



Python Programming-Applications and Future

Kalyani Adawadkar¹

¹Head of Department, Department of Information Technology, Sigma Institute of Engineering

Abstract — Python is a high-level and powerful object-oriented programming language created by Guido van Rossum. Because of its simple syntax, it is a very good choice for those who are learning programming for the first time. It is used in vast number of applications due to the various standard libraries that come along with it and its capacity to integrate with other languages and use their features. This paper describes the main features of Python programming. It lists out the difference between Python and other language with the help of some code. This paper then discusses applications of Python programming. To end with we will see a few good examples where python programming is being used.

Keywords-programming, open source, object-oriented, interoperability, sockets

I. INTRODUCTION

Python was conceived in late 1980s, while its implementation began in late 1989 by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands. It was implemented as a successor of ABC language capable of exception handling and interfacing with the operating system Amoeba. Van Rossum is Python's principal author, and his continuing central role in deciding the road to development of Python is reflected in the title given to him by the Python community, benevolent dictator for life (BDFL) [4,5,6]. The End Of Life date (EOL, sunset date) for Python 2.7 was initially set at 2015, then delayed to 2020 out of concern that a large body of existing code cannot easily be forward-ported to Python 3.[7,8]

Python was designed to be highly extensible by its creator. Python can be easily embedded in existing applications which require a programmable interface. It is a small core language with a large standard library and an easily extensible interpreter. An important goal of Python developer is that programmers should have fun using this language.

One of the greatest strength of Python is its large standard library which provides tools suited for many tasks. Various modules for creating graphical user interface, connecting to relational databases, generating pseudorandom numbers, arithmetic calculations, regular expressions etc. are included in the library.

Python's development is conducted largely through the *Python Enhancement Proposal* (PEP) process. The PEP process is the primary mechanism for proposing major new features, for collecting feedback on an issue, and for documenting the design decisions that have gone into Python. Outstanding PEPs are commented and reviewed by the Python Community. [9]

As of March 2017, Python is the fifth most popular language. [10]. A study carried out on Python found it to be more productive than conventional languages for problem solving involving string manipulation and search in a dictionary. The social news networking site Reddit is written entirely in Python. Python has been used in Artificial Intelligence tasks. Python is often used for Natural Language processing tasks. Many Linux distributions use installers written in Python. Python has also extensive use in Information Security Industry. The Raspberry Pi single-board computer project has started using Python as its main user-programming language.

II. FEATURES OF PYTHON PROGRAMMING

Python is a very high-level, dynamic, object-oriented, general purpose programming language that uses interpreter and can be used in a vast domain of applications. Python is very flexible, because of its ability to use modules that were designed in other programming languages. Features of Python are:

➤ **Python is fast and powerful**

It includes an internal standard library that provides all facilities that are needed for programming from the basic operations to the advanced functions.

➤ **Python is simple and lovely**

Readability was the main thought out point when designing the syntax of Python language. The language is very simple for a beginner to understand as compared with other languages C, C++. In C and C++ the learners have to be very careful of the syntax. In Python the learner need to take care of only the indentation. The readability achieved as the result of indentation has number of benefits. The capacity of Python to balance high level programming with low level optimization is the strongest point of Python. One more likable point of Python from coder's point of view is the amount of code you write to solve a particular program. Let us for example write a code which reads 2 files and writes those files one after the other to the 3rd file.

Code in C	Code in Python
<pre>#include<stdio.h> void main() { FILE *fin,*fout; char ch; fin=fopen("a.txt","r"); fout=fopen("c.txt","w"); ch=getc(fin); While(!feof(fin)) { putc(ch,fout); ch=getc(fin); } fclose(fin); fin=fopen("b.txt","r"); ch=getc(fin); While(!feof(fin)) { putc(ch,fout); ch=getc(fin); } fclose(fin) fclose(fout); } fin=open("a.txt","r") x=fin.read() fin.close() fin=open("b.txt","r") y=fin.read() fin.close() s=x+y fout=open("c.txt","w") fout.write(s)</pre>	<pre>fin=open("a.txt","r") x=fin.read() fin.close() fin=open("b.txt","r") y=fin.read() fin.close() s=x+y fout=open("c.txt","w") fout.write(s)</pre>

➤ **Language interoperability**

One of the excellent features of Python is that it glues with other languages. We can call MATLAB functions from Python using MLabWrap. Similarly we can access R, FORTRAN, and C, C++ libraries as well by using RPy, F2pyCtypes, Cython, and SWIG respectively.

➤ **Data Structures**

Python is rich with lists, tuples, sets, dictionaries, strings and many other types built in. We could even add Numbly, Skippy etc. and leverage their benefits. Python has a large and well equipped standard library. It provides tools suited too many tasks. Python is a complete programming solution with greatest advantages namely

1. It is an open source and completely free even for commercial use.
2. It runs natively on Windows, Mac OS, Linux and others, as does its standard library.
3. It fits quick scripting and large development projects equally well.

