Understanding the Execution of Python Program

Let understand how the execution of the Python program happens. Let’s consider the below example.

Example:

```
a = 10
b = 10
print("Sum ", (a+b))
```

Output:

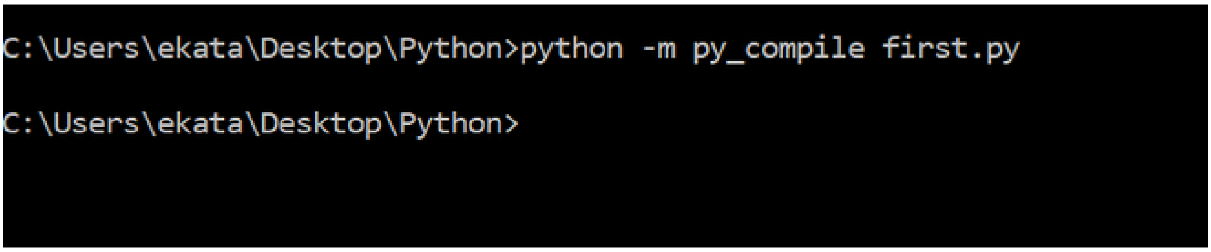
Sum 20

Suppose the above python program is saved as **first.py**. Here first is the name and .py is the extension. The execution of the Python program involves 2 Steps:

- Compilation
- Interpreter

Compilation

The program is converted into **byte code.** Byte code is a fixed set of instructions that represent arithmetic, comparison, memory operations, etc. It can run on any operating system and hardware. The byte code instructions are created in the **.pyc** file. The .pyc file is not explicitly created as Python handles it internally but it can be viewed with the following command:



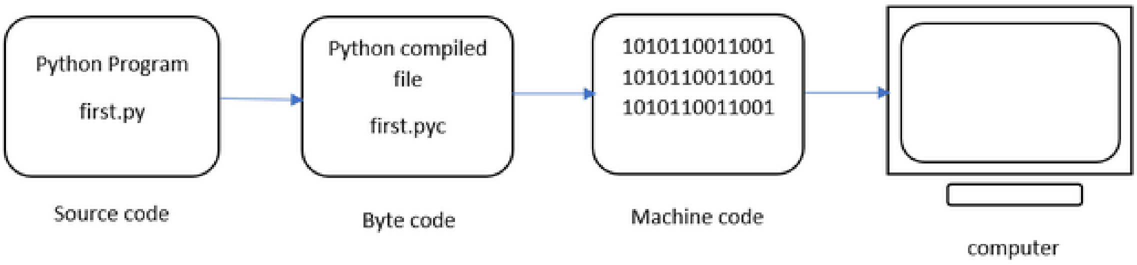
```
C:\Users\ekata\Desktop\Python>python -m py_compile first.py

C:\Users\ekata\Desktop\Python>
```

-m and py_compile represent module and module name respectively. This module is responsible to generate .pyc file. The compiler creates a directory named __pycache__ where it stores the first.cpython-38.pyc file.

Interpreter

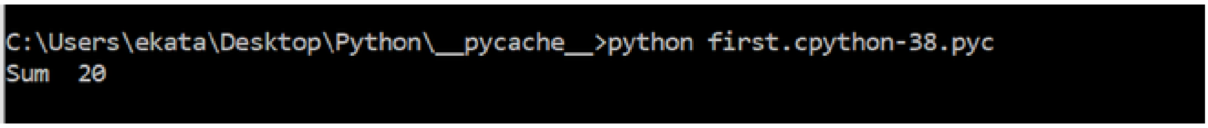
The next step involves converting the byte code (.pyc file) into machine code. This step is necessary as the computer can understand only machine code (binary code). Python Virtual Machine (PVM) first understands the operating system and processor in the computer and then converts it into machine code. Further, these machine code instructions are executed by processor and the results are displayed.



Execution of Python Program

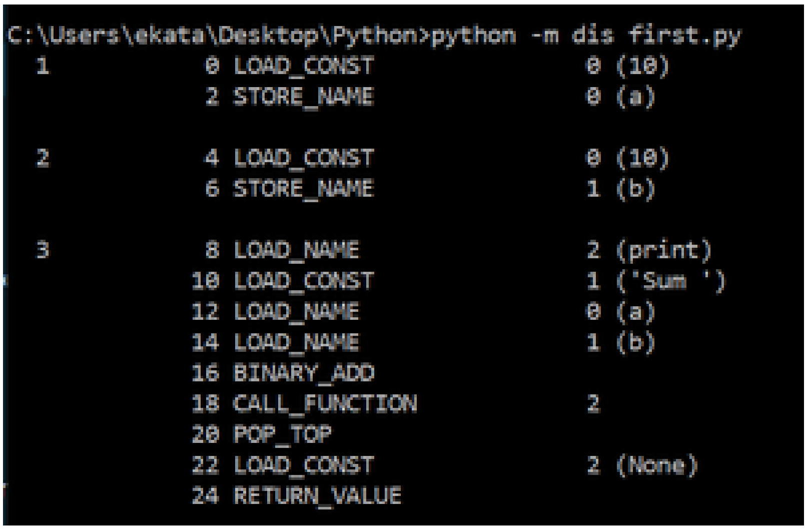
However, the interpreter inside the PVM translates the program line by line thereby consuming a lot of time. To overcome this, a compiler known as Just In Time (JIT) is added to PVM. JIT compiler improves the execution speed of the Python program. This compiler is not used in all Python environments like CPython which is standard Python software.

To execute the first.cpython-38.pyc we can use the following command:



```
C:\Users\ekata\Desktop\Python\__pycache__>python first.cpython-38.pyc
Sum  20
```

To view the byte code of the file – first.py we can type the following command as :



```
C:\Users\ekata\Desktop\Python>python -m dis first.py
  1           0 LOAD_CONST               0 (10)
              2 STORE_NAME               0 (a)

  2           4 LOAD_CONST               0 (10)
              6 STORE_NAME               1 (b)

  3           8 LOAD_NAME                2 (print)
             10 LOAD_CONST               1 ('Sum ')
             12 LOAD_NAME                0 (a)
             14 LOAD_NAME                1 (b)
             16 BINARY_ADD
             18 CALL_FUNCTION            2
             20 POP_TOP
             22 LOAD_CONST               2 (None)
             24 RETURN_VALUE
```

The **dis** command is known as “disassembler” that displays the byte code in an understandable format. The code represents 5 columns:

1. Line Number
2. offset position of byte code
3. name of byte code instruction
4. instruction’s argument
5. constants or names (in brackets)