

Use of Python in Research

Prashant kumar¹, Dr Ashendra Kumar Saxena²

¹MCA, CCSIT, Teerthanker Mahaveer University, Moradabad

²Professor, CCSIT, Teerthanker Mahaveer University, Moradabad

¹tmu.prashant@gmail.com

²drashendra.computers@tmu.ac.in

Abstract— In this paper we are going to represent need and use of python in research. Python with simple and easy to learn syntax along with its power of being a general purpose scripting language enables researchers to solve complex problems interactively. The variety and quality of features of python. Python provide various libraries and platform which is useful for education, software development, scientific and numerical areas.

Python used various useful programming technique such as object oriented, functional, parallel programming etc. Python as the perfect programming for research environment.

Keywords— python, scripting language, matlab, object oriented.

I. INTRODUCTION

Python is a modern programming language that is increasingly used in the field of research. Python is interactive, interpreted and object oriented language. It has the following reason for which it used for the research. The perfect language for research was one that allowed concise and direct translation from notepad scribbling to code. On the surface, this is reasonable. Python language use very simple syntax. It increases the readability. Python has a strong point to ease of balancing the high programming language with the low level optimization. In the python language we can write few line of code in comparison of other language. Python excels at gluing other languages together. One can call MATLAB functions from Python (through the MATLAB engine) using mat lab wrap, easing transitions from MATLAB to Python. Python incorporates module, class, function, and method documentation directly into the language itself. In python use two types of comments – programming level comments that are start by # that are not check by the compiler, and documentation comments that are specified by a doc string after the function or method name. These documentation strings add tags to the methods which are accessible

by anyone using an interactive Python shell or by automatic documentation generators.

The beauty of Python's system becomes apparent when using Sphinx, a documentation generation system originally built for the Python language documentation. To allow sufficient presentation flexibility, it allows re structured text directives, a simple, readable markup language that is becoming widely used in code documentation. Python become a most popular language today. Many application, website, game, operating system are developed by using python because python provides various user interface, programing technique, scalability, interactive mode libraries and platforms which are better than the other language.

II. HISTORY

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is famous name not after the constrictor snake but it is famous from British comedy troupe Monty Python's Flying Circus. Python is bourn out from various languages such as ABC, Modula-3, C, C++, Algol-68, Small Talk and Unix shell and other scripting language, but it is bourn out from ABC language. Rossum work on an Amoeba distributed system when Amoeba needed a scripting language, van Rossum developed Python. One of the principle strengths of this new language is thathow to make easy it was to extend, and its support for multiple platforms - a vital innovation in the days of the first personal computers. Python is become capable for communicating with the libraries and modules.

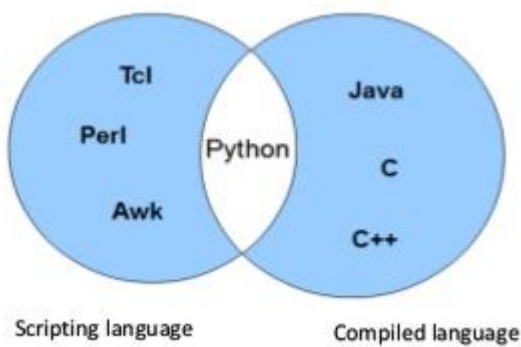


Fig.1 composition of python

Python 2.0 was released by the Be Open Python Labs team in 2000. The ethos of 2.0 was very much more open and community oriented in its development process, with much greater transparency. Python moved its repository to Source Forge, granting write access to its CVS tree more people and an easy way to report bugs and submit patches. As the release notes stated, 'the most important change in Python 2.0 may not be to the code at all, but to how Python is developed'.

Python 2.7 is still used today - and will be supported until 2020. But the word from development is clear - there will be no 2.8. Instead, support remains focused upon 2.7's usurping younger brother - Python 3.

In 2008, Python 3 was released on an almost-unthinkable premise - a complete overhaul of the language, with no backwards compatibility. The decision was controversial, and born in part of the desire to clean house on Python. There was a great emphasis on removing duplicative constructs and modules, to ensure that in Python 3 there was one - and only one - obvious way of doing things. Despite the introduction of tools such as '2to3' that could identify quickly what would need to be changed in Python 2 code to make it work in Python 3, many users stuck with their classic codebases. Even today, there is no assumption that Python programmers will be working with Python 3.

Despite flame wars raging across the Python community, Python 3's future ascendancy was something of inevitability. Python 2 remains a

supported language (for now). But as much as it may still be the default choice of Python, Python 3 is the language's future.

