

Computer Science Principles & Python Kickoff

Session 01 | 14-01-2026

Mentor: Monal

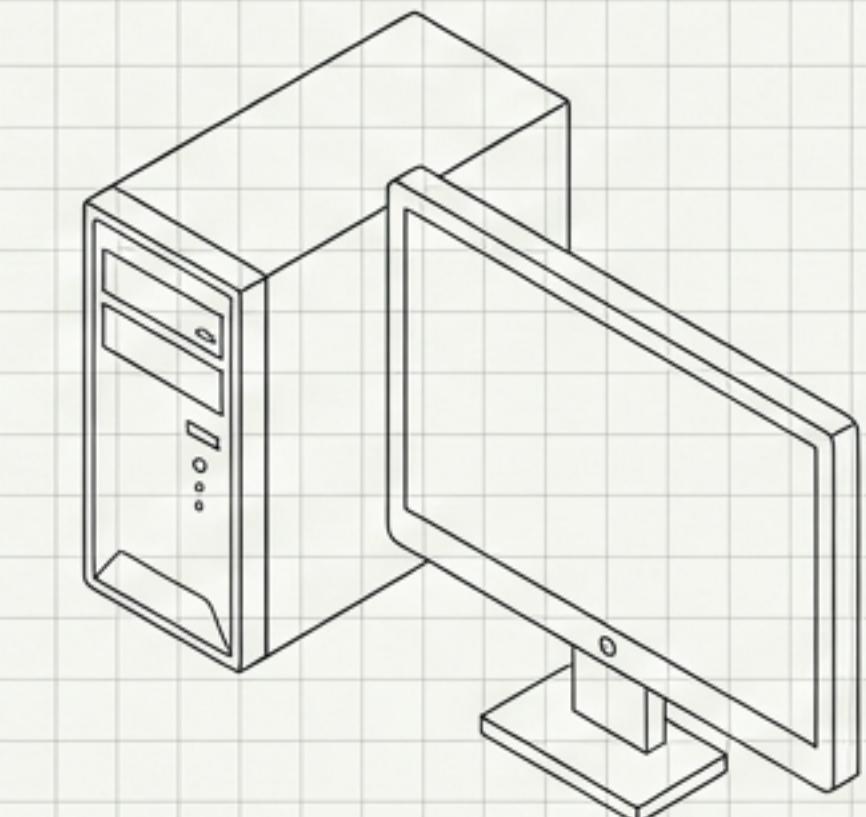
Weekly Schedule

Wednesday: 9:30 PM - 10:30 PM IST (**Boost Session**)

Sat/Sun: 9:00 AM - 1:00 PM IST (**Max Session**)

