

Python versus C Language: A Comparison

¹Rohit Kumar, ²Subhash Chander and ³Mandeep Chahal,
^{1,2,3}Government College Raipur Rani, Kurukshetra University, Kurukshetra, India

Abstract: C and python has been proven programming language in their respective domain. Python has been a newer language in comparison to C language and offers rich set of libraries and offers significant support for newer application areas such as machine learning, whereas c has a proven and benchmark language for system programming and real time application development. Ease of coding in one perspective that differentiate these programming languages. Other perspective is the application areas addressed by these programming languages. On this research paper all these parameters including efficiency, portability, application domains etc. has been discussed in details and comparison has been done for the same. Poularity comparison of both the langauges hgas also been done in w.r.t. each other ans w.r.t other majorly used programming languages.

Keywords: Python, C, Programming Languages, Efficiency.

I. INTRODUCTION

A. Need for Comparison

C programming language has been a predominately used programming language since 1970's and has proved itself as a revolutionary language. Its efficiency, faster execution and interaction with hardware makes it suitable for system programming and real time application development. Python on the other hand is high level language very rich in libraries and offers a lot more added features for newer application like machine learning and many more. In order to figure out a suitable language for a given problem a comprehensive comparison need to be done between both the languages [1-5]. In this paper a detailed comparison between both languages has been done and presented in detail.

In comparison to Python computer language, C language needs deeper insight for programming and implementation as python is an object oriented general purpose programming language and can be run under an interpreter. It is fully formed built-in language designed for predefined library functions. On the other hand C language is a structure oriented programming language which is applied for developing applications for hardware operations and run under a compiler. Built-in functions are also very less in C language.

II. NEEDS FOR COMPARISON

In present digital era, computer is widely used in every sphere of life let it be medical field or engineering, research field or day to day activities. More and more programming languages are also being developed to design the programs which cater the needs of these activities. Hence one should have the insights for these languages so that it can be decided as which computer language will suit for developing any particular program.

A. Introduction to Python

Since its development by Guido Van Rossum in 1991, the Python language has been widely used for developing the

object oriented programs. Code readability is its main feature. Other features of the language includes that it is an interactive programming language which uses dynamic typing. It supports imperative, object-oriented and functional programs which makes its features as hybrid language. It is used for the development of full scale programs besides its use as a scripting language [5-10].

B. Python for Machine Learning

In this article we will have an overview for machine learning and try to tell that how it is different from statistical inference as machine learning is an important concept in data analysis and modeling. It is obvious that in last twenty years a large number of Algorithms have been developed and implemented in most of the programming languages. We will also review Scikit-learn package in the Python language popularly used in data science. It is a convenient toolkit for educational and behavior statisticians why because this package includes implementations of comprehensive list of machine learning methods with unified data and modeling procedure conventions.

C. Characteristics of python:

Characteristics of python programming language includes that it can be used for real world programming due to its design. It is very high-level, dynamic, object oriented, general purpose programming language which uses interpreter and can be used in varieties of applications. Its portability increases its popularity as it can be used with many operating systems such as Windows, Linux, Unix, Amigo, MacOS etc. Python programs can be run without making any changes from one platform to another. But its performance is quite low because python is dynamically typed language and too flexible. Hence it needs a lot of referencing to ensure the definition of something. It is not easy to maintain also as something can mean differently depending on the context.

D. Open Source

Despite its all program rights reserved for the python institute, it is an open source and has no limitations in using, changing and distributing even for commercial use. Changes can be made in python's source code also and it can support COM, .Net etc objects.

To enhance the performance of an application, python code can be combined easily with the pieces of C/C++ or other languages which makes this language an extensible and embeddable language.

Unlike C/C++, one need not to be worried for daunting tasks like memory management, garbage collection and so on with python language as it a high level interpreted language. Same when python code is run, it instantly converts it to the language that can be understood by the computer, so there is no matter of worry for lower level operations.

Python has a number of standard libraries such as network and internet programming to solve the common tasks. Because of this feature one need to write all the code oneself which makes the life of a programmer comfortable with python language.

With the help of object oriented programming (OOP) features of the python language, complex problems can be solved by breaking these problems into smaller sets by creating objects. Some other features of this language include, System Programming, Graphical User Interface development, Network and internet programming etc.

