(Q). What are the pros and cons of Python?

(A). Pros of Python.

* Easy to learn and Read
  - Python use simple, readable syntax close to natural language.
  - Great for beginners and rapid Prototyping.

2. large Standard Library and Ecosystem
- comes with many built-in modules (for file handling, web servers, etc).
- Huge number of third-party packages (Numpy, Pandas, Django, Tensorflow, etc).

3. Cross platform compatibility.
* Runs on windows, macos, and linux without major changes.

4. versatile and multi-purpose
* Used in Web Development, data science, AI/ML, automation, CyberSecurity, IoT, etc.

5. Strong community support.
* one of the largest developer communities.
* Abundant tutorials, Documentation and Q&A support (eg. stack overflow)

6. Integration Capabilities.
* Easily integrates with other languages (C, C++, Java) and databases.
* Useful for scripting and automation of repetitive tasks.

7. Extensive frameworks
- Django and flask (web apps), Python Pytorch and Tensorflow (AI), Pandas (data), etc.

Cons of Python

1. Slower execution speed.
- Being an interpreted language, it's slower than C, C++, or Java.
* Not ideal for performance-critical applications.

2. High memory Usage -
* Python's dynamic typing and data structures consume more memory.
* Not suitable for low-memory devices or mobile app backends.
3. Weak in mobile and Game Development -
* Fewer frameworks for mobile development.
* Most mobile apps are built using Swift, Kotlin, or Flutter instead.
4. Runtime Errors -
* Since it's dynamically typed, errors often appear only during execution.
5. Not ideal for Multi-threading -
* The Global Interpreter Lock (GIL) prevents true parallel execution of threads.
* Limits performance in CPU-bound tasks.
6. Dependency Management Issues -
* Virtual environments help, but managing multiple versions of libraries can be tricky.

Category	Pros	Cons
Ease of Use	Simple syntax, beginner friendly	Indentation errors for newcomers
Performance	Fast development	Slower Runtime
Applications	versatile (AI, Web, Data)	Weak in mobile & embedded system.
Community	Huge global support	many outdated libraries still exist
Maintenance	Readable & Clean Code	Dynamic typing can cause runtime bugs

collection and Unicode support

Python 3.0	2008	major overhaul - not backward compatible, improved unicode, print as a function, better division rules.
Python 3.6-3.9	2016-2020	f-string, async/await, type hints, improved performance.
Python 3.10+	2021 Present	Structural pattern matching (match statement), enhanced error messages, better type checking.

4. Python 2 vs Python 3 -

- Python 2 was used for nearly 20 years but officially retired on January 1, 2020.
- Python 3 is now the only supported version, with better performance, clarity, and modern syntax.

5. Community & Growth

- Python is maintained by the Python Software Foundation (PSF).
- It's open source and community-driven.
- Became one of the most popular languages worldwide due to its versatility in:
  - Data Science
  - Machine learning
  - Web Development
  - Automation
  - CyberSecurity

6 Present and Future

* Python continues to evolve with a focus on-
* Speed optimization (eg- Python 3.11 and later are significantly faster).
* Type safety (through gradual typing)
* AI/ML integrations.

As of 2025, Python remains among the top 3 most used programming languages globally they are

Alongside
Javascript
Java.

② History of PYTHON?

A. Overview of history of Python.

1. Origin and Creation

* Creator - GUIDO VAN ROSSUM, a Dutch programmer
* Created in - late 1980's
* First released - February 20, 1991.
* Place of creation - Centrum Wiskunde & Informatica (CWI) Netherlands.

Guido wanted a language that was.

* Easy to read & learn
* Powerful enough for system administration and scripting tasks.
* An improvement over the ABC language during christmas 1989 as a ~~hobby project~~ (a teaching language at CWI)

So, he started designing Python during christmas 1989 as a hobby project.

2. Name origin -

* The name 'PYTHON' was inspired not by the snake, but by the British comedy show "MONTY PYTHON'S FLYING CIRCUS".
* Guido wanted the language to be fun and approachable, not overly formal.

3 Major version Timelines

version	year	key features
Python 1.0	1991	First official release; included exceptions, functions, and core data types like str, list, dict.
Python 2.0	2000	Introduced list comprehensions, garbage