Python is currently on a Hot Streak in Programming Language

Mr. Praveen S. I.[1] Mr. Puneeth N. T.[2] Mrs. Jayasheela D. K.[3] Mrs. Radhika P. A.[4] Miss. Vani S. Itagi[5]
[1]M. Sc. Student [2,3,4]Assistant Professor [5]Programmer
[1]Department of Computer Science
[1]S. M. E. Abdul Kalam College of M. Sc. (CS) PG College, Gadag, India [2,3,4]K L E Institute of Technology Hubballi, India [5]REC Engineering College Hulkoti, India

Abstract— This paper, we are going to represent an introduction to Python programming language and prove Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985 – 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). Python is named after a TV Show called 'Monty Python's Flying Circus' and not after Python-the snake. Python 3.0 was released in 2008. Although this version is supposed to be backward incompatibles, later on many of its important features have been back ported to be compatible with the version 2.7. This gives enough understanding on Python 3 version programming language.

Sample Code: #!/usr/bin/python3
print ("Hello, Python!")

Key words: Python Programming Language, Python is Hot Streak in Programming Language

I. Introduction

Python is a powerful modern computer programming language. It bears some similarities to Fortran, one of the earliest programming languages, but it is much more powerful than Fortran. Python allows you to use variables without declaring them (i.e., it determines types implicitly), and it relies on indentation as a control structure. You are not forced to define classes in Python (unlike Java) but you are free to do so when convenient.

Python was developed by Guido van Rossum, and it is free software. Free as in "free beer," in that you can obtain Python without spending any money. But Python is also free in other important ways, for example you are free to copy it as many times as you like, and free to study the source code, and make changes to it. There is a worldwide movement behind the idea of free software, initiated in 1983 by Richard Stallman. [1]

This document focuses on learning Python for the purpose of doing mathematical calculations. We assume the reader has some knowledge of basic mathematics, but we try not to assume any previous exposure to computer programming, although some such exposure would certainly be helpful. Python is a good choice for mathematical calculations, since we can write code quickly, test it easily, and its syntax is similar to the way mathematical ideas are expressed in the mathematical literature. By learning Python you will also be learning a major tool used by many web developers. [1]

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, http://www.python.org/, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation. [2]

- Little-known standard library modules collections, contextlib, and concurrent. futures, logging, & sched
- Flit for simplifying the process of submitting a Python package to the Python Package Index (PyPI)
- Colorama and begins for making your command-line applications friendlier for users
- Pyqtgraph and pywebview for creating graphical user interfaces (GUIs)
- Watchdog, psutil, and ptpython for working closely with the operating system
- Hug for exposing APIs for other users' programs to consume
- Arrow and parsedatetime for working with dates and times
- Third-party general-purpose libraries: Boltons, Cython, and the awesome-python curated list.[3]

A. NumPy/SciPy

This pair of libraries provide array and matrix structures, linear algebra routines, numerical optimization, random number generation, statistics routines, differential equation modeling, Fourier transforms and signal processing, image processing, sparse and masked arrays, spatial computation, and numerous other mathematical routines. Together, they cover most of MATLAB's basic functionality and parts of many of the toolkits, and include support for reading and writing MATLAB files. Additionally, they now have great documentation (vastly improved from a few years ago) and a very active community. t IPython: One of the best things in Python is IPython, an enhanced interactive Python shell that makes debugging, profiling code, interactive plotting. It supports tab completion on objects, integrated debugging, module finding, and more — essentially; it does almost everything you'd expect a command line programming interface to do. Additionally, Cython: Referenced earlier, Cython is a painless way of embedding compiled, optimized bits of code in a larger Python program. It SQLAlchemy: SQLAlchemy makes leveraging the power of a database incredibly simple and intuitive. It is essentially a wrapper around an SQL database. You build queries using intuitive operators, and then it generates the SQL, queries the database, and returns an iterator over the results. Combining it with sqlite — embedded in Python's standard library — allows one to leverage databases for scientific work with impressive ease. And, if you tell sqlite to build its database in memory, you've got another powerful data structure. To slightly plagiarize xkcd, SQLAlchemy makes databases fun again. PyTables: PyTables is a great way of managing large amounts of data in an organized, reliable, and efficient fashion. It optimizes resources, automatically transferring data between disk and memory as needed. It also supports on-the-fly (DE) compression and works seamlessly with NumPy arrays. t PyQt: For writing user interfaces in C++, I recommend it is, in my experience, difficult to beat QT. PyQt brings the ease of QT to Python. And I do mean ease — using the interactive QT designer, I've build a reasonably complex GUI-driven scientific application with only a few dozen lines of custom GUI code. The 28 PROGRAMMING entire things was done in a few days. The code is cross-platform over Linux, Mac OS X, and Windows. If you need to develop a front end to your data framework, and don't mind the license (GPL for PyQT, LGPL for QT), this is, in my experience, the easiest way to do so. [4]