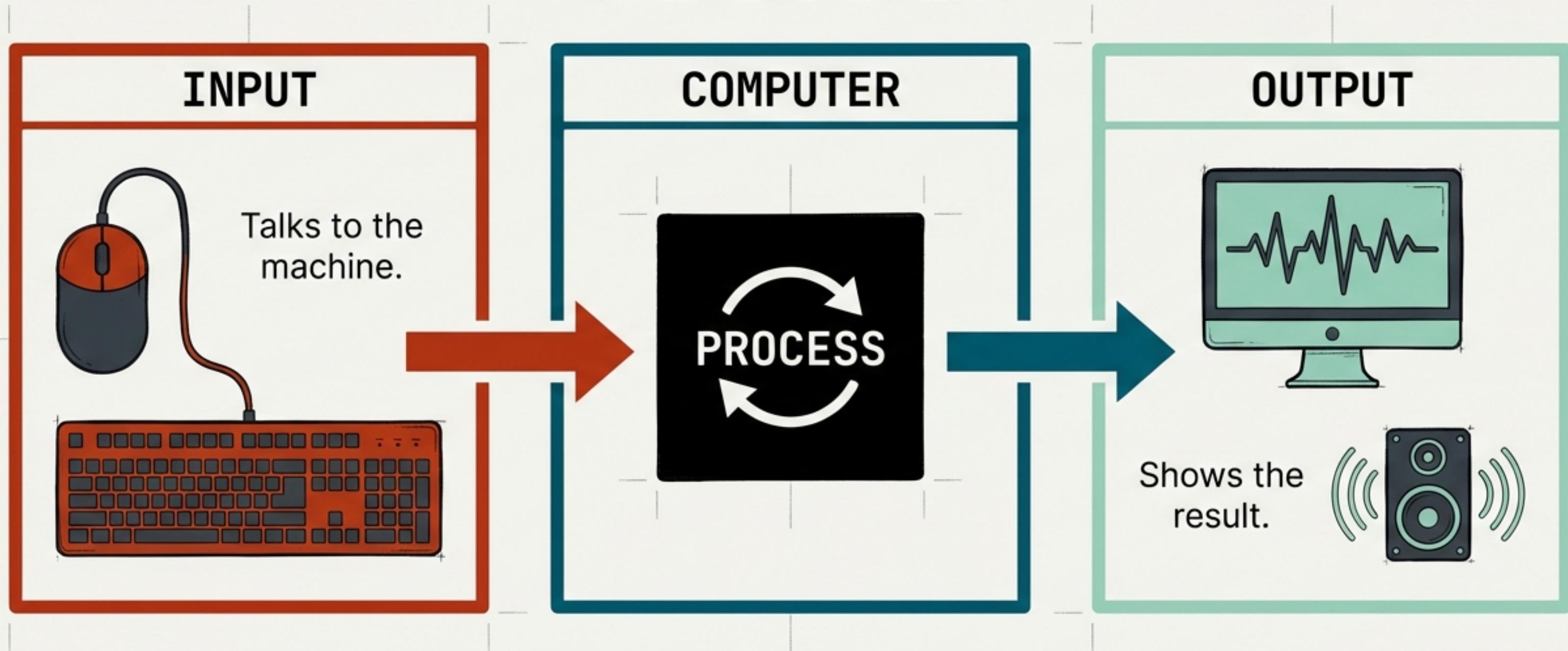
Agenda

1. The Computer (Demystification)
2. The Tools (Installation & Logic) → **why**
3. Special Announcement



THE CORE DEFINITION

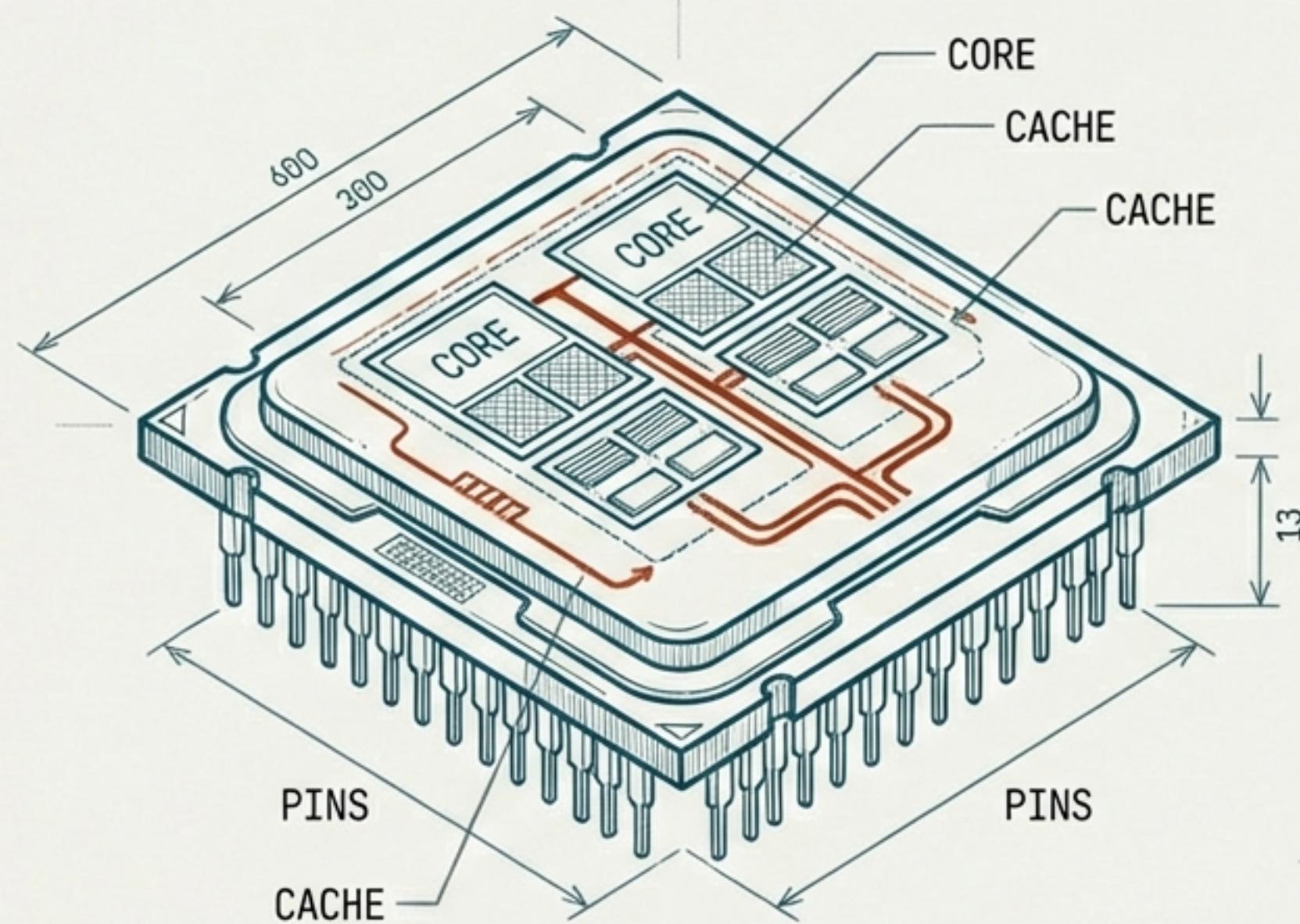
The Universal Workflow



A computer is simply a machine that follows a specific flow:
Input leads to Processing, which creates Output.

INSIDE THE BOX: THE BRAIN

CENTRAL PROCESSING UNIT (CPU)



ROLE:	Calculation & Processing
CHARACTERISTIC:	Low Memory / High Speed
CLOCK SPEED:	$\sim 3.0 \text{ GHz}$
$3.0 \text{ GHz} = \sim 3 \text{ Billion Calculations Per Second}$	

• Think of it as a genius with no long-term memory.

MEMORY HIERARCHY

VOLATILE VS. PERMANENT

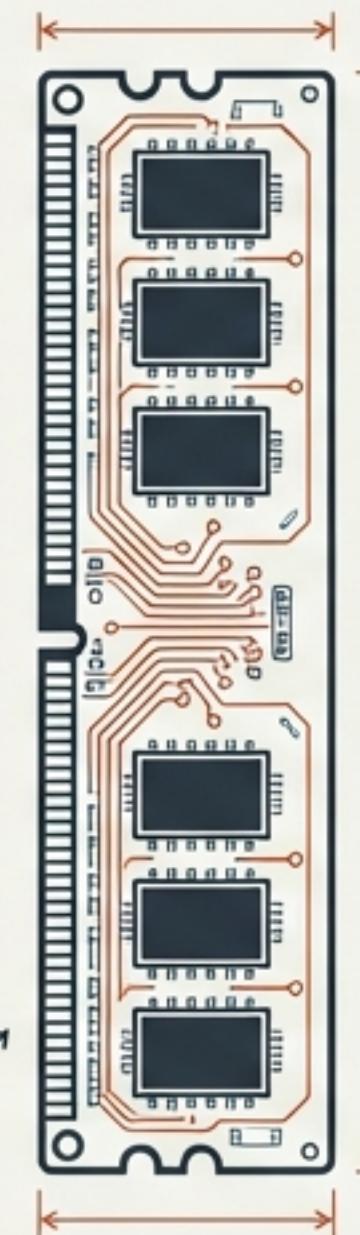
RAM (Random Access Memory)

VOLATILE

The active workspace.

When you open an app, it lives here. Cleared instantly when power is cut.

Think of it as a temporary desk.



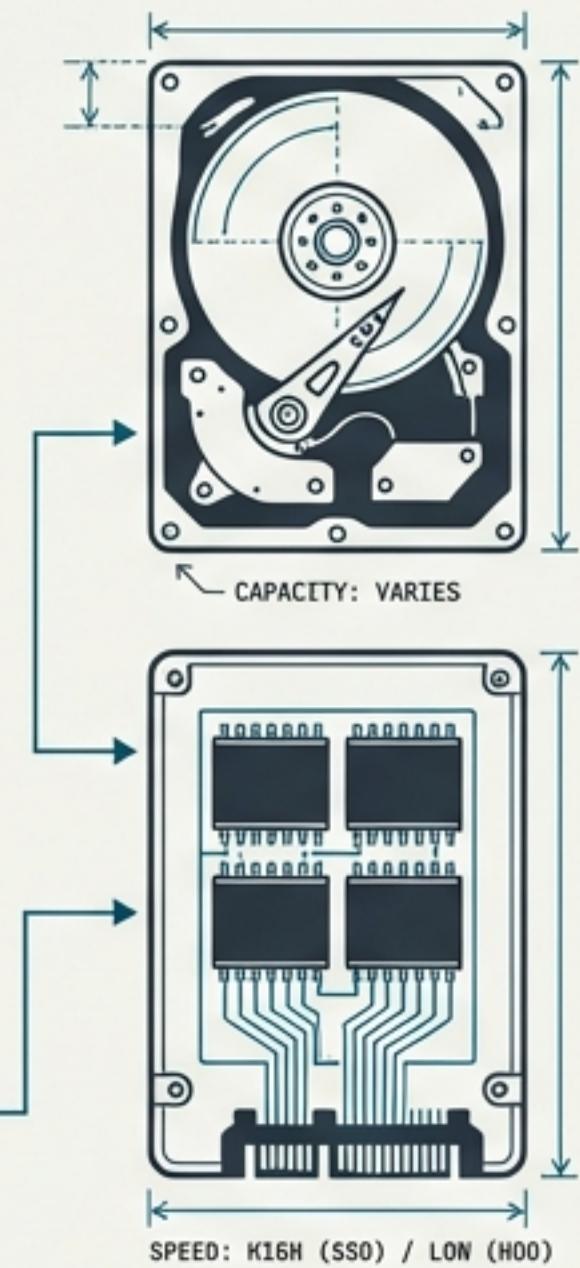
STORAGE (HDD & SSD)

PERMANENT

The filing cabinet.

