



Workshop on Python

By Robotic Society of RUET





Nazib Abrar

- Researcher, Robotics & Computer Vision AI
- Junior Software Developer, FronTech Limited
- Assistant Technical Coordinator, RSR
- Student, Mechatronics Engineering, RUET

 in/astrar-nazib

TABLE OF CONTENTS

01

INTRODUCTION
To Computer
Architecture

02

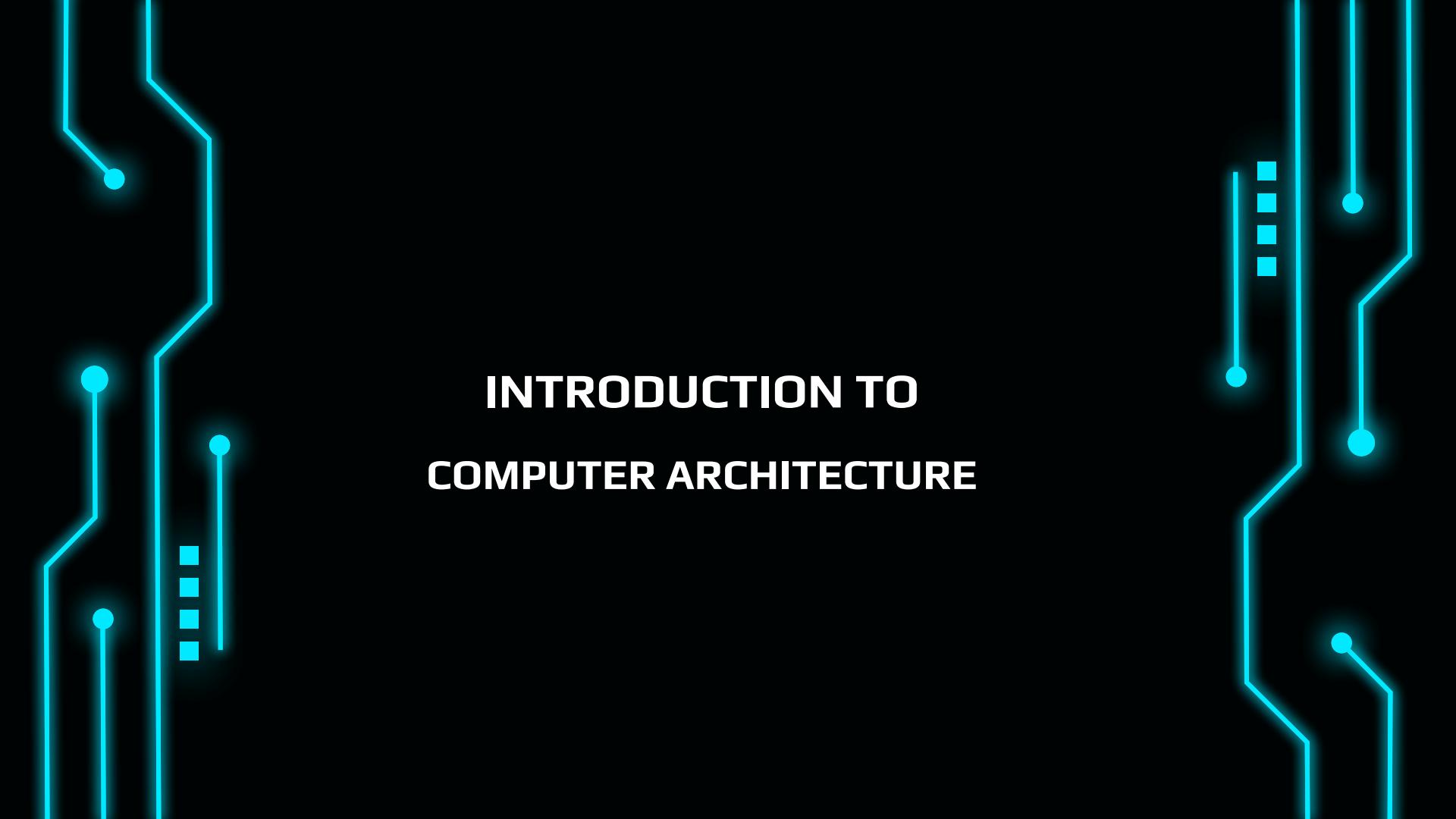
Setting up the
Development
Environment

03

Basic Python
Syntax

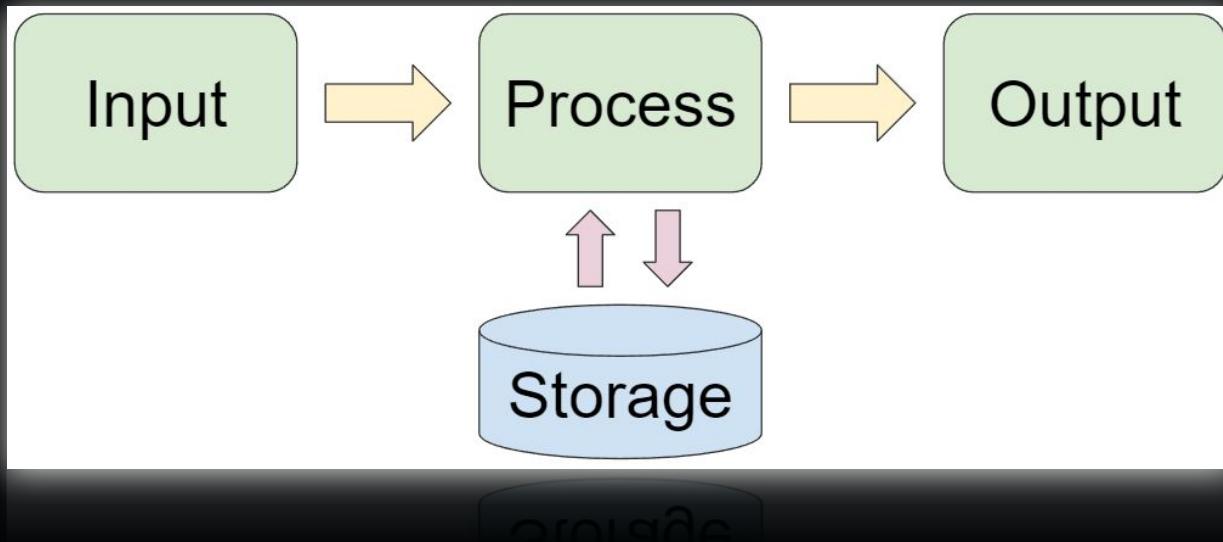
04

Example