II. Advantages & Disadvantages

With the capabilities of running on most operating system platforms from Windows to Linux to OS X, Python is a language with very few boundaries. A program written in Python, for one platform, using only the standard libraries can easily be ported to another operating system without need for recompiling or repackaging. Python is even capable of operating on several types of Nokia cell phones.

Python does not enforce a strict type on containers or variables. With this concept, developers can design a container to hold different types of data (integer, string, float etc.) A single container for different data types means less memory is used to create distinct containers for different data types. Along with less memory usage, the user can implement the same functions on different data types rather than defining a distinct function for each type. [5]

III. How to be Secure?

All the things related to security described are serious issues and need to be addressed. To make our computer systems more secure we need to make the software that it runs, more reliable and free of errors and bugs.

A. Open Source Security

Though it is a debate, it is now being widely accepted that instead of keeping the code secret, making it available to everyone for review and changes helps in making it more secure. While it might be hard to perceive this at first, consider the case of scientific research. Research produced by scientists is reviewed by peers for any errors and flaws, corrections are made and the same iterative process is followed before it is acknowledged or is applied in the industry. Same is the case with security of the software. [6]

Python is a multi-paradigm programming language: object-oriented programming and structured programming are fully supported, and there are a number of language features which support functional programming and aspect-oriented programming. [7]

The core philosophy is includes aphorisms such as: [8]

- Beautiful is better than ugly
- Explicit is better than implicit
- Simple is better than complex
- Complex is better than complicated
- Readability counts

Rather than requiring all desired functionality to be built into the language's core, Python was designed to be highly extensible.

Python has a large standard library, commonly cited as one of Python's greatest strengths, providing tools suited to many tasks. This is deliberate and has been described as a "batteries included" Python philosophy.

IV. Some Kinds of Programs?

Python is a well-designed language that can be used for real world programming. The most common program is types that can be written by Python.

- System Programming
- Graphical User Interface (GUI)
- Network & Internet Programming
- Components Integrity
- Database Programming
- Numerical Programming

V. Complements of Python

Python has some alternatives that might be useful in different conditions. For example, by using Jython instead of Python, you can compile the source codes to Java Byte Code (JBC) and it can be run on the Java Virtual Machine systems including PCs, cell phones, etc. This alternative can use Java libraries, too. Another alternative is IronPython. It has the similar approach of Jython but on .Net technology. In fact it translates the source code to Microsoft Intermediate Language (MSIL), like what .Net programming languages do.Other alternative is ActivePython, which is a complete package that includes IDE and CPython (a version of Python interpreter).Also, many complements are distributed for Python. PyPy is one of them that can generate different outputs like JBC, MSIL and object code. Other complements like Pyjamas and ShedSkin translate the source code to Java scripts and C++ source codes. Also, Cython and Pyrex generate C codes. Two compilers that output executive files are Py2exe and Pyinstaller for Windows and Linux. Nokia released a Python interpreter with PyS60 name in 2005. It contained standard python modules and some more for working with Symbian OS on Nokia 60 series. For Windows CE, Python CE was distributed, too.