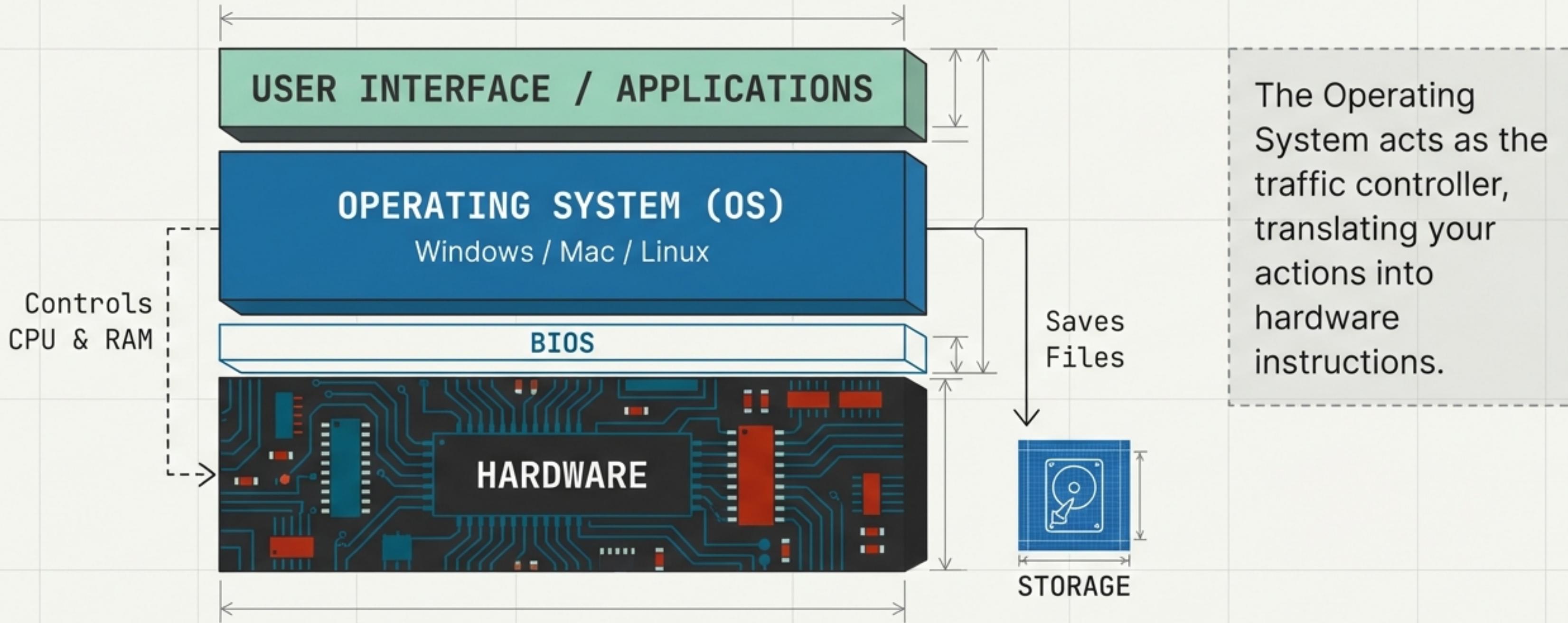
Where files, images, and videos exist forever.

SSD (Solid State) = Fast.
HDD (Spinning Disk) = Slow.