Problems

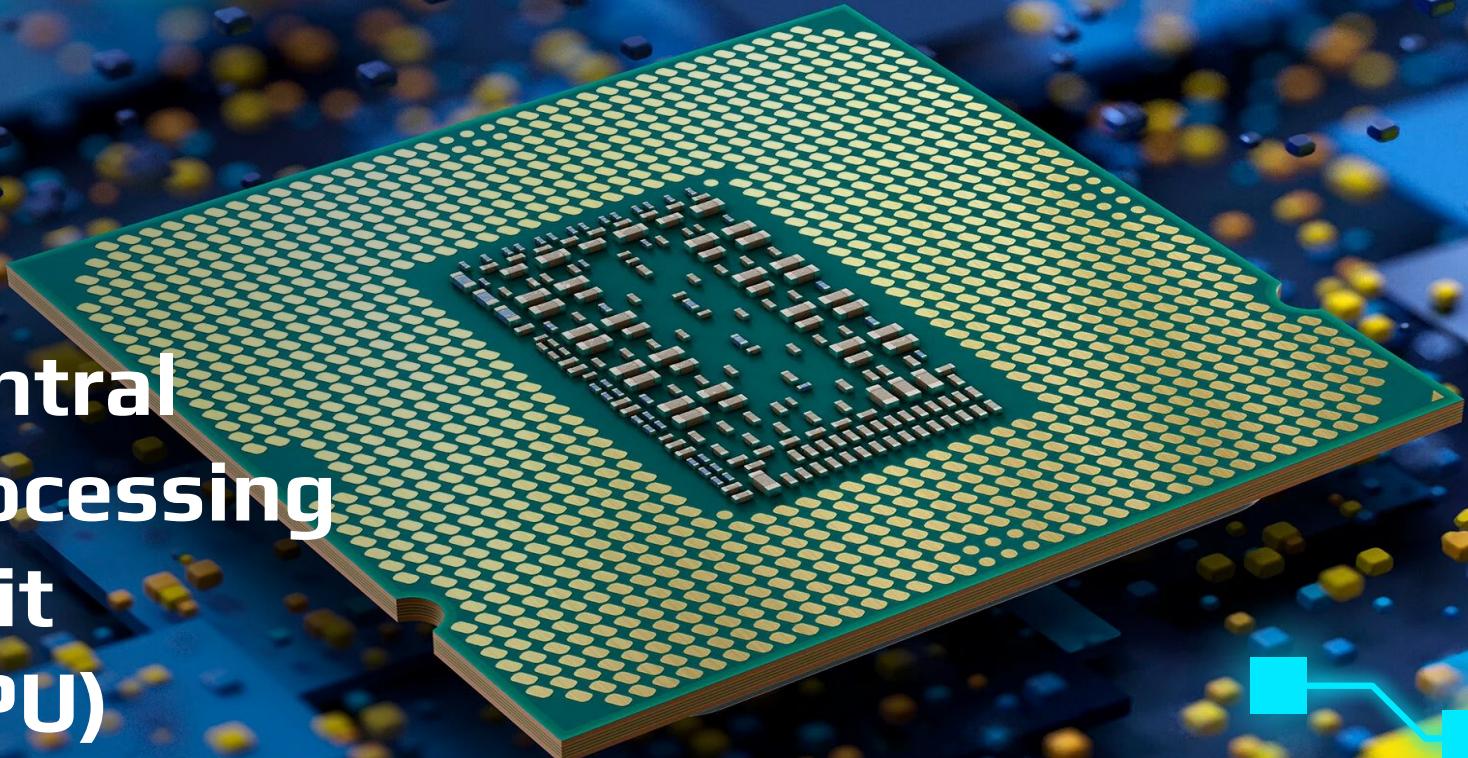


INTRODUCTION TO COMPUTER ARCHITECTURE

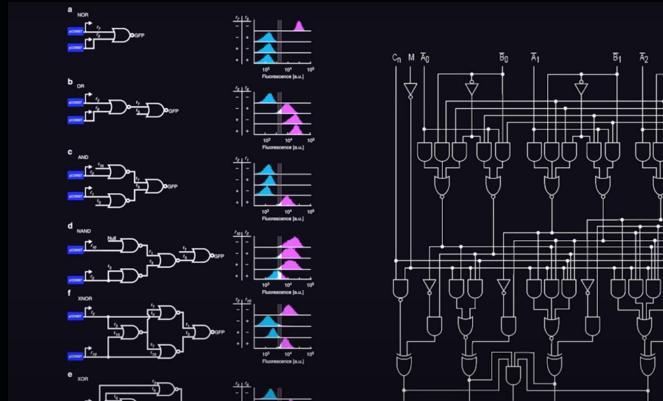
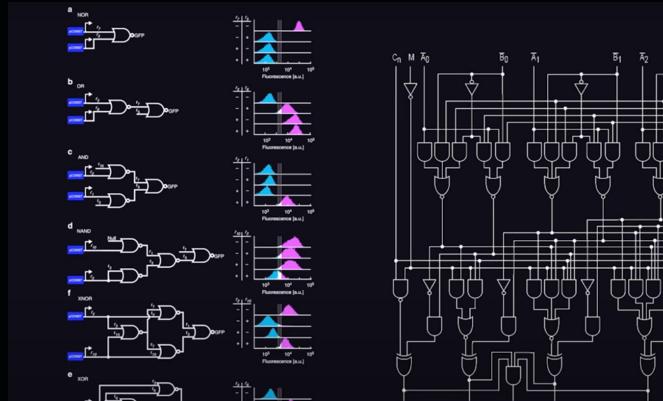
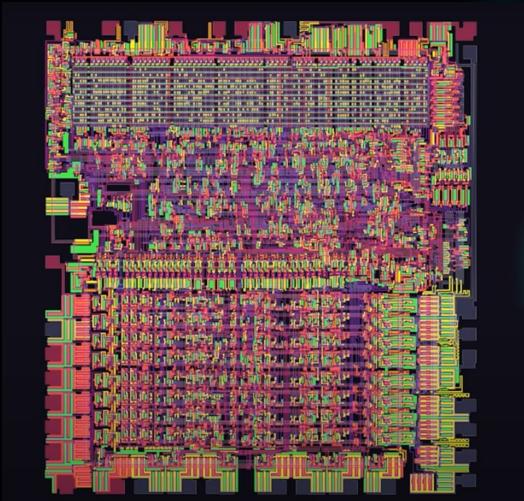
Working Process of Computers