Python's user base is vast and growing - it's not going away any time soon. Utilized by the likes of Nokia, Google, and even NASA for its easy syntax, it looks to have a bright future ahead of it supported by a huge community of OS developers. Its support of multiple programming paradigms, including object-oriented Python programming, functional Python programming, and parallel programming models makes it a highly adaptive choice - and its uptake keeps growing.

III. TECHNIQUE USED IN PYTHON

There are following technique used in python.

A. Modular

Modular programming technique helps to develop the program which are readable, reliable and can be easily maintained without the too many effort. We can use the modular software design technique. Especially if your application program has a limited size. There exists a variety of concepts to design software in modular form. Modular programming is a software design technique in which to break your program into several separate parts. These parts are called modules. The focus for this separation is that a module does not depend upon other modules. In other words: Minimization of dependencies is the goal. When creating a modular system, several modules are built separately and more or less individually. The executable application will be created by placing them together.

B. Object Oriented programming

Python is an object-oriented language. It uses the feature of the object-oriented programming language. In the object-oriented programming everything is a part of an object. Object is an entity that is existed in the real world. The way in which we organizing your program into to the combination of data and functionality and wrap it inside something called an object. This is called the object-oriented programming paradigm. Most of the time we can use procedural programming, but it has many disadvantage when we write the large program. This problem can be solved by using the object-oriented programming, you can use object-oriented programming techniques.

The four major principles of object orientation are:

- Encapsulation
- Data Abstraction
- Polymorphism
- Inheritance

An object oriented program is based on classes. Class contains data and method. In OOP, each object can communicate to each other and receive messages, process data, and send messages to other objects. In the inheritance we can extend the properties of one class into the other class.

C. Interactive programming

Python use the two modes: normal and interactive. In the normal mode the scripted are saved in .py file. These file are run by the python interpreter. In the Interactive mode the command line shell gives immediate response for each statement, while running previously enter statements in active memory. A new line are served into the interpreter, the fed program is calculated both in part and in whole.

D. Structured Programming

Structured programming allows writing a program which is readable and well structured. This helps that you and other are able to reuse this program. Each section of the program is independent, so you can easily select what you most need to learn and concentrate on that.

IV. APPLICATION OF PYTHON

A. GUI based application

Python has very simple syntax. Python used modular approach and have text processing tools which are very useful to develop the GUI based application. Python also provide desirable choice for developing desktop-based applications on the multiple operating systems. There are many GUI toolkits like wxPython, PyQt or PyGtk available which help developers create highly functional Graphical User Interface (GUI). The various applications developed using Python includes:

1) *Image Processing and Graphic Design Applications:* Python is very useful for developing the graphical application. It has been used to make 2D imaging software such as Inkscape,

GIMP, Paint Shop Pro and Scribus. Further, 3D animation packages, like Blender, 3ds Max, Cinema 4D, Houdini, Light wave and Maya, also use Python in variable proportions.

2) *Scientific and Computational Applications:* The python language provides the higher speeds, productivity and availability of tools, such as Scientific Python and Numeric Python, have resulted in Python becoming an integral part of applications involved in computation and processing of scientific data. 3D modelling software, such as Free CAD, and finite element method software, such as Abaqus, is coded in Python.



Fig.2 Application of python

3) *Games:* Python provide the several modules, libraries and platforms that support development of games. Foreexample, PySoy is used to developed the 3D game engine supporting Python 3, and PyGame is used for providing functionality and a library for game development. There have been numerous games built using Python including Civilization-IV, Disney's Toon town Online, Vega Strike etc.

4) *Web Frame work and application:* Python provide variety of web-frameworks such as CherryPy, meet Django, Turbo Gears, Pyramid, Bottle, Flask etc. These frameworks provide standard libraries and modules which simplify tasks related to content management, interaction with database and interfacing with different internet protocols such as HTTP, SMTP, XML-RPC, FTP and POP. Plone, a content management system; ERP5, an open source ERP which is used in aerospace, apparel and banking; Odoo – a consolidated suite of business applications; and Google App engine

are a few of the popular web applications based on Python.

B. Enterprises and Business application

With features that include special libraries, extensibility, scalability and easily readable syntax, Python is a suitable coding language for customizing larger applications. Reddit, which was originally written in Common Lisp, was rewritten in Python in 2005. Python also contributed in a large part to functionality in YouTube.