III. APPLICATIONS OF PYTHON PROGRAMMING

Python is a best choice for real world programming as it is a very well designed language. The most common program types that can be written by Python are as below:

3.1. System Programming:

Python provides internal interfaces for working with services of operating system due to which it is suitable for system programming. These interfaces provide some functions such as: files and directories operations, parallel processing etc. It

contains tools for working with system resources such as: environmental variables, files, sockets, pipe, processes, multiple threads, command line, standard stream interfaces, shell programming etc. [2]

3.2. Graphical User Interface:

Tkinter & GxPython are standard object oriented interfaces that provides essential tools for designing GUI. Generated GUIs give a similar look on different platforms.

3.3. Network and Internet Programming:

Many tools are provided for network programmers namely client-server connection, socket programming, FTP, Telnet, email functions, RPC, SOAP etc. Also some third party tools like mod-Python allow servers like apache to run python scripts.

3.4. Component integrity:

Tools like Swing and SIP can import the compiled codes of other languages for using in Python.

3.5. Database Programming:

Python interfaces support common databases like Sybase, Oracle, Informix, MYSQL, PostgreSQL, SQLite etc.

3.6. Numerical programming:

Python with NumPy module provides powerful tools for working with mathematical libraries, by using simple Python codes.

3.7. Other Programming applications:

- PyGame is a tool for Game Programming
- PIL is used for image processing
- PyRo for Robotic Programming
- NLTK is a package for artificial intelligence, network simulation and shell programming.
- Python is a free open source program and consequently it causes more popularity among users.

IV. LIST OF CORPORATIONS USING PYTHON FOR DIFFERENT FUNCTIONS

Nowadays many corporations use this tool for different functions. Some of them are as below:

- Google is one of the Python users that included this language in its web search system and Employed Python's creator, too.
- YouTube video sharing service makes extensive use of Python.
- Popular BitTorrent peer-to-peer file sharing system is written by Python.
- ESRI uses Python as an end-user customization tool for its popular GIS mapping products.
- NASA, Los Alamos, Fermilab, JPL, and others use Python for scientific programming tasks.
- IRobot uses Python to develop commercial robotic vacuum cleaners.
- Intel, Cisco, Hewlett-Packard, Seagate, Qualcomm, and IBM use Python for hardware testing.
- NSA uses Python for cryptography and intelligence analysis.
- IronPort email server product uses more than 1 million lines of Python code to do its job.
- One Laptop per Child (OLPC) project builds its user interface and activity model in Python.
- Industrial Light & Magic, Pixar, and others use Python in the production of movie animation.
- JPMorgan Chase, UBS, Getco, and Citadel apply Python for financial market forecasting.[3]

IV. CONCLUSION

We have seen various features of Python Programming. We have seen its applications and a list of corporations using Python tools. It is a very vast language and has got various facets which can be researched upon. Few of the topics in which Python's application can be researched are Information Security, Artificial Intelligence, and Big Data Analytics etc.

REFERENCES

- [1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," *Phil. Trans. Roy. Soc. London*, vol. A247, pp. 529–551, April 1955. (references)
- [2] J. Clerk Maxwell, *A Treatise on Electricity and Magnetism*, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
- [3] Python for Education: Permutations Andrzej Kapanowski Marian Smoluchowski Institute of Physics Jagiellonian University, Cracow, Poland
- [4] "History and License". Retrieved 5 December 2016. "All Python releases are Open Source"
- [5] Venners, Bill (13 January 2003). "The Making of Python". *Artima Developer*. Artima. Retrieved 22 March 2007
- [6] van Rossum, Guido (20 January 2009). "A Brief Timeline of Python". *The History of Python*. Google. Retrieved 20 January 2009.
- [7] "PEP 373 -- Python 2.7 Release Schedule". *python.org*. Retrieved 9 January 2017.
- [8] "PEP 466 -- Network Security Enhancements for Python 2.7.x". *python.org*. Retrieved 9 January 2017.

- [9] Warsaw, Barry; Hylton, Jeremy; Goodger, David (13 June 2000). "PEP 1 – PEP Purpose and Guidelines". Python Enhancement Proposals. Python Software Foundation. *Retrieved 19 April 2011*.
- [10] "TIOBE Index". *TIOBE - The Software Quality Company*. Retrieved 7 March 2017.
- [11] "Quotes about Python". *Python Software Foundation*. Retrieved 8 January 2012.
- [12] "Python for Artificial Intelligence". *Wiki.python.org*. 19 July 2012. *Archived from the original on 1 November 2012*. Retrieved 3 December 2012.
- [13] Paine, Jocelyn, ed. (August 2005). "AI in Python". *AI Expert Newsletter*. Amzi!. Retrieved 11 February 2012.
- [14] "PyAIML 0.8.5 : Python Package Index". *Pypi.python.org*. Retrieved 17 July 2013.
- [15] Russell, Stuart J. & Norvig, Peter (2009). *Artificial Intelligence: A Modern Approach (3rd ed.)*. Upper Saddle River, NJ: Prentice Hall. p. 1062. ISBN 978-0-13-604259-4. Retrieved 11 February 2012.
- [16] "Natural Language Toolkit"
- [17] "Immunity: Knowing You're Secure"
- [18] "Corelabs site"