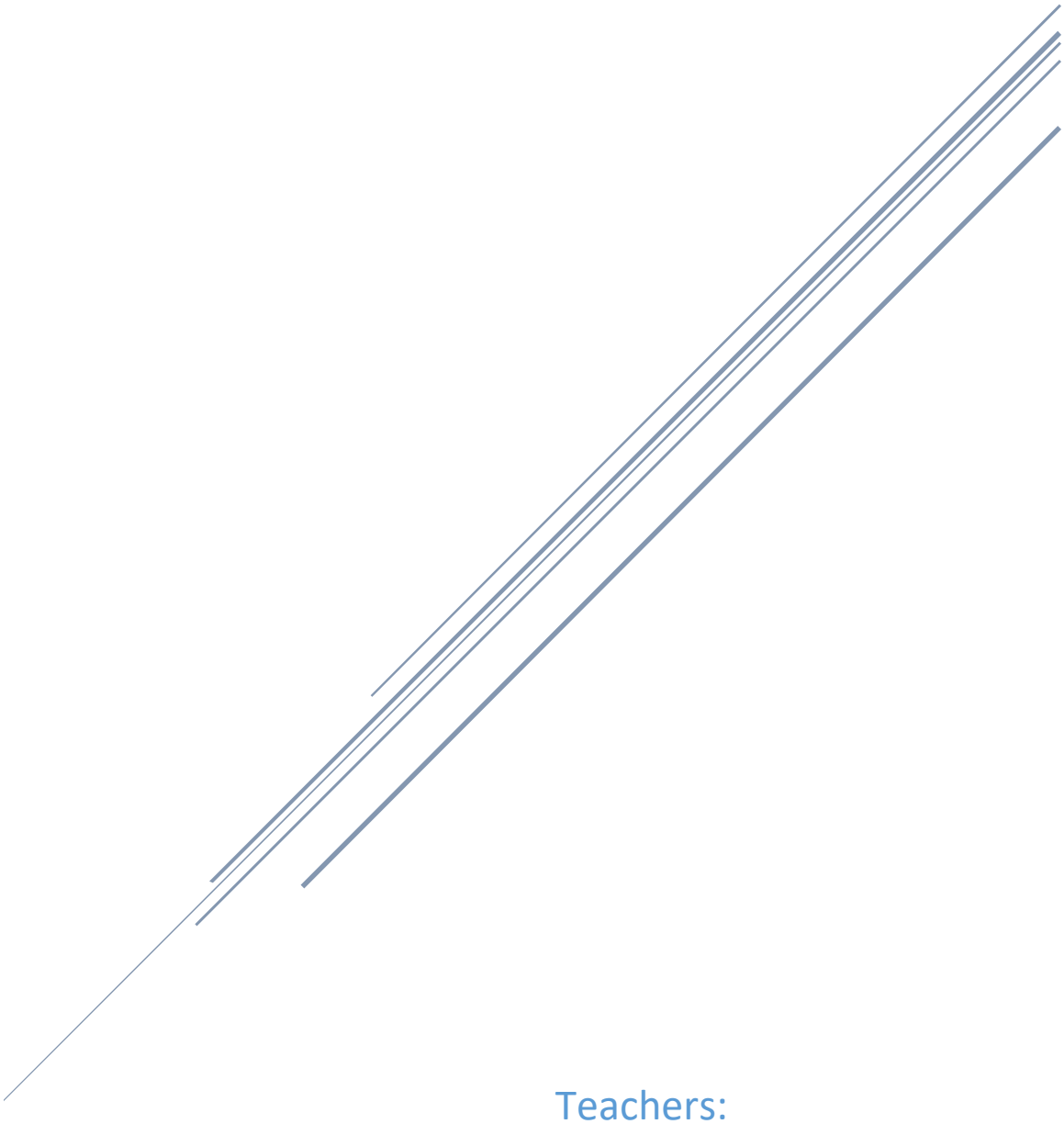
CODE AND SUMMARY OF CLASS 1

Sahir Ahmed Sheikh
Saturday (2 – 5)



Teachers:
Muhammad Bilal And Ali Aftab Sheikh

Code And Summary Of Class 1 – Saturday (2 – 5) | Quarter 3

Today was our first Quarter 3 class on Python.