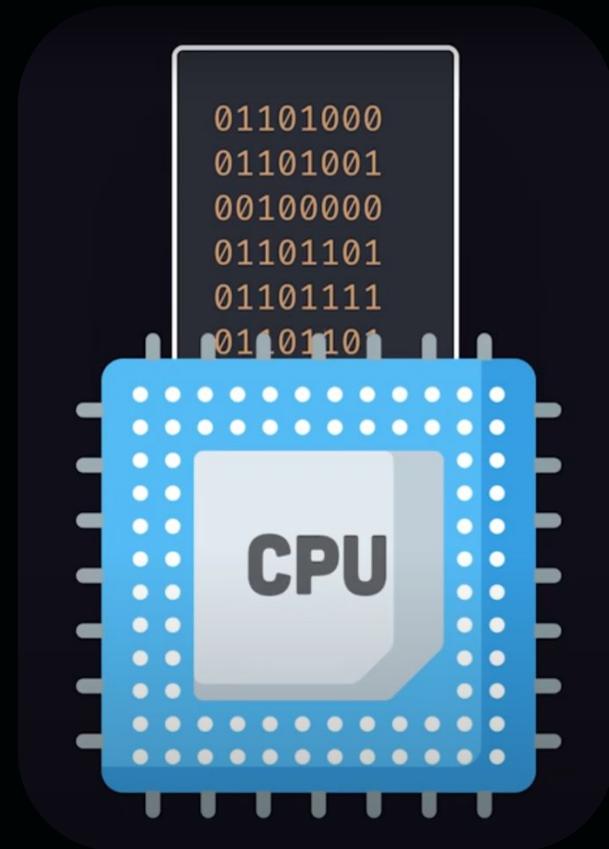
Central Processing Unit (CPU)