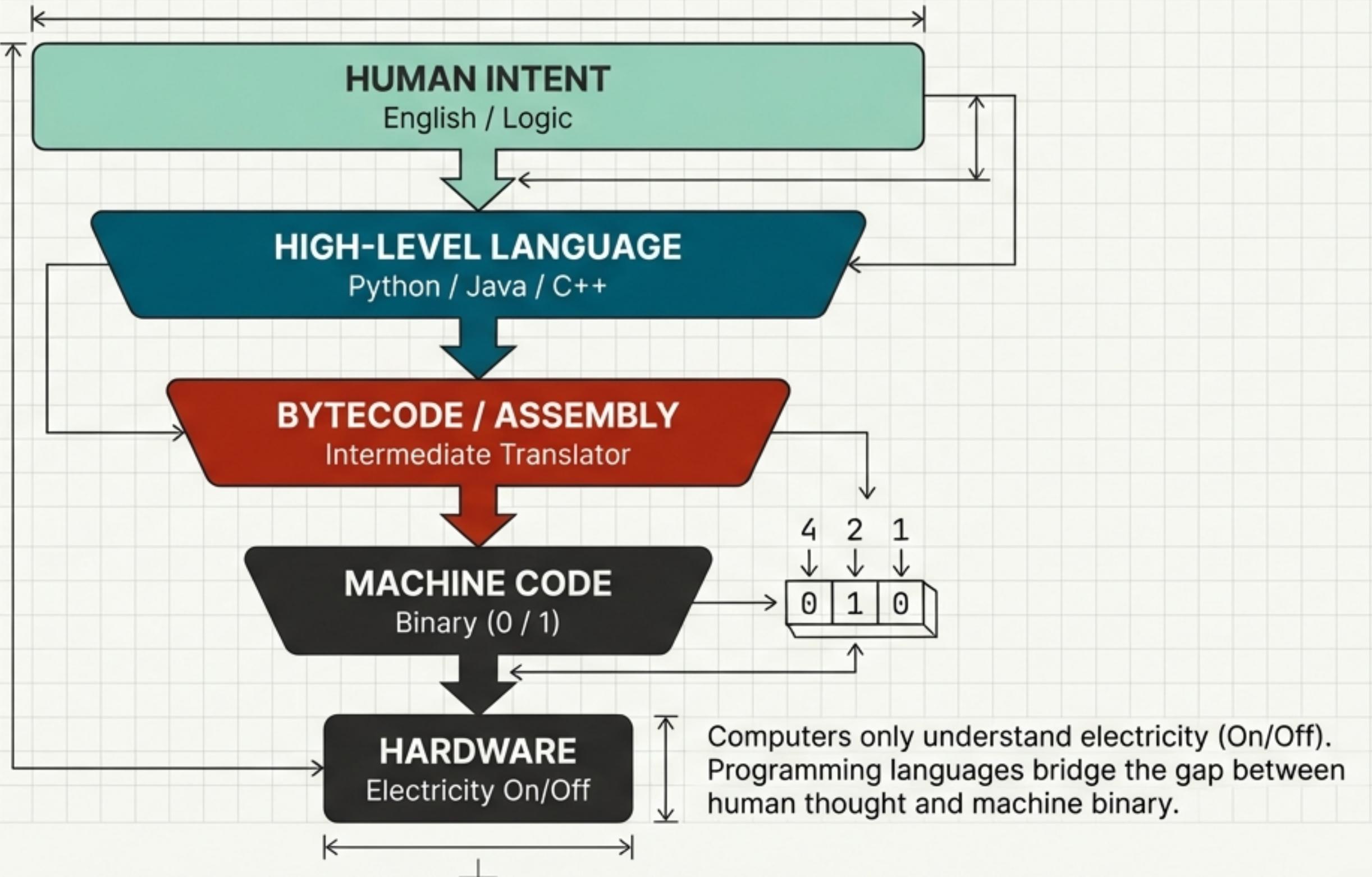
THE SOFTWARE STACK

MANAGING THE HARDWARE



TALKING TO THE MACHINE

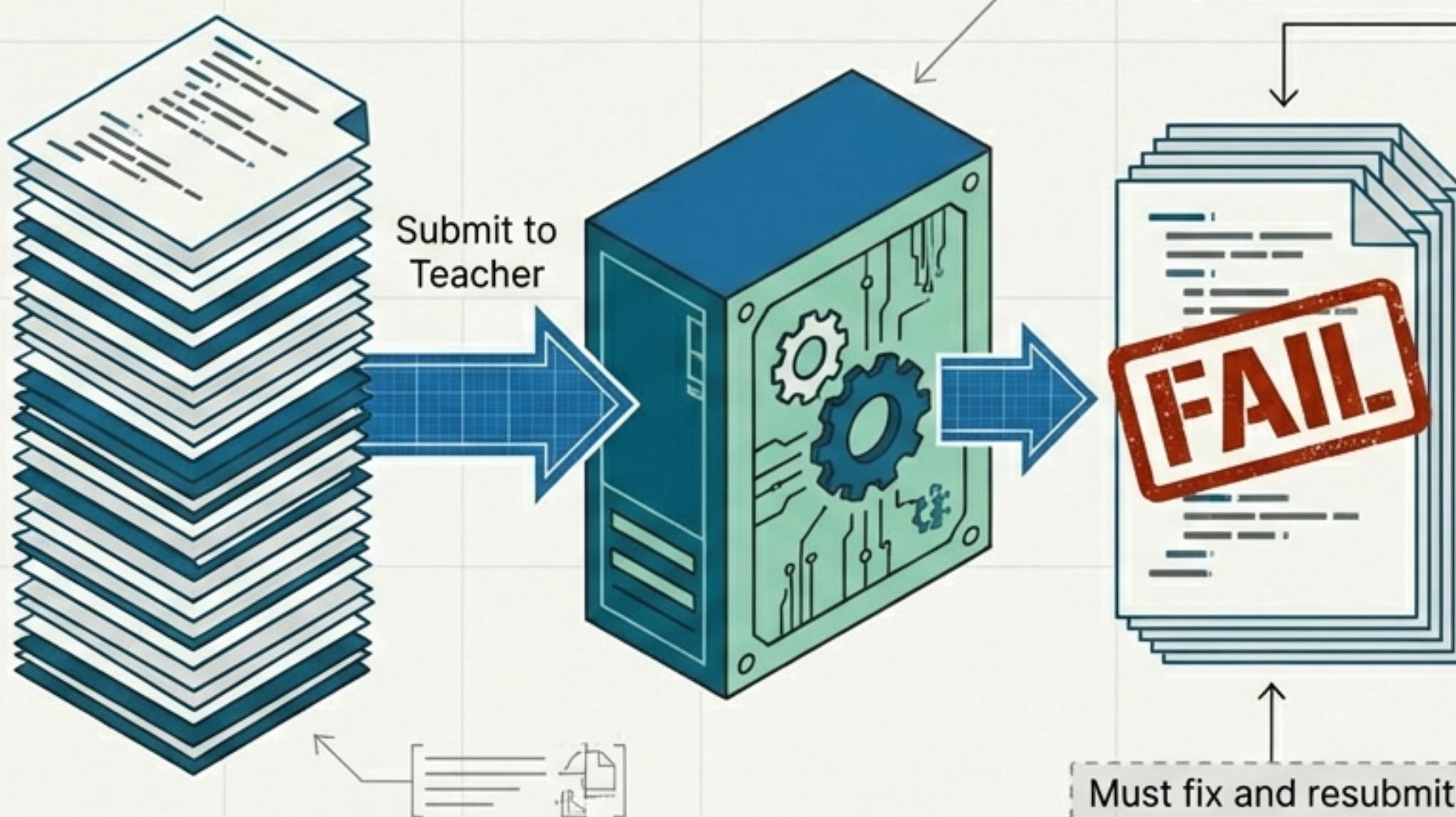
The Translation Gap



THE TRANSLATORS: COMPILER

The “Submit All” Approach

100 LINES OF CODE

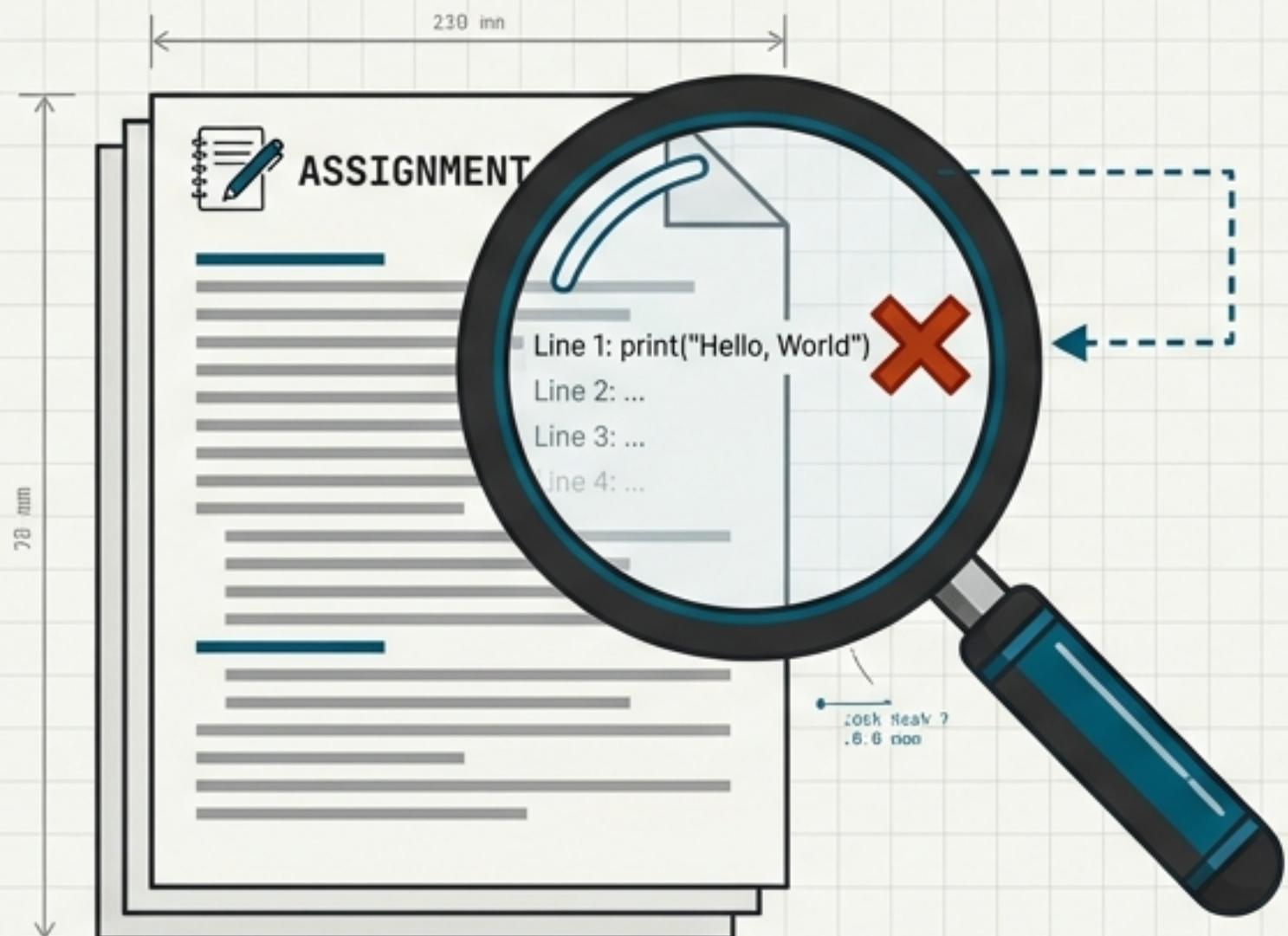


Key Characteristics

- Translates the ENTIRE program at once.
- Fast execution (once successful).
- Slow feedback loop (must fix and resubmit the whole stack).
- Examples: C, C++

THE TRANSLATORS: INTERPRETER

The "Line-by-Line" Approach



Key Characteristics

- Translates ONE line at a time.
- Stops immediately at the first error.
- Rapid feedback loop (perfect for learning).
- Slower execution speed.
- Examples: Python, JavaScript

Python is an Interpreted Language. It prioritizes developer speed over machine speed.

WHY PYTHON?