First, our instructor introduced himself to the new students and explained that we would be covering Python using the **Panaverse** repository. The repo link is: https://github.com/panaversity/learn-modern-ai-python.git

Key Points from the Class:

- Python needs to be completed within 4 weeks, after which we will move on to AI.
- We started with **Step 00**, which covers **Python Colab**.
- The repository learn-modern-ai-python contains structured steps for learning Python.
    - Inside **Step 00**, there is a subdirectory for **Google Colab**.
    - Within this, there are multiple steps from **Step 01 to Step 15**, covering various topics:
        - **Step 01:** Introduction to Python
        - **Step 02:** Data Types
        - And so on...

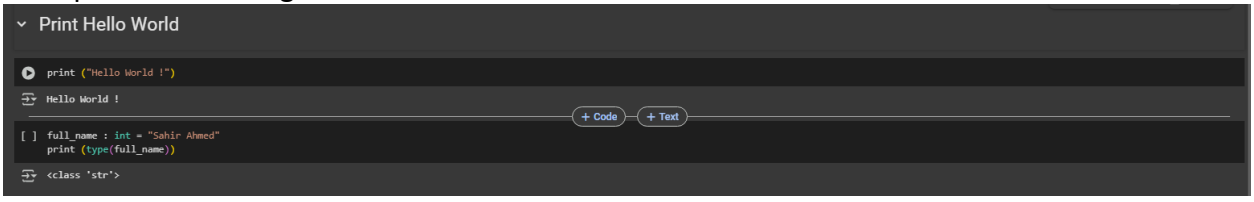
Google Colab: Introduction & Usage

Google Colab is a free cloud-based environment **that allows us to write and execute Python code in our browser. It is mainly used for** machine learning, data science, and AI projects **since it provides** free access to GPUs and TPUs**.**

How to Use Google Colab

1. Search **Google Colab** in your browser and open the second link (🔗 colab.google).
2. Click on **New Notebook** to create a Python notebook.
3. In the top-right corner, click **Connect** to assign RAM.
4. You can create **multiple code cells** to write and execute Python code.
5. You can also add **text cells** (Markdown) to write explanations separately.
    - In **VS Code**, we use # for comments, but in Colab, we can directly use text cells.

Example Code in Google Colab:



Markdown Basics (for text formatting in Colab)

- # for **Headings**
- **Bold** for **Bold Text**
- *Italic* for *Italicized Text*
- - for Bullet Points
- 1. for Numbered Lists

CPU vs GPU vs TPU

We learned about **CPU, GPU, and TPU** with diagrams.

CPU (Central Processing Unit)

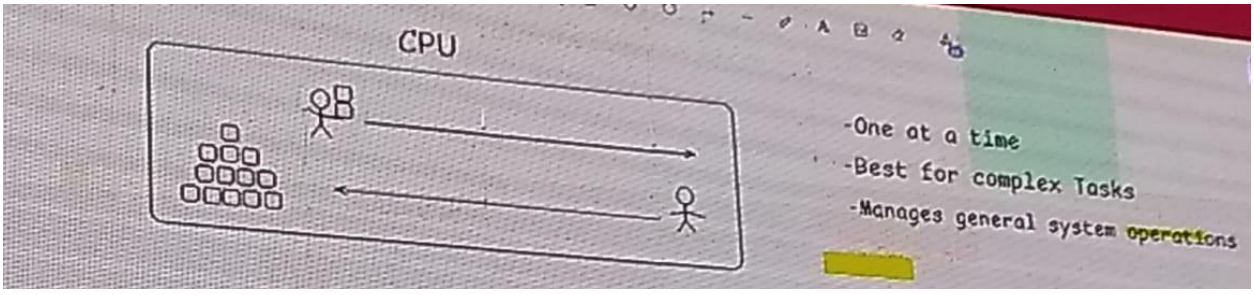
The **CPU** is the brain of a computer that handles general-purpose processing. It is designed for sequential tasks and executes a few complex instructions at a time.

Key Features:

- **One Task at a Time:** CPUs process tasks **sequentially**, meaning they complete one instruction before moving to the next.
- **Best for Complex Tasks:** Since CPUs have powerful cores, they are optimized for **single-threaded** and complex computations.
- **Manages General System Operations:** The CPU is responsible for handling OS operations, running applications, and performing logic-based tasks.

