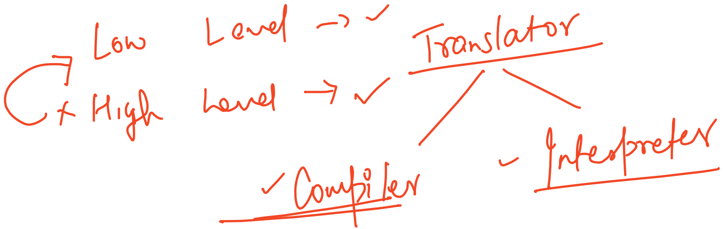
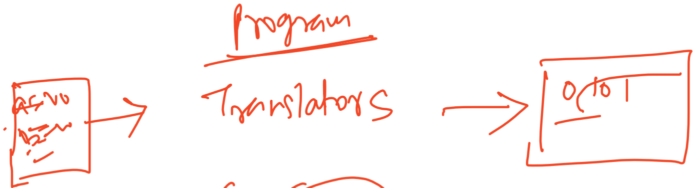
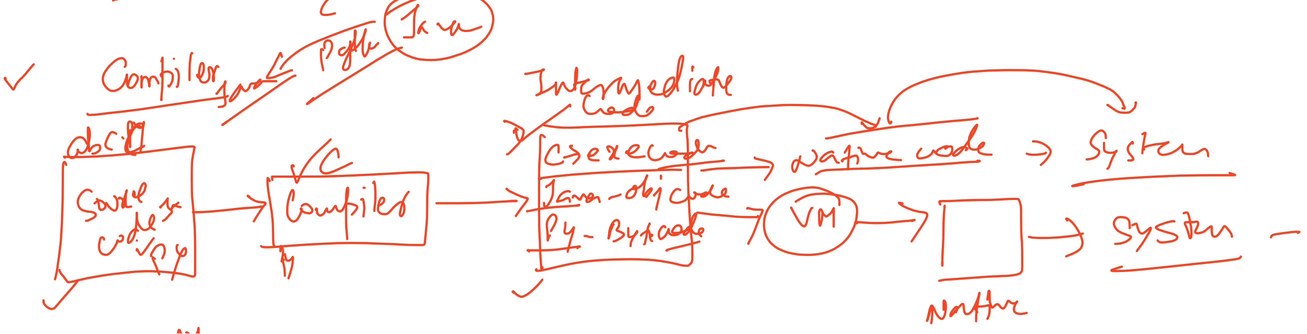
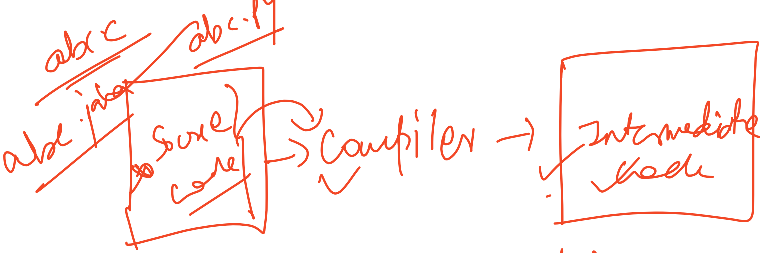

Low Level → ✓
High Level → ✓
Translator
✓ Compiler
✓ Interpreter


Program
Translators
0/1 0 1


C
Pythn
Java
✓ Compiler
Java
abc.c
Source code .c .py
✓C
Compiler
Intermediate Code
C exe code
Java-obj code
Py-Bytecode
native code
System
VM
System
Native

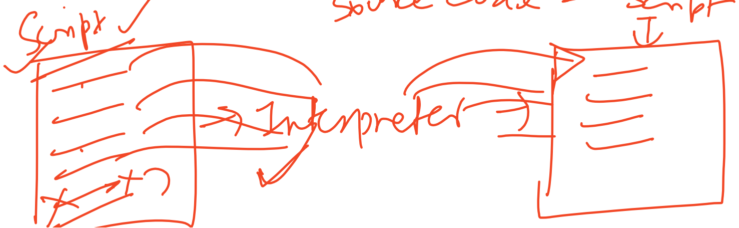


① Compiler compiles whole source code to Intermediate code

② Intermediate code is mandatory

③ If you have any error in source code the ~~Byte code~~ intermediate code will not be generated