The Modern Standard

MATURE

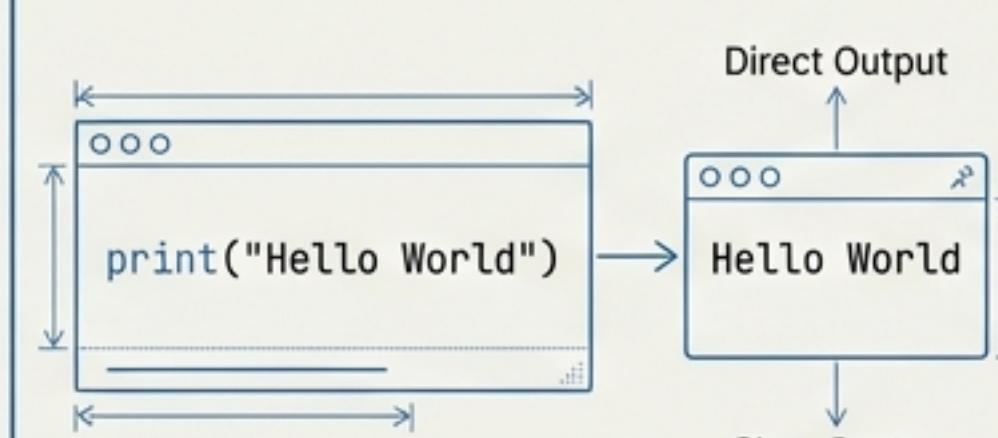
33 Years Old



1991: Python 1.0 2000: Python 2.0 2008: Python 3.0 Present: Continuous Growth

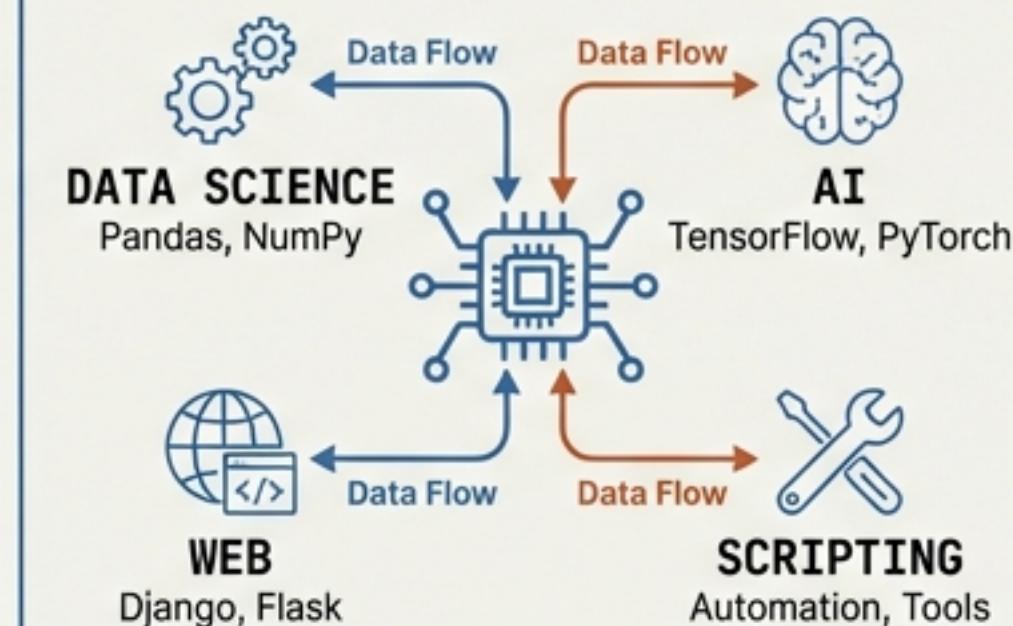
READABLE

High-Level English Syntax



VERSATILE

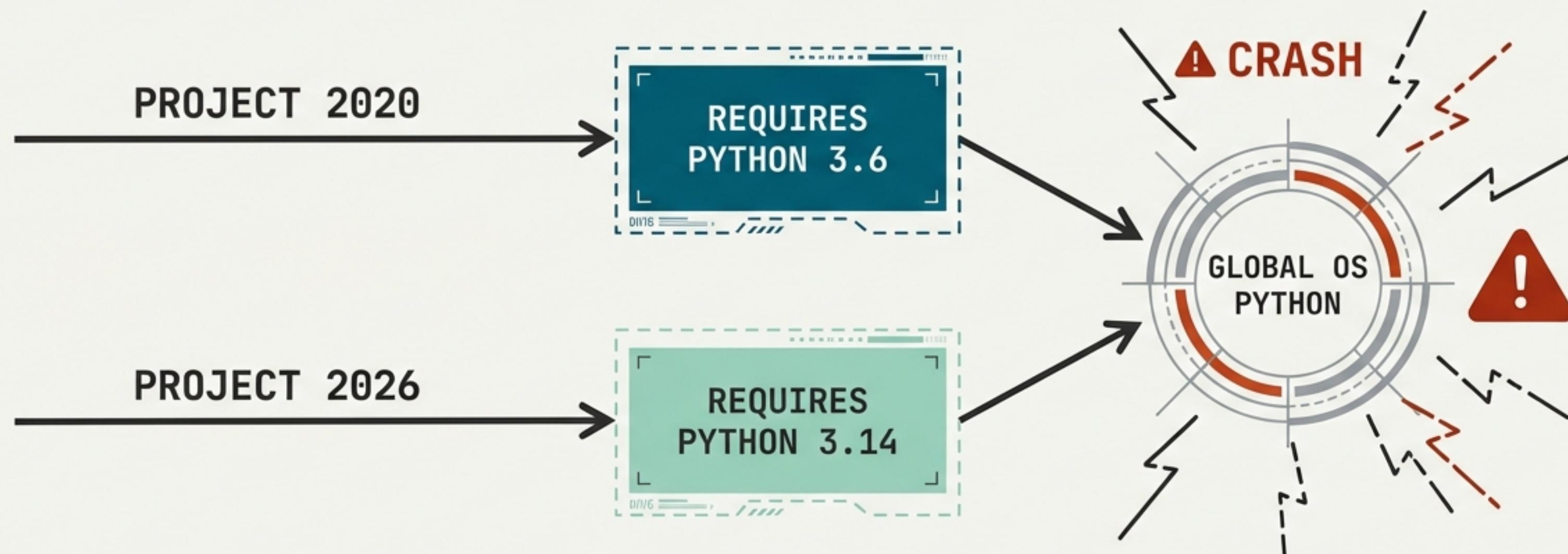
Data Science, AI, Web, Scripting



"From simple scripts to Instagram-scale applications."

THE PROBLEM: DEPENDENCY CONFLICT

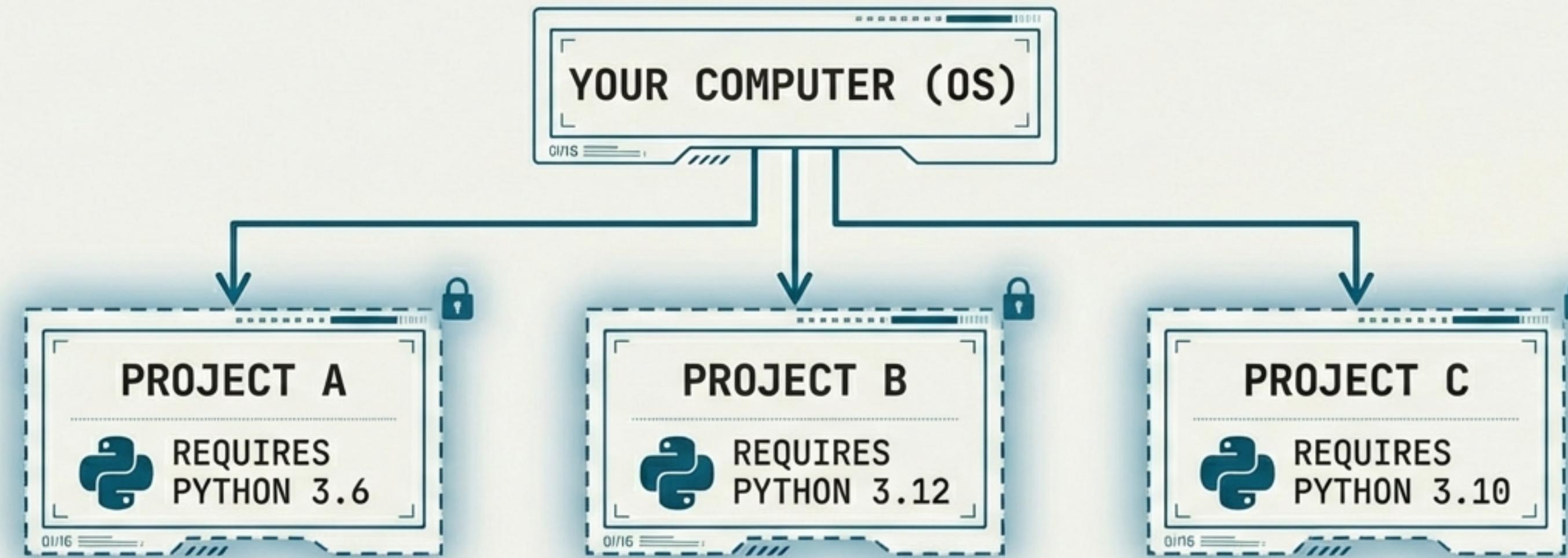
When Projects Collide



You cannot install two different versions of Python globally without breaking one of the projects.