Common Uses of a CPU:

- Operating system management
- Running software applications
- Handling logic-based calculations
- Managing memory and input/output operations



GPU (Graphics Processing Unit)

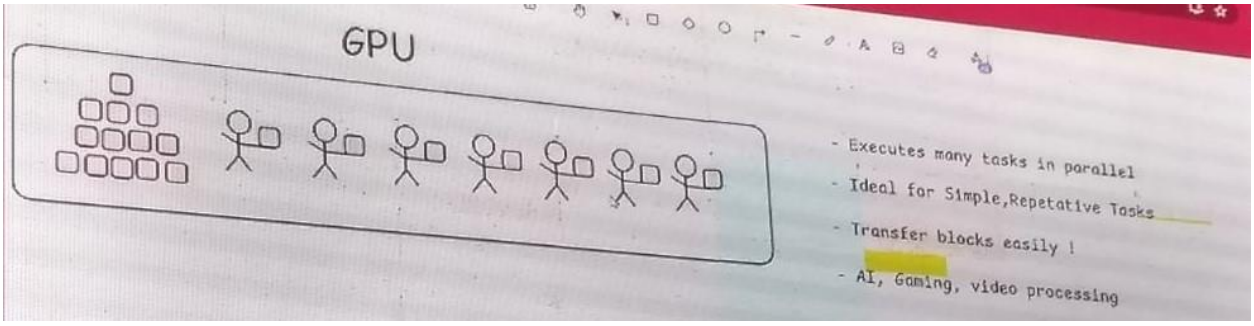
The **GPU** is a specialized processor designed for handling **parallel processing**. Unlike CPUs, GPUs consist of thousands of smaller cores that work together to handle multiple tasks simultaneously.

Key Features:

- **Parallel Execution:** A GPU can execute many tasks **simultaneously**, making it ideal for workloads that require massive data processing.
- **Ideal for Simple, Repetitive Tasks:** It excels in processing large amounts of data where the same operation is performed multiple times.
- **Transfers Blocks of Data Easily:** Because GPUs are optimized for handling large datasets, they can move and process blocks of data efficiently.
- **Used in AI, Gaming, and Video Processing:** GPUs power deep learning, 3D rendering, cryptocurrency mining, and real-time graphics processing.

Common Uses of a GPU:

- Machine learning and deep learning computations
- Gaming and rendering high-quality graphics
- Video editing and processing
- Simulations and big data analysis



TPU (Tensor Processing Unit)

A **TPU** is a specialized AI accelerator chip developed by **Google** for handling machine learning workloads efficiently. It is optimized specifically for **tensor operations**, which are heavily used in deep learning models.

Key Features:

- **Designed for Machine Learning:** TPUs are built specifically for AI workloads and outperform GPUs in training deep learning models.

- **Optimized for TensorFlow:** Since TPUs are developed by Google, they are deeply integrated with TensorFlow, making AI training and inference much faster.
- **Higher Energy Efficiency:** TPUs consume less power compared to GPUs while delivering superior performance in AI tasks.

Common Uses of a TPU:

- Training deep learning models (e.g., neural networks)
- Accelerating AI applications like Google Search, Google Photos, and speech recognition
- Performing large-scale AI inference tasks efficiently

We didn't cover **TPU** in detail since we won't be using it much for now.

Differences Between CPU, GPU, and TPU

Feature	CPU	GPU	TPU
Processing Type	Sequential	Parallel	Optimized for AI
Best For	General computing	Graphics & parallel computing	Machine learning
Speed	Slower for parallel tasks	Faster in parallel tasks	Fastest in AI processing
Power Consumption	Moderate	High	Lower than GPU
Examples of Use	Operating systems, software, logic operations	Gaming, video processing, AI	Deep learning, AI inference

Conclusion

- **CPUs** are designed for **general-purpose** tasks and handle complex logic.
- **GPUs** excel in **parallel processing**, making them ideal for **graphics rendering and AI workloads**.
- **TPUs** are **AI-specific processors** optimized for **machine learning and deep learning**.

Step 01: Introduction to Python

We started learning **Python fundamentals**, beginning with:

Why Learn Python?

Python is a **high-level, interpreted, and versatile programming language** known for its **simplicity and readability**.

