

Introduction of python

Rahee d sutrave.

Information about python:

- Python is a very popular general-purpose interpreted, interactive, object-oriented, and high-level programming language.
- Python is dynamically-typed and garbage-collected 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 consistently rated as one of the world's most popular programming languages.

There are many other good reasons which makes Python as the top choice of any programmer:

- Python is Open Source which means its available free of cost.
- Python is simple and so easy to learn.
- Python is versatile and can be used to create many different things.
- Python has powerful development libraries include AI, ML etc.
- Python is much in demand and ensures high salary

Key features of python:

1. Easy to Code: Python is a very high-level programming language, yet it is effortless to learn.
2. Easy to Read: Python code looks like simple English words. There is no use of semicolons or brackets, and the indentations define the code block.
3. Free and Open-Source: Python is developed under an OSI-approved open source license.
4. Robust Standard Library: Python has an extensive standard library available for anyone to use.
5. Interpreted: When a programming language is interpreted, it means that the source code is executed line by line, and not all at once.
6. Portable: Python is portable in the sense that the same code can be used on different machines.
7. Object-Oriented and Procedure-Oriented: A programming language is object-oriented if it focuses design around data and objects, rather than functions and logic.
8. Extensible: A programming language is said to be extensible if it can be extended to other languages.
9. Expressive: Python needs to use only a few lines of code to perform complex tasks.

Type of language python is:

Python as scripting language:

- Python is scripting, general-purpose, high-level, and interpreted programming language. It also provides the object-oriented programming approach. The filename extension of Python can be various types such as .py .pyw. .pyc, .pyd, .pyz.
- The scripting language is referred to perform the task based on automating a repeated task. It includes same types of steps while implementing the procedure or program. It reduces time and cuts the costs further. The scripting languages are interpreted language instead of a compiled language.

Python as programming language:

- Python is widely used in programming and is an interpreted language which occurs during runtime. Python is a famed programming language ,since it offers high-level programming,
- it's vastly popular for developers and high-level programmers everywhere. Python is popular because it uses simplistic code and clear syntax and that matters because even a beginner can understand it. Learning Python can be a smart move because it is widely used in development and programming.

About PEP 8:

- PEP is Python Enhancement Proposal.
- PEP-8, is a document that provides guidelines and best practices on how to write Python code.
- It was written in 2001 by Guido van Rossum, Barry Warsaw, and Nick Coghlan. The primary focus of PEP 8 is to improve the readability and consistency of Python code.
- A PEP is a document that describes new features proposed for Python and documents aspects of Python, like design and style, for the community.

Python as interpreted language:

Python is an interpreted language, which means the source code of a Python program is converted into bytecode that is then executed by the Python virtual machine. Python is different from major compiled languages, such as C and C + +, as Python code is not required to be built and linked like code for these languages. This distinction makes for two important points:

- Python code is fast to develop: As the code is not needed to be compiled and built, Python code can be readily changed and executed. This makes for a fast development cycle.
- Python code is not as fast in execution: Since the code is not directly compiled and executed and an additional layer of the Python virtual machine is responsible for execution, Python code runs a little slow as compared to conventional languages like C, C + +, etc.

Memory storage management in python:

- Memory management in Python involves a private heap containing all Python objects and data structures. The management of this private heap is ensured internally by the Python memory manager.
- The Python memory manager has different components which deal with various dynamic storage management aspects, like sharing, segmentation, pre allocation or caching.
- At the lowest level, a raw memory allocator ensures that there is enough room in the private heap for storing all Python-related data by interacting with the memory manager of the operating system.
- On top of the raw memory allocator, several object-specific allocators operate on the same heap and implement distinct memory management policies adapted to the peculiarities of every object type.
- The management of the Python heap is performed by the interpreter itself and that the user has no control over it.
- The allocation of heap space for Python objects and other internal buffers is performed on demand by the Python memory manager through the Python/C API functions listed in this document.
- The Python memory manager thus delegates some of the work to the object-specific allocators, but ensures that the latter operate within the bounds of the private heap.

Namespace in python:

- A namespace is a collection of currently defined symbolic names along with information about the object that each name references.
- The namespace as a dictionary in which the keys are the object names and the values are the objects themselves.
- Each key-value pair maps a name to its corresponding object.
- As Python executes a program, it creates namespaces as necessary and deletes them when they're no longer needed. Typically, many namespaces will exist at any given time.

In a Python program, there are four types of namespaces:

- 1.Built-In
- 2.Global
- 3.Enclosing
- 4.Local

1.Built-In namespaces:

The built-in namespace contains the names of all of Python's built-in objects. These are available at all times when Python is running.

Ex: Arithmetic Error, Assertion Error, Attribute Error, Base Exception ,max(), len()

2. Global Namespaces:

- The global namespace contains any names defined at the level of the main program. Python creates the global namespace when the main program body starts, and it remains in existence until the interpreter terminates.
- Ex: It creates a global namespace for any module that your program loads with the import statement .

3. Enclosing and Local Namespaces:

- The interpreter creates a new namespace whenever a function executes. That namespace is local to the function and remains in existence until the function terminates.
- Ex: User defined main functions are enclosing functions. While subfunctions in main functions are local functions.