Introduction to Python Programming Language

- Python is a high-level, interpreted, interactive and object-oriented scripting language.
- Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.
- It has simple easy-to-use syntax.
- It is a versatile scripting language which makes it attractive for Application Development.
- Python supports both object oriented programming and procedure oriented programming paradigm.
- We don't need to use data types to declare variable because it is dynamically typed.
- Python makes the development and debugging faster because there is no compilation step included in python development.
- Python is interpreted: Python is processed at runtime by the interpreter. We do not need to compile our program before executing it.
- Python is interactive: we can interact with the interpreter directly to write our program.
- Python is a beginner's language: Python is a great language for the beginner-level programmers.

History of python

- 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 derived from many other programming languages including ABC, Modula-3, C, C++, algol-68, SmallTalk, and Unix shell and other programming languages.
- Python is copyrighted. Like Perl, Python source code is now available under the GNU General public License (GPl), hence called open source language.

- Many of the python’s features originated from an interpreted language called ABC.
- Python actually got this name from BBC comedy series from the seventies “Monty Python’s Flying Circus”.

Python Features:

1. **Easy-to-learn:** Python has few keywords, simple structure, and a clearly defined syntax.
2. **Simple:** Python is a simple language. The pseudo-code nature of python is one of its greatest strengths. It allows learner to concentrate on solution to the problem rather than the syntax.
3. **Easy-to-Read:** Python code is more clearly defined and easy to understand.
4. **Easy-to-maintain:** python source code is easy to maintain.
5. **A broad standard library:** python’s bulk of library is very portable and cross-platform compatible on UNIX, Windows and Machintosh.
6. **Interactive Mode:** Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
7. **Portable:** python can run on a wide variety of hardware platforms and has the same interface on all platforms.
8. **Databases:** Python provides interfaces to all major commercial databases.
9. **GUI Programming:** python supports GUI applications that can be created and ported to many system calls, libraries, and windows systems.
10. **Free and Open source:** python is an example of FOSS(Free and open source software) i.e. we can freely distribute copies of his software, read the software’s source code, make changes to it, use pieces of it in new free programs.
11. **Interpreted:** a program written in compiled language like C or C++ is translated from the source language. i.e C/C++ into a language spoken by computer (binary code) using compilers. When we run the program, the linker/loader software just stores the binary code in the computer’s memory and starts executing from the first instruction in the program. When we use an interpreted language like python, there is no separate compilation and execution steps. We just run the program from the source code. Internally

python converts source code into an intermediate code called byte codes and then translates into native language of specific computer and then runs it.

12. **Embeddable:** we can embed Python within C/C++ program to give scripting capabilities for users.
13. **Extensive Libraries:** the python standard library is huge indeed.

Debugging:

- Debugging is the routine process of locating and removing computer program bugs, errors or abnormalities.
- Debugging checks, detects and corrects errors r bugs to allow proper program operation according o set specifications.
- In the debugging process complete software program are regularly compiled and executed to identify and rectify issues.
- For efficiency, each component is debugged separately at first, followed by the program as a whole.

Errors:

- Bug in a program is called as "Error".
- Types of Errors:
  1. Syntax Errors:
     - These are the most basic type of errors.
     - They arise when python interpreter is unable to understand a line of code.
     - Syntax errors are almost always fatal i.e. a piece of code containing syntax errors can't be executed successfully.
     - In IDLE, it will highlight where the syntax error is.
     - Most syntax errors are incorrect indentation or incorrect arguments.
  2. Semantic Errors:
     - Semantic or logic errors are problems with the design of the program.
     - These usually do not produce any error message but instead cause program to behave incorrectly.

- These errors are often caused by accidentally using one variable in place where a different variable is intended, or by simply doing some math incorrectly.