Introduction to C Language C language, a general-purpose programming language, was created by Dennis Ritchie at the Bell Laboratories in 1972. Though it is an old language, but still it is a very popular. This language is closely associated with UNIX as initially it was developed to write the programs of UNIX OS.

III. C PROGRAMMING LANGUAGE

Since the syntax of C language is similar to the other popular languages like Java, python, C++, C# etc therefore if you have the knowledge of C language you can learn these languages also very easily. It is very fast and versatile language hence it can be used in both applications and technologies.

A. Key Features of C language

Some its significant features which make this language beneficial for developing a website and software are listed below:

1. The basic syntax style of this language is very simple and easy to understand that enables the programmers to redesign or create new applications and increases its comprehensibility. Hence it is used as an introductory language to develop programs for school students.
2. C is a faster language as it is a statistically typed language and an statically typed language is always faster than a dynamic language. It has compiler-based programs which makes the compilation and execution of codes faster.
3. In comparison to other newer languages which come with numerous features, C has only essential features. Hence C has only limited features to process which makes it faster.
4. Portability is another feature of this language which means machine independency i.e. Without doing any specific changes the fraction of a code created in C language can be run on various machines.
5. Extensibility is very significant feature of this language as the new features can be added to the codes already written with few alterations.
6. It has extensive set of libraries with several built-in functions which makes it easier even for a beginner. User defined functions can be added to C libraries which enables the programmer to develop a vast variety of programs.
7. Dynamic Memory Management (DMM) feature make this language one of the most significant feature as with the help of this feature, the size of the data structure can be utilized and managed during runtime. Free(), malloc(), calloc and realloc() are such functions available in C language by which the operations on data structure and memory allocations

can be performed and allocated memories can be freed at any time.

8. Since C is a general purpose structured language therefore a code can be broken down into many parts using functions and can be stored in the form of libraries for future use. This feature also increases its visual appeal and makes it less prone to any error.
9. C is mid-level programming language though it was designed initially to support low –level programming and because of this feature, C allows hardware manipulations also.
10. Use of pointers in C language makes it user friendly to operate with memory, arrays, functions and structures.
11. With the help of recursion feature provided in C language, a function can be created that can call itself any number of time until a given condition is true.

B. Interest Chart of C and Python:

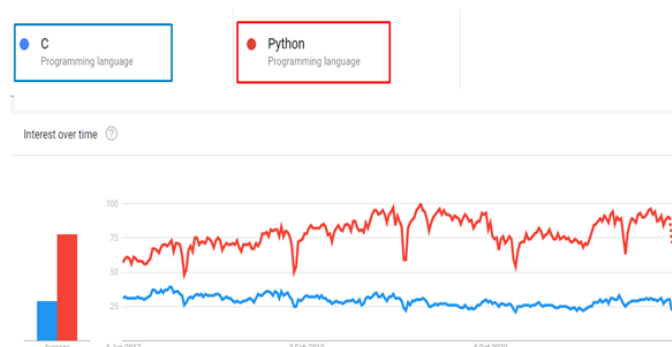


Figure: 1 User Interest Comparison

IV. COMPARISON OF OTHER POPULAR LANGUAGES

Most popular programming language in 2020

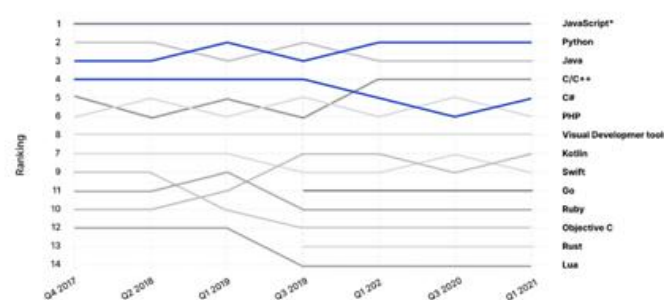


Figure 2: Ranking Comparison of C, Python with other languages.

In figure 2 ranking in terms of users has been shown. Comparison include popular languages like Java, C++, C# and many more.

Tabular Comparison of C and Python:

A detailed tabular comparison of both the programming languages has been presented in table 1. User interest level between python and C and with other languages has been presented in figure 1 and figure 2 respectively.

