Python Inner Working Explained

📝 1. Writing the Code

You write Python code in a `.py` file using a text editor or IDE:

```
print("Hello, World!")
```

⚙ 2. Compilation to Bytecode

- Python **first compiles** your code to an intermediate form called **bytecode**.
- Bytecode is **not machine code**, but a lower-level, platform-independent representation.
- It's usually saved in `.pyc` files inside a `__pycache__` folder.

□ Example:

```
print("Hello")   # → Compiled to Bytecode like LOAD_NAME, CALL_FUNCTION, etc.
```

🐍 3. CPython (The Default Interpreter)

- **CPython** is the **default Python implementation**, written in **C language**.
- CPython takes the bytecode and **interprets** it using the next component — the **PVM**.

💡 There are other implementations too:

- **Jython** (Python in Java)
- **IronPython** (Python in .NET)
- **PyPy** (Faster Python with JIT)

□ 4. PVM (Python Virtual Machine)

- The **Python Virtual Machine (PVM)** reads and **executes bytecode line by line**.
- It handles:
  - Memory management (via Garbage Collector)
  - Dynamic typing
  - Function calls, exceptions, etc.

So in simple words:

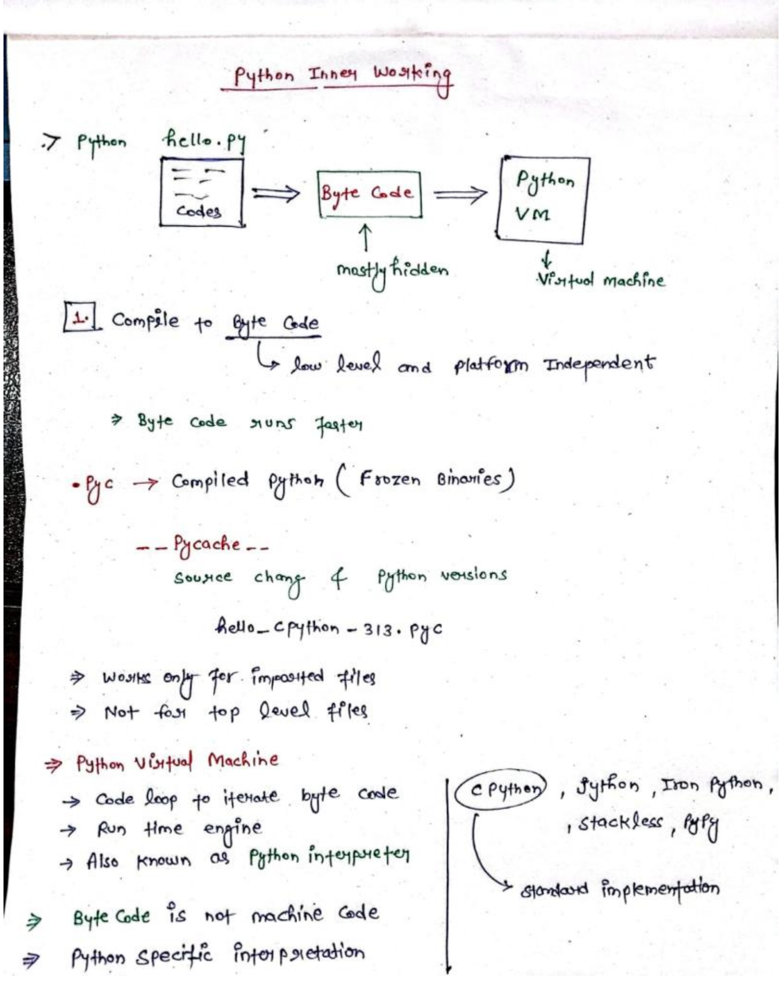
```
.py (source code)
   ↓ Compile
.pyc (bytecode)
   ↓ Interpret
PVM executes instructions
```

🔄 Summary of Flow:

```
Your Code (.py)
   ↓   [Compilation]
Bytecode (.pyc)
   ↓   [Interpretation]
Python Virtual Machine (PVM)
   ↓
Program Output (on screen)
```

✅ Why This Is Cool

- You don’t need to compile manually.
- Python handles both compiling & interpreting **under the hood**.
- That’s why Python is called a **“compiled + interpreted” language**.



Python Inner Working

⇒ Python hello.py

Codes ⟹ Byte Code ⟹ Python VM

↑ mostly hidden

↓ Virtual machine

1. Compile to Byte Code

↳ low level and platform Independent

⇒ Byte code runs faster

- Pyc → Compiled python (Frozen Binaries)

--Pycache--

source chang & Python versions

hello_cpython-313.pyc

⇒ Works only for imported files

⇒ Not for top level files

⇒ Python Virtual Machine

→ Code loop to iterate byte code

→ Run time engine

→ Also known as Python interpreter

⇒ Byte Code is not machine Code

⇒ Python Specific interpretation

CPython , Jython , Iron Python , , stackless , PyPy

→ standard implementation