THE SOLUTION: VIRTUAL ENVIRONMENTS

Project Isolation



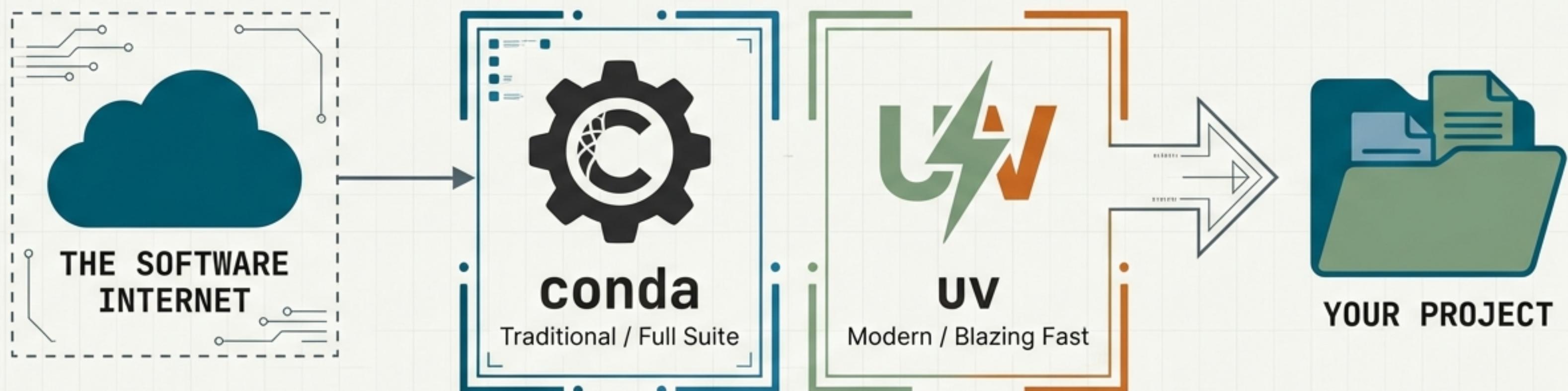
Each project gets its own private box.
What happens inside, stays inside.

TOOLS: `venv`, `conda`

THE LOGISTICS: PACKAGE MANAGERS

Importing Capability

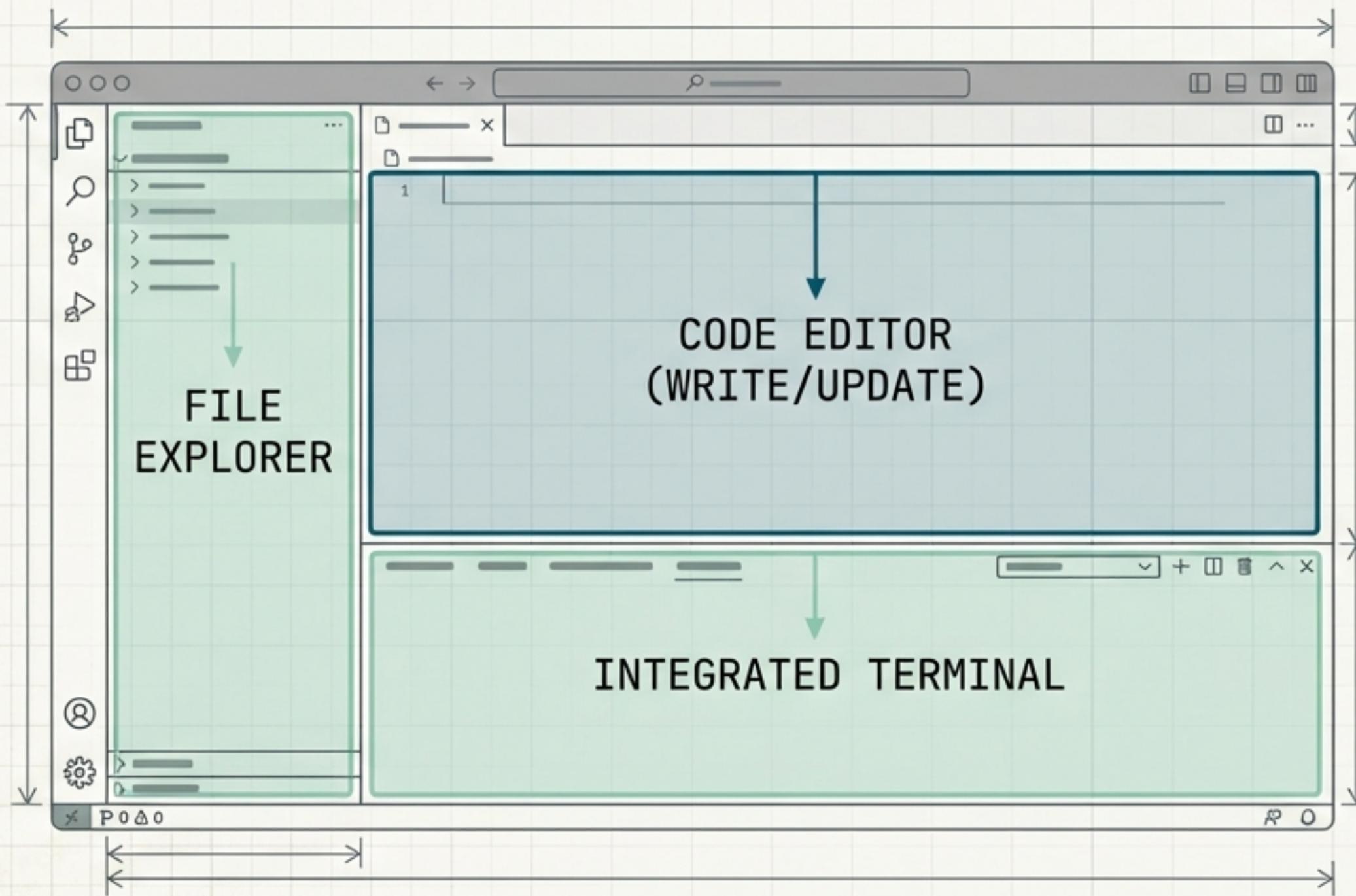
Just as you need a manager to handle logistics, Python needs tools to install and manage external software libraries.



Think of them as the delivery service that brings the parts you need to build your car.

THE WORKBENCH: THE IDE

Integrated Development Environment



✓ PRIMARY: VS CODE

ALTERNATIVES:
PYCHARM, CURSOR

✗ NOT RECOMMENDED:
~~NOTEPAD, RICH TEXT
EDITORS~~

The place where we Write,
Update, Create, and Delete
code.