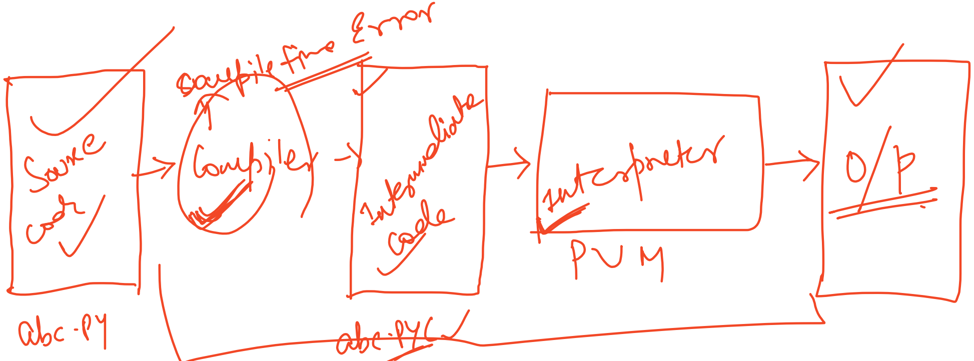
Source code ≈ Script



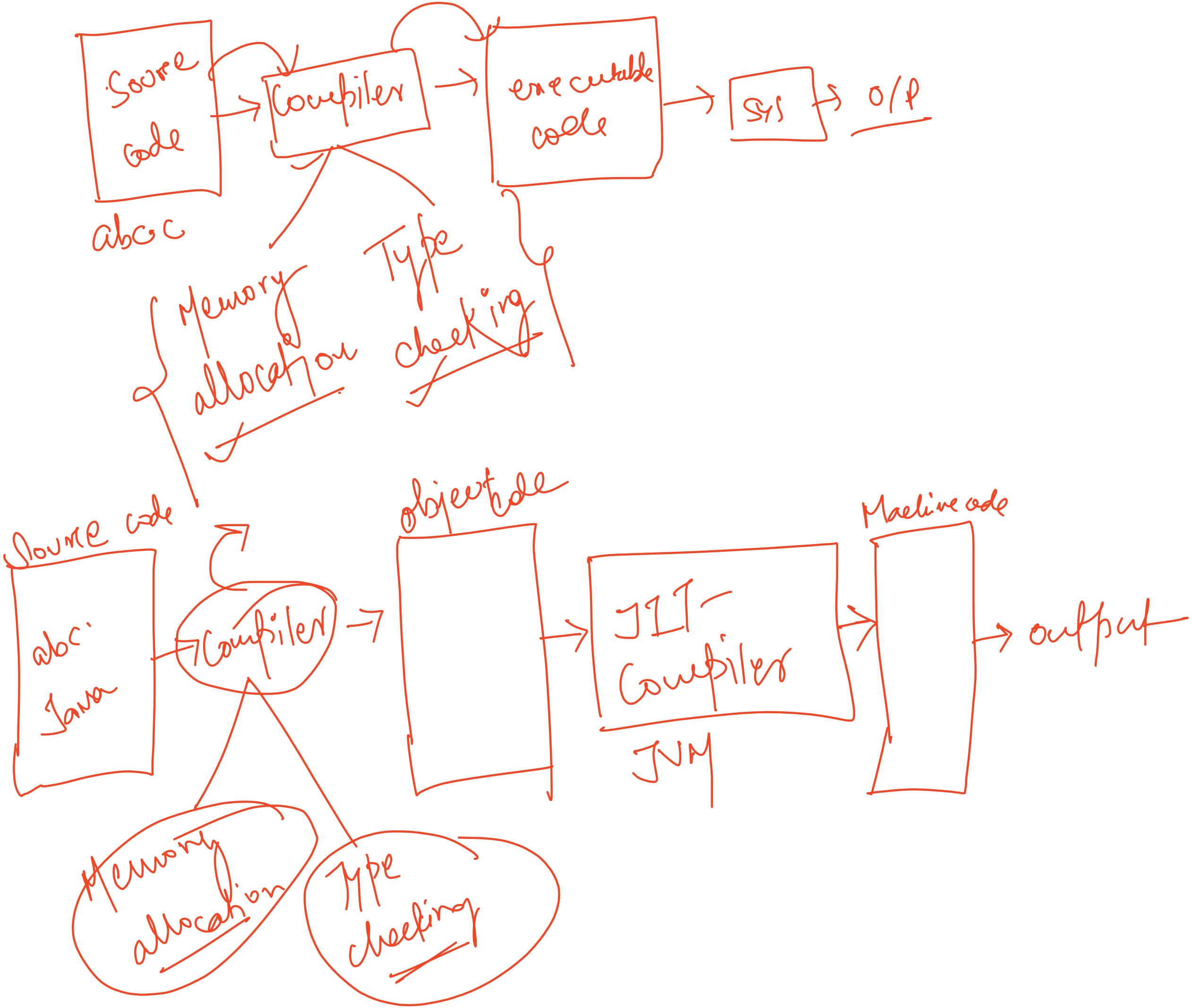
① Interpreter is line by line / instruction by instruction execution