VI. Python Users

Python is a free open source program and consequently it causes more popularity among users. Nowadays many corporations use this tool for different functions.

- YouTube video sharing service makes extensive use of Python.
- Popular Bit Torrent 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, Fermi lab, 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.
- Iron Port 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.

VII. Python Implementations

The main Python implementation is written in the C language and is called CPython. It is the one that majority of people refer to, when they talk about Python. When the language evolves, the C implementation is changed accordingly. Besides C, Python is available in a few other implementations that are trying to keep up with the mainstream. Most of them are a few milestones behind CPython, but provide a great opportunity to use and promote the language in a specific environment. [10]

A. Jython

Jython is a Java implementation of the language. It compiles the code into Java byte code, and allows the developers to seamlessly use Java classes within their Python modules. (In Python, a file containing code is called a module.) Jython allows people to use Python as the top-level scripting language on complex application systems, for example J2EE. It also brings Java applications into Python applications. [11]

B. IronPython

IronPython brings Python into .NET. The project is supported by Microsoft, where IronPython's lead developers work. The latest stable version is 1.1 (released in April 2007) and implements Python 2.4.3. It is available in ASP.NET, and let's people use the Python code in their .NET application in the same way as Jython does in Java. It is quite an important implementation for the promotion of a language. [12]

C. PyPy

PyPy is probably the most exciting implementation, as its goal is to rewrite Python into Python. In PyPy, the Python interpreter is itself written in Python. We have a C code layer carrying out the nuts-and-bolts work for the CPython implementation of Python. But in the PyPy implementation, this C code layer is rewritten in pure Python. [13]

VIII. Future of Python

According to the TIOBE index, Python is the 4th most popular programming language out of 100 With the rise of Ruby on Rails and more recently Node.js, Python's usage as the main prototyping language for backend web development has diminished somewhat, especially since it has a fragmented MVC ecosystem. However, with big data becoming more and more important, Python has become a skill that is more in demand than ever especially it can be integrated into web applications. As an open source project, Python is actively worked on with a moderate update cycle, pushing out new versions every year or so to make sure it remains relevant. In terms of search volume for anyone interested in learning Python, it has skyrocketed to the 1st place when compared to other languages. [14]

IX. Conclusion

We already discussed in two software development researcher with respect to keeping them secure and not prone to exploits. Each of these has its own merits and demerits. However, as we have seen, open source paradigm has far more benefits than the secret code paradigm which may allow a company to leave a hole or backdoor for later exploit The open source model provides many benefits in the form of free software along with transfer of technology that is source code for our review and modification which should result in better security for our Information Technology infrastructure.

References

[1] Python Basis – S. R. Doty 27th August, 2008.
[2] Python Beginners by Guido Van Rossum and Fred L. Drake.
[3] Python Libraries by Ca;eb Hatttingh.
[4] Beginning Python, Advanced Python Exercise by Dave Kunlman.
[5] Why Python Rocks for Research by Hoyt Koepke.
[6] Love Letter to Python by Mike Levin's 5th January, 2011.
[7] White lock, Natalie, "The security implication of open source software" IBM – 2001.
[8] The Cain Gang Limited, "Python Meta classes" – 10th December, 2009.
[9] Peters, Tim Python Software Foundation – 24th November, 2008.
[10] Moderator of the San Francisco Bay Area Python Interest Group by Packet Publishing Ltd.
[11] Vesa-Matti Hartikainen – 12th May, 2006.
[12] Manning Publications Co. by Michael J. Foord – 2009.
[13] Vesa-Matti Hartikainen – 12th May, 2006.
[14] "The RedMonk Programming Language" Rankings: January 2011 – tecosystem.