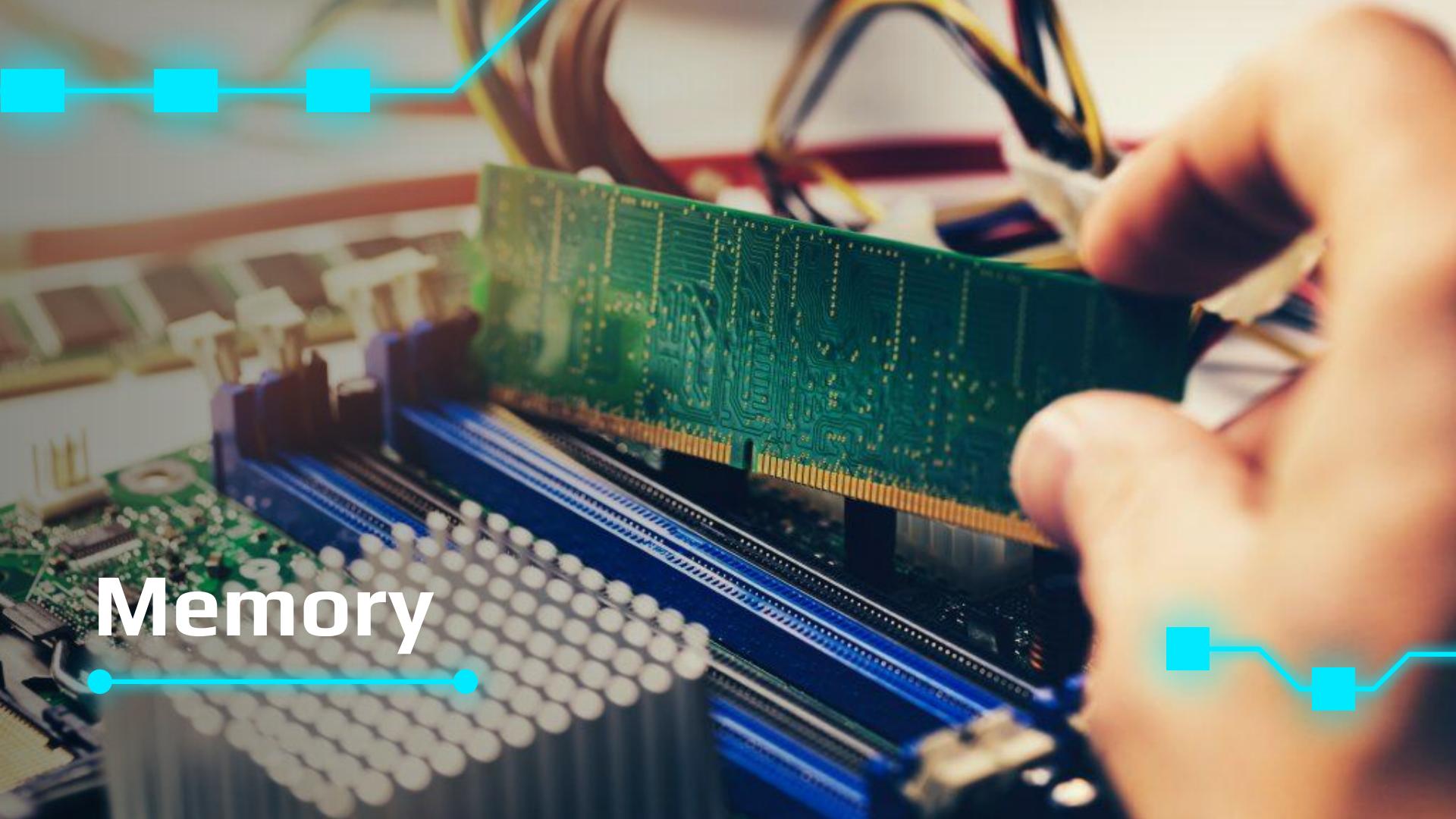


CPU Is Basically a Powerful Calculator



**Writing Code is
Writing Instructions
for CPU**



A close-up photograph of a person's hands holding a green RAM (Random Access Memory) module. The RAM has gold-colored metal contacts on one edge and a printed circuit board with various electronic components on the other. It is being held above a blue and white computer motherboard. In the background, there are blurred components like a CPU cooler and other RAM modules. A bright cyan line with square markers traces from the top left towards the RAM, and another line with a circular marker traces from the bottom left towards the RAM.