- Supports **multiple paradigms** (procedural, OOP, functional programming).
- Used in **web development, AI, data science, automation, etc.**
- **Cross-platform** and beginner-friendly.

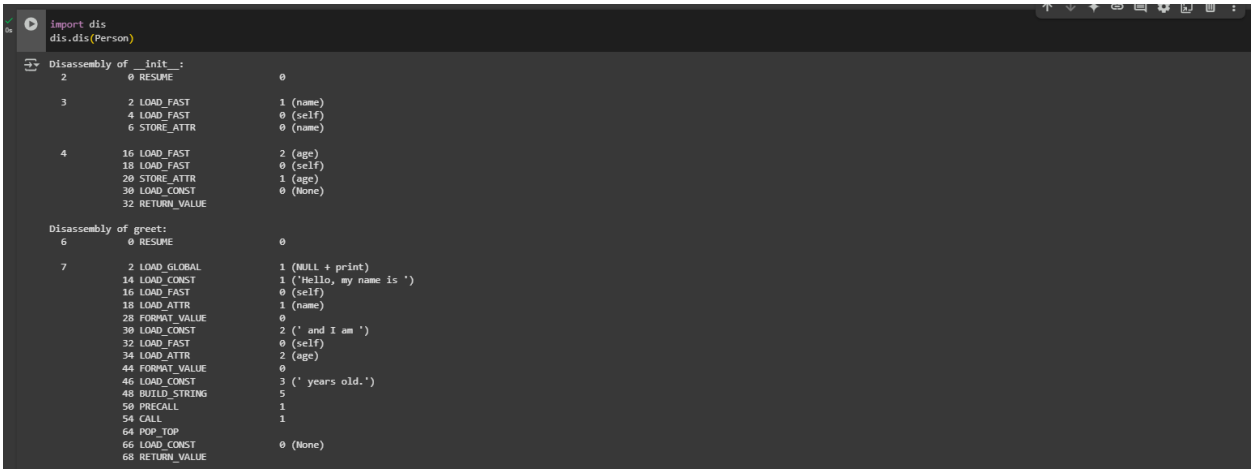
How Python Works (Compilation Process)

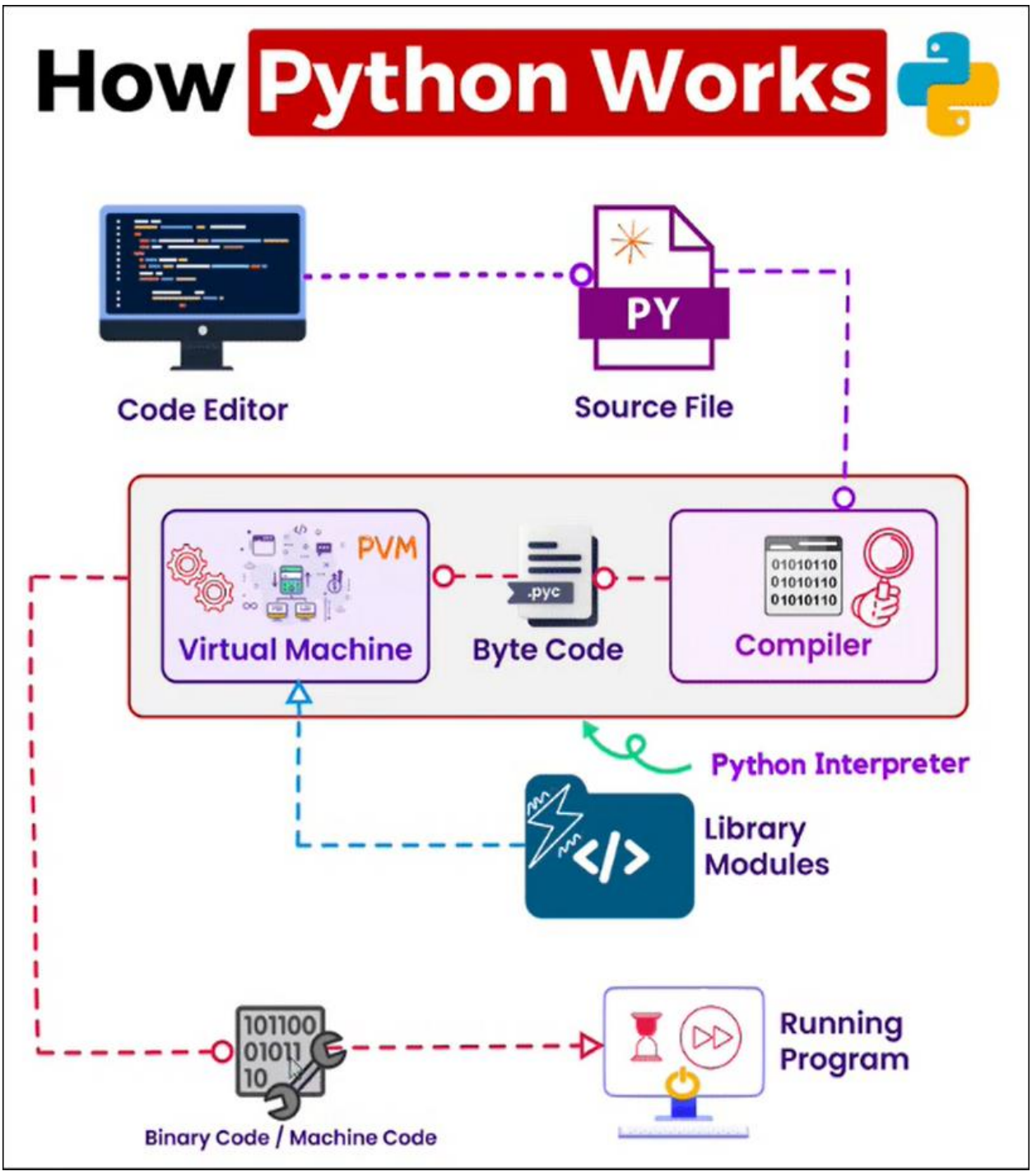
1. **Write Code** → in a Python file (.py extension).
2. **Compilation** → Python compiles the code into **bytecode**.
3. **Bytecode Execution** → The Python Virtual Machine (PVM) executes it.
4. **Machine Code** → The program runs on your computer.

