INTRODUCTION OF PYTHON

Python is a high-level, interpreted programming language, it is designed to be easy to read and write with simple syntax.

Python is an open-source language, meaning that it is free to use and distribute, and it has a large and active community of developers who contribute to its development and use it for a wide range of applications.

History of Python

Python was created in the late 1980s by Guido van Rossum a Dutch programmer. He was the BDFL (Benevolent Dictator For Life) of python.

He was working at the National Research Institute for Mathematics and Computer Science in the Netherlands.

The first version of Python, Python 0.9.0, was released in February 1991. It was a simple language with a small set of features, but it quickly gained popularity among developers for its ease of use and readability.

In 1994, Python 1.0 was released, which added features like map, filter, and reduce, as well as support for modules and exceptions. The release of Python 1.5 in 1997 added support for garbage collection and introduced the "import" statement.

Over the years, Python continued to evolve and add new features. Python 2.0, released in 2000, Python 3.0, released in 2008, made significant changes to the language, including, removing some features and introducing new syntax and features.

Current Version - 3.11.3
For reference/Documentation/Downloads - https://www.python.org

Today, Python is one of the most popular programming languages in the world, and is widely used in industries such as web development, data science, machine learning, and more.

Its simplicity, ease of use, and large ecosystem of libraries and frameworks have made it a favorite among developers of all skill levels.

FEATURES OF PYTHON LANGUAGE

- High-level Language
- Easy to learn, easy to read and write
- Free and open source
- Dynamic Typing - means no need to declare a variable or its type before using them.
- Duck Typing
- Garbage Collection - has automatic memory management
- Functional programming
- support OOPs concepts
- Scalable - ability to handle large projects and datasets.
- GUI Programming - Tkinter, JPython, wxPython, etc.,
- Interactive Mode Vs Scripting Mode.

COLOR CODING SCHEMES OF PYTHON

Color		Description
Green	-	Unicode string values
Blue	-	Python processed return value
Orange	-	Keywords
Purple	-	Built in functions
Red	-	Error and exception
Black	-	Expression

FLAVORS OF PYTHON

- Cpython (or) cython - python + C programming
- Jpython (or) jython - Python + java + JVM
- IronPython - Python + Dotnet + MVC
- RubyPython - Python + Ruby
- APython - Anaconda + Python + Big data
- Pyston - Python + JIT Interface for Dropbox
- Nuitka - Transpiler which converts python to executable codes
- Numba - Python + Numpy programs to low level codes
- Brython - creating browsers in PY Environment

ZEN OF PYTHON

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Flat is better than nested.
- Readability counts.
- Special cases aren't special enough to break the rules.
- Although practicality beats purity.
- Errors should never pass silently.
- Now is better than never.
- Although never is often better than *right* now.
- If the implementation is hard to explain, it's a bad idea.
- If the implementation is easy to explain, it may be a good idea.