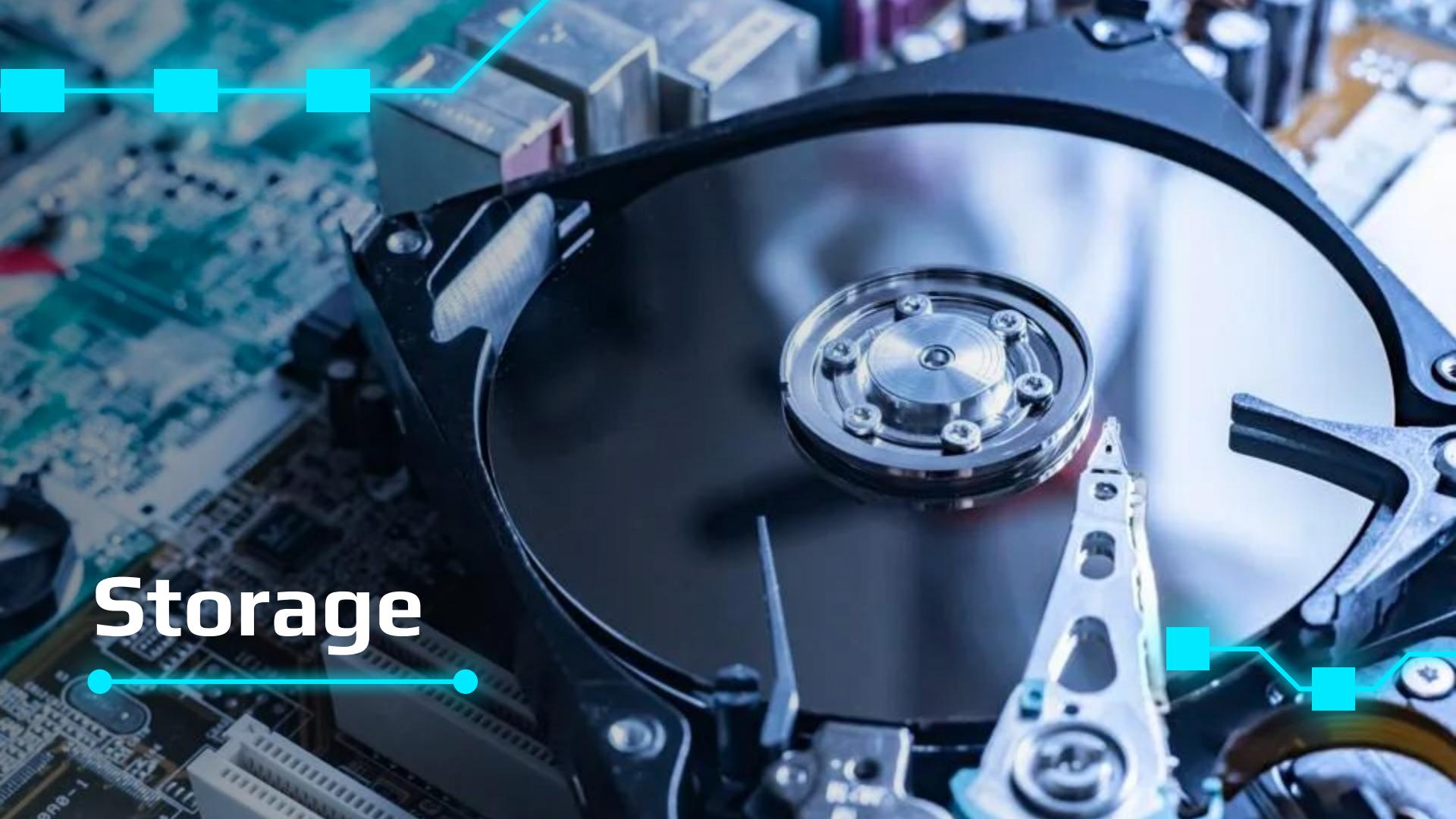
Memory

What Role Memory Plays?

Provides a temporary Storage

Resets and erases all data if power is lost.

Contains the information that CPU needs to process any data.



Storage

What Role Storage Plays?

Stores Data for long time

Does not lose data if power is lost.

Slower than memory

Compiler Vs Interpreter

Input	Takes an entire program at a time	Takes a single line of code or instruction at a time
Output	It generates intermediate object code	It does not produce any such
Speed	Faster	Slower
Memory	Requires more memory	Requires less memory
Errors	Display all errors after compilation	Displays error of the line being interpreted
Languages	C, C++, C#, Scala	PHP, Python, Ruby

Python Programming Language

- Created by Guido van Rossum in the late 1980s.
- Released in 1991 as Python 0.9.0.
- Named after the comedy group Monty Python.
- Guiding principle: Readability counts.



