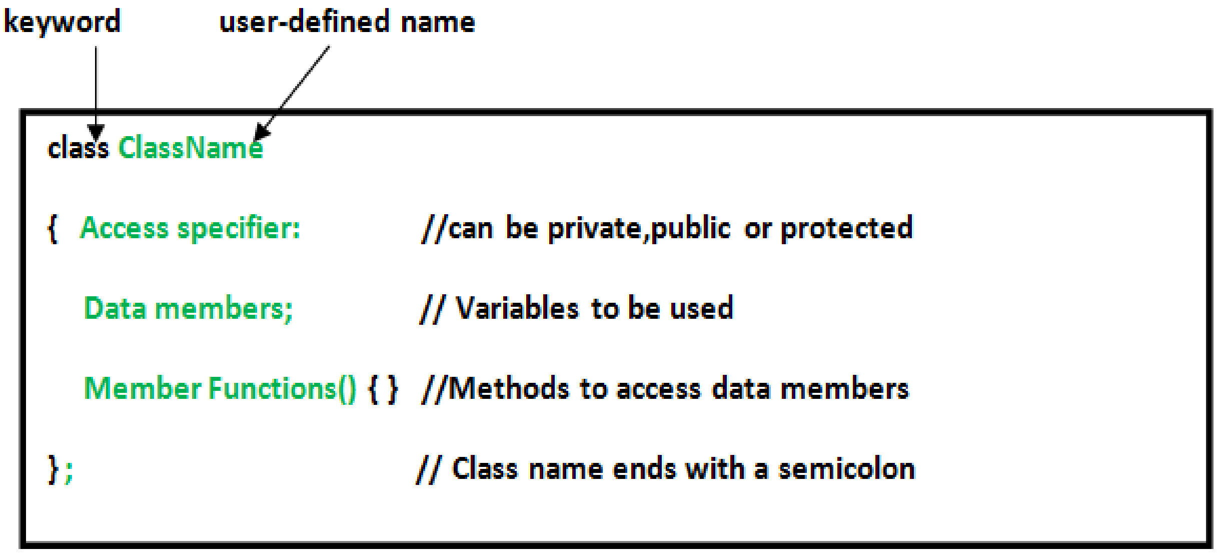
Python is a dynamically-typed programming language that supports object-oriented and functional programming. It's the brainchild of Guido van Rossum, who began writing Python in late 1989 and early 1990 as the scripting language for the Amoeba operating system. Its design was inspired by a number of other languages, including C, Modula-3, and particularly the educational language ABC. Van Rossum released it publicly in 1991, and since then Python has been gathering a large and enthusiastic group of users including professional programmers, educators, and computer scientists. Van Rossum, who is known to all as just Guido, or sometimes the BDFL ("benevolent dictator for life"), is still Python's lead developer and designer. Python gets its name from the Monty Python comedy troupe, and Python culture is peppered with references to those irreverent comedians (e.g. IDLE, after Eric Idle, is the name of the IDE that comes with Python).

Python is an elegant, simple, and practical language. It began life as a scripting language, and while that is still probably it's most common application, it has been used to solve many other problems. It compares favorably to Perl, the most obvious difference being Python's clean, easy to read syntax; Python programs are typically exceptionally readable and understandable. In contrast, Perl is often disparaged for its "line noise" syntax, due to its many arbitrarily chosen operators. Like Perl, Python is run as a large and successful open source project. It uses a collaborative design process where anyone can offer a new feature written up as a "Python Enhancement Proposal", or PEP. PEPs are debated on the computer language. Python newsgroup, or on one of the developer mailing lists. The final decision on which PEPs to accept is made either by developer consensus or through a “pronouncement by Guido.

There is not one single feature of Python that makes it such a good language. It has cherry-picked many of the best features from other languages, and combined them in a clear and synergistic way. One of Python's slogans is "batteries included". Python can do many useful things right out of the box, and provides all the basic tools and features that programmers have come to expect of a major programming language. It comes with a graphical IDE (including keyword coloring editor, a debugger, and an interactive command shell --- all written in Python with source code included), the Tkinter GUI library, a profiler, and numerous other tools that most other languages require you to add as extra packages. Not all these tools do everything a professional might need, but they often do; for beginning programmers, they are generally quick to learn and easy to use.

### *1.1* ***Reasons*** **to Choose Python as First Language**

You don't need to define the type of a variable in Python. Also, it's not necessary to add semicolon at the end of the statement.
Python enforces you to follow good practices (like proper indentation). These small things can make learning much easier for beginners.
Python allows you to write programs having greater functionality with fewer lines of code. Here's a link to the source code of Tic-tac-toe game with a graphical interface and a smart computer opponent in less than 500 lines of code. This is just an example. You will be amazed how much you can do with Python Python has a large supporting community. There are numerous active forums online which can be handy if you are stuck once you learn the basics.



```
class ClassName

{  Access specifier:           //can be private,public or protected

   Data members;               // Variables to be used

   Member Functions() { }      //Methods to access data members

};                             // Class name ends with a semicolon
```

### 1.2 Simple Expression in Python

**The Regular Function**

A *regular expression* is a special sequence of characters that helps you match or find other strings or sets of strings, using a specialized syntax held in a pattern. Regular expressions are widely used in UNIX world.

