

# Python- A Promising Future

Priyanka Gupta

**Abstract – This Paper is an attempt to promote the use of Python for general Scientific Research and development in today’s world where lot of programming language are available, and Programmers are happy coding with Java. Net. Do we really require a language and if yes what is the difference it makes.**

The answer lies here:-

We need something that we can deploy on PCs, Macs, handhelds, phones, the web and it has to let us build GUIs quickly and painlessly and believe Python can do all that. This Paper is presented with comparative study of Python with C++ and how Python fits into the picture with various platforms it can run on.

## I. INTRODUCTION

Ignoring Python in 21<sup>st</sup> century can prove to be a costly mistake. Python is lot like any other general purpose programming language with statements, expressions, operators, functions, modules, methods and classes. And there is something more that Python provides-Lists that makes your work lot easier. For many things we can rely on Python’s numerous data structures, classes, nested functions, iterators, the flexible function calling syntax, an extensive sink included standard library, great scientific libraries and outstanding documentation.

## II. Python features

1. **Readability-** Python’s syntax is

very well thought out. Unlike many scripting languages was a primary consideration when Python’s syntax was designed. Infact the bliss is that turning pseudocode into correct python code is a matter of correct indentation. This readability has a number of beneficial effects. The following are the words of Python’s original author:-

This emphasis on readability is no accident. As an object oriented language python aims to encourage the creation of reusable code. Even if we all wrote perfect documentation all of the time, code can hardly be considered reusable if it’s not readable. Many of the python’s features, in addition to its use of indentation, conspire to make Python code highly readable. This pseudo-code nature of python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the language itself.

As an example, consider the following code:-

```
def func(a, flag=0):  
    ‘Whenever we type anything by defining the  
    functions it should be properly indented’  
    return true  
# Calling the above function func  
(b, flag=1)
```

In the above code there are three aspects:

- a. It is a small, self contained function that only requires three lines to define, including documentation.
  - b. Second, a default argument for the flag is specified in a way that is instantly readable.
  - c. It is easy to parse this code.
2. **Every String has its place-**

All programs process data and one of the most important types of data is text. It is possible with the help of methods in Python and to bend the data according to the requirement.

**3. As the Program grow, the code often becomes more complex:-**

And complex code can be hard to read and even harder to maintain. One way to manage the complexity is to create functions. Functions are snippets of code that are used as needed from within your program.

**4. Sort it out**

As the program develop, so does data handling needs. Using an individual variable for each piece of data gets really old. So Python presents Container

**5. Going GUI**

GUIs don't just process events. They also handle data.

**6. Free and Open Source**

Python is an example of a FLOSS. In simple terms, you can freely distribute copies of this software, read it's source code, make changes to it, use pieces of it in new free programs, and that you know you can do these things. FLOSS is based on the concept of a community which shares knowledge. This is one of the reason why Python is so good –it has been created and is constantly improved by a community who just want to see a better Python.

**III. The Problems**

The emphasis on java & other commercial languages in comp.sc has established the perception of CS as a dry & technically difficult discipline among students who are still seeking career. But their negative perception may have a devastating effect on comp.sc programs & therefore should not be ignored.

**IV. Problems with java**

In java the problems is that the class concept requires the knowledge of numerous underlying concepts, such as methods, type, parameter, array, access-level.

**V. Practical**

While purely education language were appropriate for beginning program education, they were not particularly suitable for commercial programming.

Any languages that should be used for introductory comp.sc education is subject to 2 conflicting requirements.

First-such a language should be simple enough so that it can be interesting & can be handled by the amateurs.

Second-such a languages is expected to be commercial & object oriented, which implies significant complexity.

**VI. Moving To Python**

We have investigated the use of interpreted languages to create a programmable, dynamic environment in which components can be tied together at a high level. such an approach has taught us about the features of the interpreted languages that are a key to a successful component integration.

**VII. Python's suitability**

A promising unique quality of python is that it is neither a purely educational languages, nor a purely commercial languages. It was originally designed for education but soon gained popularity among practical programmers who became the driving force in its evolution.

Python is just not a teaching language. Ignoring Python can prove to be a costly mistake. Because of its simplicity and elegance it has been embraced by top technology companies such as Google, Dropbox , Mozilla ,Quora ,Intel ,IBM.

## **VIII. REFERENCES**

[1]Atanas Radenski,“Python First:A Lab Based Digital Introduction to Computer Science”

[2]Anshul Kumar Jain, Manu Sheel Gupta  
The Python Papers Monograph 2:16

[3] M.F. Sanner, “Python: A programming language for software Integration and development.”, The Scripps Research Institute

## **IX. BIBLIOGRAPHY**

Priyanka Gupta is Master Of Computer Applications and Management and currently working in teaching profession. She has an area of interest in Cryptography and Information Security.