Why Python?

1. Readability and Simplicity:

- Clear Syntax: Easy to read and write.
- Clean Code: Emphasizes code readability.

2. Versatility:

- General-purpose: Web development, data science, machine learning, automation.
- Cross-Platform: Works on Windows, macOS, Linux

5. Compatibility and Integration:

- Interoperability: Easily integrates with other languages.
- Legacy Systems: Supports legacy code and systems.



Why we, Robotics Enthusiasts, Need Python?

1. Robotics Libraries and Frameworks:

- ROS (Robot Operating System): Python is widely used in ROS.
- Robotics Frameworks: PyRobot, Pypot, PyBullet and more.

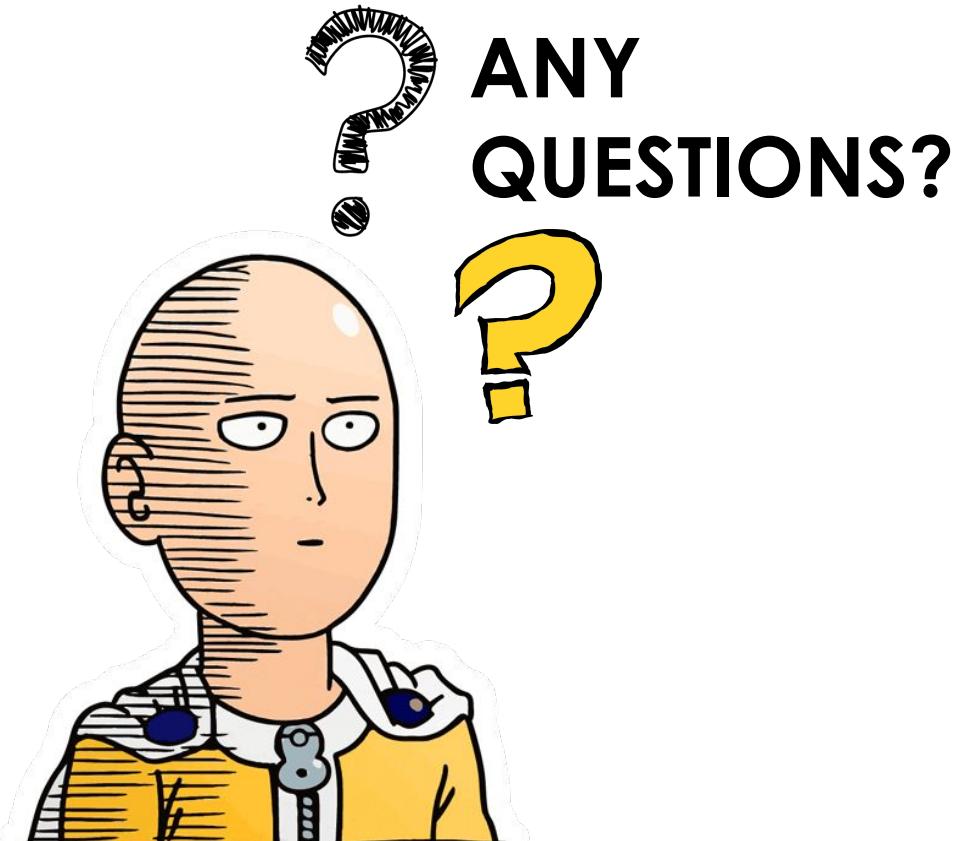
2. Machine Learning and AI:

- Integration: Python seamlessly integrates with ML and AI libraries.
- TensorFlow, PyTorch: Ideal for building intelligent robots.

3. Rapid Prototyping:

- Quick Development: Python accelerates the development cycle.
- Experimentation: Ideal for testing algorithms and ideas.





**ANY
QUESTIONS?**