The module **re** provides full support for Perl-like regular expressions in Python. The re module raises the exception re.error if an error occurs while compiling or using a regular expression.

We would cover two important functions, which would be used to handle regular expressions. But a small thing first: There are various characters, which would have special meaning when they are used in regular expression. To avoid any confusion while dealing with regular expressions, we would use Raw Strings as **r'expression'**.

### *The* match *Function*

This function attempts to match RE *pattern* to *string* with optional *flags*.
Here is the syntax for this function −
re.match (pattern, string, flags=0)
Here is the description of the parameters –

| Sr.No. | Parameter & Description |
|---|---|
| 1 | **pattern**<br>This is the regular expression to be matched. |
| 2 | **string**<br>This is the string, which would be searched to match the pattern at the beginning of string. |
| 3 | **flags**<br>You can specify different flags using bitwise OR (\|). These are modifiers, which are listed in the table below. |

### *The* search *Function*

This function searches for first occurrence of RE *pattern* within *string* with optional *flags*.
Here is the syntax for this function −
re.search(pattern, string, flags=0)

Here is the description of the parameters –

| Sr.No. | Parameter & Description |
|---|---|
| 1 | **pattern**<br>This is the regular expression to be matched. |
| 2 | **string**<br>This is the string, which would be searched to match the pattern anywhere in the string. |
| 3 | **flags**<br>You can specify different flags using bitwise OR (\|). These are modifiers, which are listed in the table below. |

The *re.search* function returns a match object on success, none on failure. We use *group(num)* or *groups()* function of match object to get matched expression.

### Assigning Values to Variables

Python variables do not need explicit declaration to reserve memory space. The declaration happens automatically when you assign a value to a variable. The equal sign (=) is used to assign values to variables.The operand to the left of the = operator is the name of the variable and the operand to the right of the = operator is the value stored in the variable.

example –

```
#!/usr/bin/python

counter = 100          # An integer assignment
miles   = 1000.0       # A floating point
name    = "John"       # A string

print counter
print miles
print name
```

***Multiple Assignment***

***Python allows you to assign a single value to several variables simultaneously. For example −***

a = b = c = 1
Here, an integer object is created with the value 1, and all three variables are assigned to the same memory location. You can also assign multiple objects to multiple variables.
For example −
a,b,c = 1,2,"john"
Here, two integer objects with values 1 and 2 are assigned to variables a and b respectively, and one string object with the value "john" is assigned to the variable c.

**Standard Data Types**

***The data stored in memory can be of many types. For example, a person's age is stored as a numeric value and his or her address is stored as alphanumeric characters. Python has various standard data types that are used to define the operations possible on them and the storage method for each of them.***

Python has five standard data types −

- Numbers
- String
- List
- Tuple
- Dictionary

### 1.3 Web Application Development

Python has the lion’s share in the field of web development. Many employers look for full-stack programmers who know Python. And you can become one of them by learning frameworks (WAF) like Django, Flask, CherryPy, and Bottle which give extensive support for web development. All of these are developed using Python. These frameworks deliver essential features to simplify tasks related to content management, accessing backend database, and handling network protocols like HTTP, SMTP, XML-RPC, FTP, and POP. Some of the known online products created in Python are Plone (Content Management System), Zope application server, Quixote web framework, and ERP5, an open source enterprise solution used in the aerospace field.

### 1.4 Scientific and Numeric Computing

Python has become the obvious choice for working in Scientific and Numeric Applications. And there are multiple reasons for this advancement. First and foremost is that Python is a free and open source language. And it allows to modify and redistribute its source code.

Next, the reason for becoming it more dominant in the field of Scientific and Numeric is the rapidly growing number of specialized modules like NumPy, SciPy, Pandas, matplotlib, and IPython. All of these are available for free and provide a fair alternative to paid products like Matlab.

Hence, Python is becoming a leader in this field. The focus of Python language is to bring more productivity and increase readability.

### 1.5 GUI Programming

Python has some inherent qualities like clean and straightforward coding syntax as well as dynamic typing support. These work as the catalyst while developing complex GUI and image processing applications.

Python’s clean syntax and tremendous support of many GUI libraries (like wxWidgets, pyqt or pyside) made Programmers deliver graphics software like Inkscape, Scribus, Paint Shop Pro, and GIMP.

In addition to the 2D imaging solutions given above, Python is even propelling many 3D animation software like 3ds Max, Blender, Cinema 4D, Houdini, and Maya. These applications integrate with Python for automation to speed up their workflows and eliminate the need for doing them manually.

### 1.6 Software Prototyping

Python has many qualities that make it a natural choice for prototyping. The first is being an open source programming language, a massive no. of users follow and contribute to its development. Further, the lightness, versatility, scalability, and flexibility of refactoring code in Python speed up the development process from the initial prototype.

## 2. CONCLUSIONS

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python’s elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. Python gives us a easy-to-use interface to create prototypes.

## 3. REFERENCES

[1]. Charles Day,Copublished by the IEEE CS and the AIP Computing in Science & Engineering 1521-9615/14/$31.00 © 2014 IEEE.
[2]. Richard L. Halterman “LEARNING TO PROGRAM WITH PYTHON” 2011.