

1. What is programming.

To train the system to perform and particular task we use programming.

Programming is the process of writing instructions that a computer can understand and execute to perform specific tasks.

There are 3 level of programming :-

1. Low level programming / Binary programming :-

Low level programming refers to coding that is close to the hardware, where the computer programmer manages memory, CPU registers and instructions directly.

Ex :- \* Assembly language

\* C language

Features :-

- Gives high control over system resources
- Code runs very fast and efficiently.

Binary programming :-

Binary programming means working directly with machine code - the 1s and 0s that a CPU actually executes.

Ex :- 10110000 01100001

- It's the lowest possible level of programming
- Extremely hard for humans to write or understand.
- Usually generated by Compilers or assemblers from higher-level code.

② Assembly level programming :-

Assembly language is a low-level programming language that provides a near one-to-one correspondence with a computer's machine code, allowing programming to write instructions that interact directly with processor registers, memory, and I/O. It translates the mnemonics into executable machine code.

features :-

- Key features of assembly-level programming includes it's status as a low-level language.
- It provides direct, granular control over hardware resources, enabling precise management of instructions.

### ③ High-level programming :-

High level programming refers to languages and environments that simplify software development by abstracting away hardware details, enable developers to write logic using expressive constructs rather than managing low-level operations.

features :-

- Abstraction from hardware :- code is written in data structures and constructs like loops and functions, not memory addresses or processor instructions.
- Automatic memory management :- Many high-level languages handle memory allocation and garbage collection, freeing developers from manual memory management. This lowers the risk of leaks and errors.



## ① What is python and history of python.

### Introduction:-

Python is a versatile high-level programming language known for its simple syntax and readability. Key basics include its use as a calculator, the ability to write code in files, dynamic typing where variable types are determined at runtime, & significant indentation to define code blocks. It is widely used for web development, data science, AI and automation.

### Features:-

#### ① Easy to learn and use:-

Python's syntax is designed for readability often using English keywords and requiring less code compared to other languages for similar tasks. This makes it beginner-friendly and promotes faster development.

#### ② Platform Independent / Portable:-

Python code can run on various operating systems like windows, mac os and linux without modification. This is achieved by converting the code into an intermediate bytecode, which is then executed by the python virtual machine on the target platform.

#### ③ Libraries and packages:-

Python comes with a comprehensive standard library offering modules for diverse tasks. Additionally, a vast ecosystem of third-party libraries and frameworks extends



python's capabilities for areas like web development, data science, machine learning.

④ open source:-

python is freely available and can be used, modified, and distributed by anyone. This open nature has fostered a large and active community that contributes to its improvement and extensive ecosystem.

⑤ scalable:-

python can be used for a wide range of applications, from small scripts and the automation tasks to large-scale enterprise applications and complex data processing.

⑥ GUI programming support:-

python provides libraries like Tkinter, Pyqt, and Kivy for developing graphical user interfaces.

Applications of python:-

① web development:-

web development is the one of the most popular applications of python. with the availability of various python frameworks, web development has become easier.

These frameworks come with various libraries and modules that simplify web development task, including interfacing with the internet protocols like JSON, HTTP, XML, FTP, IMAP etc...

② Mobile development:-

It is a process of creating software



for mobile devices, like smartphones and tablets, primarily for the iOS and android operating systems. This can involve building native apps for a single platform, or cross-platform apps that work on both. The process includes coding, implementing backend services and testing the application on target devices.

### ③ Gaming development:-

With the gaming industry reaching new heights of success, many software development companies use python for game development. As mentioned earlier, python has many libraries that come in handy while development video games. It is the foundation for many notable video games, includes bridge commander, pirates of the Caribbean.

### ④ Desktop GUI:-

Building desktop applications of python. python has a set of libraries and toolkits to create a graphical user interface for desktop applications. These libraries help development to create a fully-functional GUI effortlessly and efficiently.

### ⑤ Data science and AI:-

pandas, Numpy, Scipy, scikit learn, Tensorflow, PyTorch and Jax power data analysis, modeling and machine learning workflows widely across industries.



Python was created by Guido van Rossum in the late 1980s in the Netherlands and first released to the public in 1991. It was conceived as a successor to the ABC language, with the goal of combining readability, simplicity and power for general-purpose programming. The name "python" was inspired by Guido's fondness for Monty Python's flying circus rather than any reptilian association.

② What are the pros and cons of python?

\* Expanded pros and python :-

1. Readable and Maintainable code
2. Extensive libraries and frameworks
3. Platform Independence
4. Strong Community and Documentation
5. Integration capabilities
6. Ideal for Automation and scripting
7. Versatile Application

• Readable pros Maintainable code :-

\* Python emphasizes code readability with its clean system and indentation rules.

\* Easier collaboration among team faster debugging.

• Extensive libraries and framework :-

\* Rich ecosystem with libraries like :-

\* Expanded Cons of python :-

• Slower execution speed :-

Interpreted language, so it's generally

Slower than complicated languages like C++ (or) Java.

\* [NOT memory Java NOT the best choice] :-

\* NOT ideas for mobile apps :-

Limited support and performance for mobile development.

\* High memory usage :-

NOT the best choice for memory-intensive tasks.

\* Weak in multi-threading :-

Due to the Global Interpreter Lock (GIL), true parallelism is limited.

\* Runtime errors :-

Dynamic typing can lead to bugs that only appear at runtime.