Table 1: Comparison of C and Python

C	Python
C language follows an imperative programming model.	Python is an object-oriented programming language
C is mainly designed for hardware applications	Python is a general purpose programming language.
Pointers are available in C language.	No pointers functionality available.
C runs under a compiler.	Python is interpreted.
A limited number of built-in functions.	Large library of built-in functions.
Code execution is faster than python.	Slower compared to C as python has garbage collection.
Implementing data structures required its functions to be explicitly implemented.	Gives ease of implementing data structures with built-in insert, append functions.
It is compulsory to declare the variable type in C.	No need to declare a type of variable.
C program syntax is harder than python.	Python programs are easier to learn, write and read.
In line, an assignment is allowed.	In line, assignment gives an error. E.g. a=5 gives an error in python.

CONCLUSION

A tough question arises as to when to use python and when to user C. C vs Python languages are similar yet have many key differences. These languages are useful languages to develop various applications. The difference both is that python is a multi-paradigm language and C is a structured programming language. Python is a general-purpose language that is used for machine learning, natural language processing, web development and many more. C is mainly used for hardware-related application development such as operating systems, network drivers. Before deciding on particular language keep in mind following things

Ease of development: – python has very fewer keywords and more free English language syntax whereas C is far more difficult to write and maintain. Hence if you want an easy development process go for python.

Performance: – Python is much slower than C as python takes significant CPU time for interpretation.

References

- [1] Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman, "Compilers: Principles, Techniques, and Tools", Addison-Wesley, 1988.
- [2] Yuri Gurevich, "Evolving Algebras: An Introductory Tutorial", Bulletin of European Association for Theoretical Computer Science, February 1991. (A slightly updated version will appear in the EATCS Book of Columns, World Scientific Publishers.)
- [3] Yuri Gurevich and James K. Huggins, "The Evolving Algebra Semantics of C: Preliminary Version", CSE-TR-141-92, EECS Department, University of Michigan, 1992.
- [4] Brian W. Kernighan and Dennis M. Ritchie, "The C Programming Language", 2nd edition, Prentice Hall, 1988.
- [5] TIOBE Software Index (2011). "TIOBE Programming Community Index Python". 1
- [6] "Programming Language Trends - O'Reilly Radar". Radar.oreilly.com. 2 August 2006.
- [7] "The RedMonk Programming Language Rankings: January 2011 – tecosystems". Redmonk.com.
- [8] Summerfield, Mark. Rapid GUI Programming with Python and Qt.
- [9] Kuhlman, Dave. "A Python Book: Beginning Python, Advanced Python, and Python Exercises"
- [10] Bernado-Mansilla, E., Ho, T. K. (2004). On classifier domains of competence. Proceedings of the 18th International Conference on Pattern Recognition, Cambridge, England, August 22–26, Vol. 1, pp. 136–139.
- [11] Bishop, C. M. (2006). Pattern recognition and machine learning. New York, NY: Springer-Verlag.
- [12] Blyth, C. R. (1972). On Simpson's paradox and the sure-thing principle. Journal of the American Statistical Association, 67, 364–366.
- [13] Caruana, R., Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. In Cohen, W., Moore, A. (Eds.), Proceedings of the 23rd International Conference on Machine learning (ICML '06, pp. 161–168). New York, NY: Association for Computing Machinery.
- [14] Chen, A., Bengtsson, T., Ho, T. K. (2009). A regression paradox for linear models: Sufficient conditions and relation to Simpson's paradox. The American Statistician, 63, 218–225.
- [15] Fu, K. S. (1981). Syntactic pattern recognition and applications. New York, NY: Prentice Hall.
- [16] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Bengio, Y. (2014). Generative adversarial nets. Proceedings of NIPS, pp. 2672–2680.
- [17] Hastie, T., Tibshirani, R., Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction. New York, NY: Springer.
- [18] LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. Nature, 521, 436.
- [19] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.
- [20] Robinson, D. (2017, September 6). The incredible growth of Python. Retrieved from <https://stackoverflow.blog/2017/09/06/incredible-growth-python/>
- [21] Stevens, S. S. (1946). On the theory of scales of measurement. Science, 103, 677–680.
- [22] Sutton, R. S., Barto, A. G. (1998). Reinforcement learning: An introduction. Cambridge, MA: MIT Press.
- [23] Witten, I. H., Frank, E., Hall, M. A., Pal, C. J. (2016). Data mining: Practical machine learning tools and techniques. San Francisco, CA: Morgan Kaufmann.