Coding Introduction:

- We need to write python programs in notepad (good approach for practice).
- We can save this program with .py or .python extension.
- Run or execute the program.
- Finally, we will get output.

```
demo1.py

print("Hello World")

Run:  py demo1.py

Output: Hello World
```

Program flow:

- It's really important to understand internal python program flow
- This answer mainly contains three parts,
  - Write and Run the program
  - Compiled python file
  - Python Virtual machine

Write and run the program:

- Python program (source code), we need to save with .py (dot py) or .python (dot python) extension

Compiled python file:

- While running the program, the first step is, internally python compiler takes this source code and creates corresponding compiled python file to source code.
  - This compiled python file is not visible.
  - It is a hidden file, stored in cache memory.
  - If we want to see this file then run this below command,
  - python –m py_compile demo.py
  - -m means module here py_compile is module name
  - This module creates compiled python file.
  - So, python creates separate folder in the current directory by the name _pycache_ to store compiled python file.

Python Virtual Machine:

- Python Virtual Machine is a predefined program.
- This plays main role in python programming while running the program

- Compiled python file (demo.pyc or demo.cpython) contains byte code instructions.
- These byte code instructions are not understandable by the microprocessor to generate output.
- So, Python Virtual Machine takes responsible to convert these byte code instructions into machine understandable format.
- Finally, we will see the output.

To see all python keywords

demo2.py

```
import keyword
print(keyword.kwlist)
```

Output:

```
['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class',
'continue', 'def', 'del',
'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if',
'import', 'in', 'is', 'lambda',
'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while',
'with', 'yield','async','await']
```