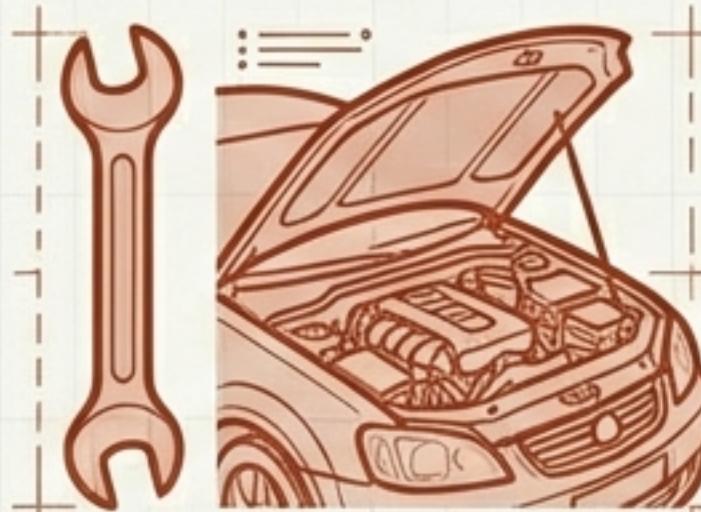
THE PHILE PHILOSOPHY OF DEBUGGING

Playing vs. Fixing



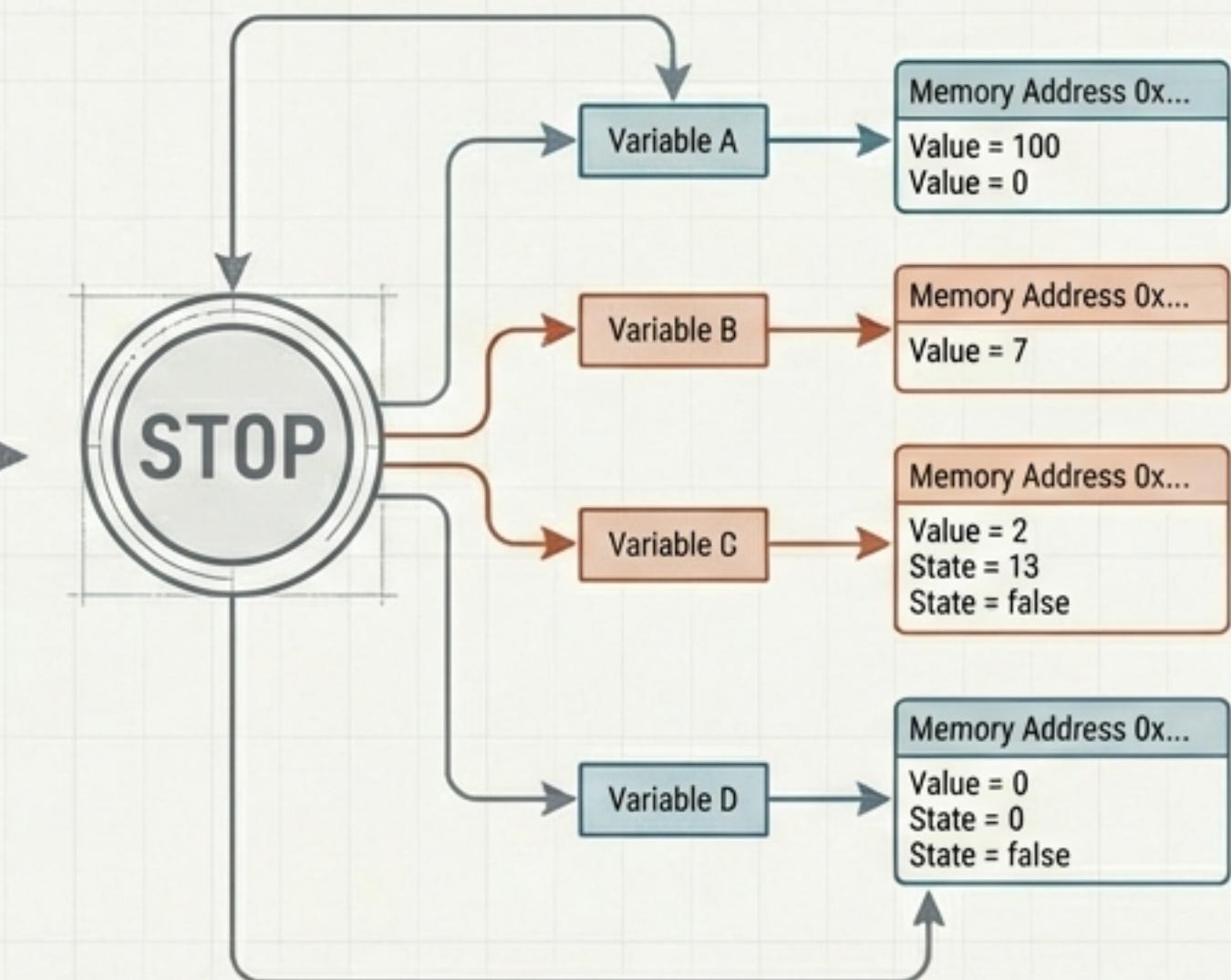
RUNNING CODE

Experiencing the flow.
Like playing a game.



DEBUGGING CODE

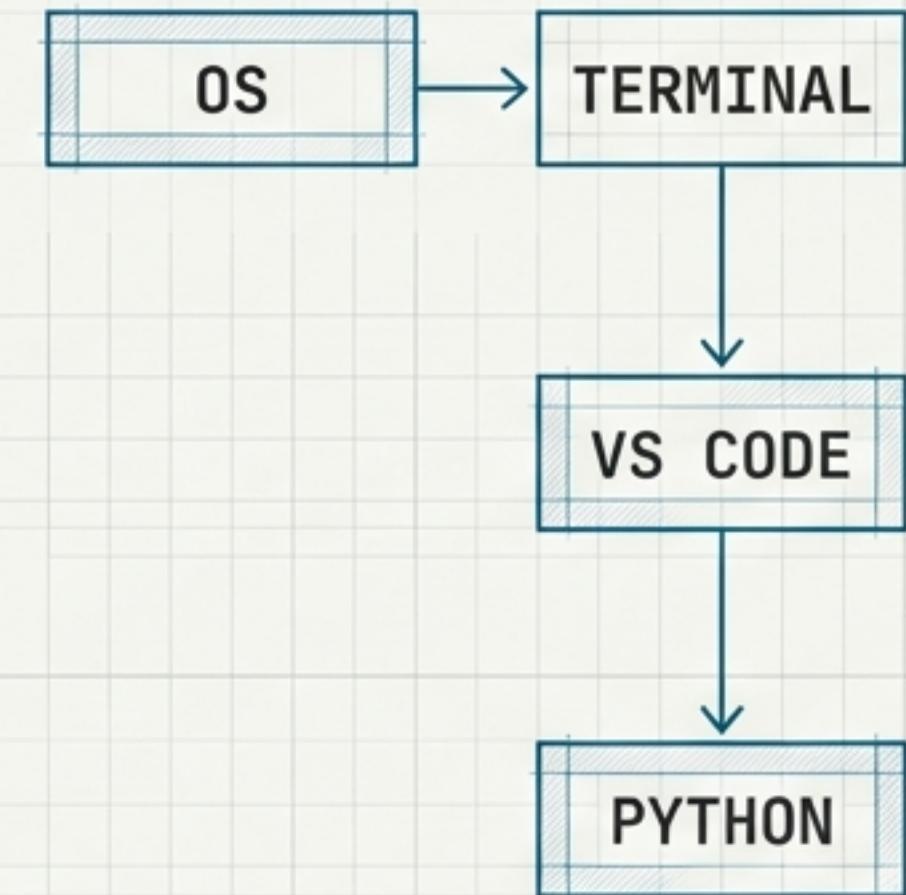
Stopping time.
Inspecting the machinery.
Fixing the game.



YOUR MISSION

Setup Checklist

- Install Python (Latest Version)
- Install VS Code (The Workbench)
- Install 'uv' (The Package Manager)
- Create your first Virtual Environment (venv)



The machine is waiting for your input. Let's build.