C. Operating Systems

Python is often an integral part of Linux distributions. Because Ubuntu's Ubiquity Installer, and Fedora's and Red Hat Enterprise Linux's Anaconda Installer are written in Python. Gentoo Linux makes use of Python for Portage, its package management system.

D. Language development

Python's design and module architecture has partial development of numerous languages. Boo language uses an object model, syntax and indentation, similar to Python. Further, syntax of languages like Apple's Swift, Coffee Script, Cobra, and OCaml all share similarity with Python.

E. Prototyping

Besides being quick and easy to learn, Python also has the open source benefit of being free with the support of a large community. This makes it the preferred choice for prototype development. Further, the agility, extensibility and scalability and ease of refactoring code associated with Python allow faster development from primary prototype.

V. LIMITATION OF PYTHON

A. Packet Availability and Maturity

There are many mat lab toolboxes that are not a part of python. There are many packages and modules that are in the under development state and poorly documented. These packages are poorly supported. NumPy, SciPy and matplotlib are very useful package in python for the research, scientific and engineering. The package NumPy and matplotlib are generally well documented and structured,

but SciPy documentation is often unclear or simply absent.

B. Matplotlib Plotting package

Matplotlib is used for non -interactive plotting but there are many problems in this. The default behaviours are not sensitive. There is a problem with the uniformity among the interfaces to the various functions and methods. Python packages follow the computer science method starting counting from zero, but the matplotlib counts subplots start from one.

C. Mathematical optimization

The Scipy library include the package which is used for optimization but there are many issues in this package (scipy.optimize). The scipy do not support for dividing work over multiple nodes in a cluster of computers. The scipy.optimize. Brute solvers allow combining brute force grid search algorithm and second stage optimization but there is no technique for transient termination conditions or other options to the second-stage optimizer.

D. Package and Management

The non-trivial application of python that are part of package do not depends on top level script. Python handle package importation in the suitable manner than the other language but there are issues for the python developers. The issues is that if one application is depend on one version of package and other application depends on other version of package. There is currently no way handle this problem. This problem can be solve if we specify the version of the package that to be import or accept range of package number. But I am not understood that in the practice how this would be applied. It can be hypothetically use a technique that are similar to that employed by the Windows operating system for the registration of DLLs.

E. Named Constants

There is no technique to define the named constant in the python. In the other language such as c++ when we define named constant to instruct to the compiler that any changes made in this generate an error.

F. Support for Microsoft excel

The excel file that containing column organized data .When we work on the file to extract the data from the columns whose numbers are not apriori known. Thexlrd andopenpyxl do not support to identify the wanted columns by strings looking in the header row rather than by numbers.

V. CONCLUSION

In this paper, point that Python is a complete programming solution, I mention three additional points, that Python is a complete programming solution I mention three additional points, each of which would make a great final thought. First, it is open source and completely free, even for commercial use, as are many of the key scientific libraries. Second, it runs natively on Windows, Mac OS, Linux, and others, as does its standard library and the third party libraries I've mentioned here. Third, it fits quick scripting and large development projects equally well. Python is best for research purpose because it provides better technique used in research area.

ACKNOWLEDGEMENT

I thank our colleagues from CCSIT who provided insight and expertise that greatly assisted the review paper, although they may not agree with all of the interpretations/conclusions of this paper. I thank Dr. Ashendra Kumar Saxena for assistance with Python, and Dr. Rakesh Kumar Dwivedi for comments that greatly improved the manuscript.

REFERENCES

- [1] T. Oliphant. Python for scientific computing. *Computing in Science and Engineering*, 9(3):10-20, 2007.
- [2] Mark, Sept. 2009, 'Learning Python', *Beginner's book to python*, 4th edition, O'Reilly Publications, ISBN 978-0-596-15806-4]
- [3] <https://www.python.org/>
- [4] http://phillipmfieldman.org/Python/Python_Limitations.html
- [5] Millman, K. Jarrod; Aivazis, Michael (2011). "Python for Scientists and Engineers" *Computing in Science and Engineering*. **13** (2): 9–12.
- [6] Oliphant, Travis (2007). *"Python for Scientific Computing"*. *Computing in Science and Engineering*
- [7] <https://www.packtpub.com/books/content/brief-history-python>
- [8] <https://www.invensis.net/blog/it/applications-of-python-in-real-world/>
- [9] http://www.python-course.eu/modules_and_modular_programming.php
- [10] <http://docs.huihoo.com/nltk/0.9.5/en/ch05.html>