What is Python Bytecode?

- Bytecode is an **intermediate representation** of Python code.
- It is **platform-independent** and stored in .pyc files.
- It helps Python **execute faster** without recompiling every time.

To see the bytecode of a function, use:





Why is Bytecode Important?

- **Cross-platform**: Runs on any system with Python installed.
- **Flexible**: Supports dynamic typing and modifications.
- **Caching**: Python stores .pyc files in __pycache__ for faster execution.

Indentation in Python

Python **strictly follows indentation** for structuring code.

```
# Correct indentation
if True:
    print("Hello, World!")
    print("This is a block of code")

# Incorrect indentation
if True:
print("Hello, World!")
  print("This is a block of code")

# Correct indentation
def greet(name: str):
    print("Hello, " + name + "!")

# Incorrect indentation
def greet(name: str):
print("Hello, " + name + "!")
```

Best Practices:

- Always use **consistent indentation** (preferably 4 spaces).
- Avoid using **tabs**, as they may cause errors.
- Use an **IDE** with auto-indentation support.

Then we understood a bit about Python syntax—what a float is, what an int is, how functions are written in Python, and that an array is called a list in Python. After that, we wrote its code. I have shared the implementation code for all of this in the link below. Feel free to check it out:

Link: https://colab.research.google.com/drive/12nR-_IV-0aXvDaOlQghVHSIFOa1wv73T?usp=sharing

Assignments Given:

✓ **Assignment 1:** Complete the first **7 steps** from the Panaverse Python repo.
Learn Modern AI with Python

✓ **Assignment 2:** Growth Mindset Challenge (Self-study)
Growth Mindset Challenge

✓ **Assignment 3:** Research and explain **Python Operators (is vs in)**

✓ **Assignment 4: Agentic AI Presentation**
Prepare a PowerPoint presentation on **Agentic AI** and upload it on **LinkedIn**.

📅 **Deadline:** Before the next class
📌 **Submission:** Upload all assignments on **LinkedIn**

This was our first class summary. We covered **Google Colab, CPU vs GPU vs TPU, Python compilation, Bytecode, Indentation rules, and python syntax.**

Next, we will continue learning more **Python fundamentals**!

Thank You for Reading!

Hope you understood Class 1 well.

"The beautiful thing about learning is that no one can take it away from you."
– B.B. King