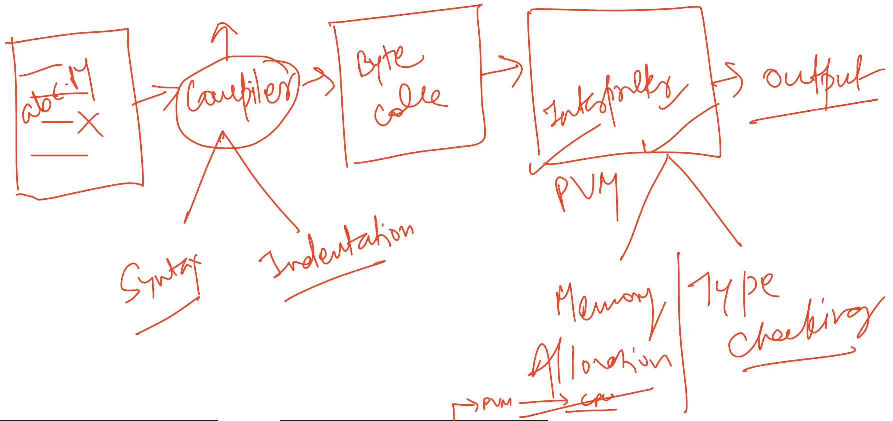
② Intermediate code is not generated

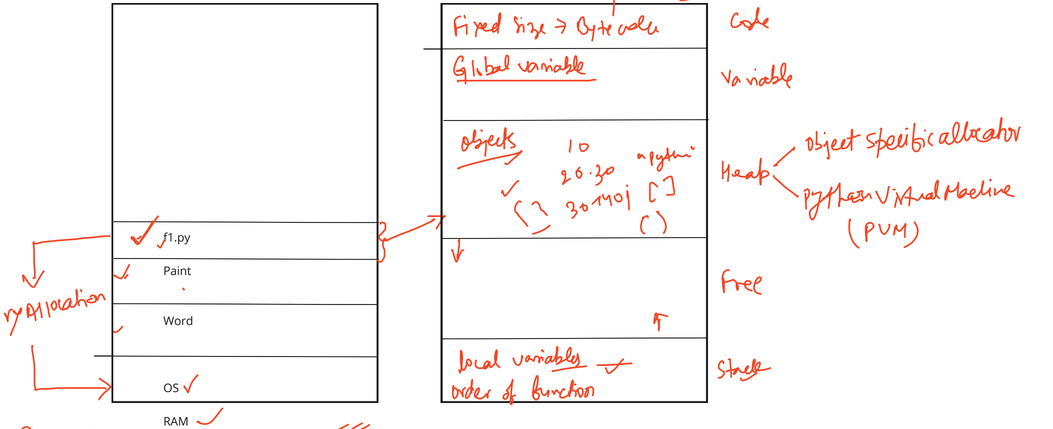
③ If ~~yo~~ Interpreter finds any error then it stops the execution then



Execution sequence of Pytho code

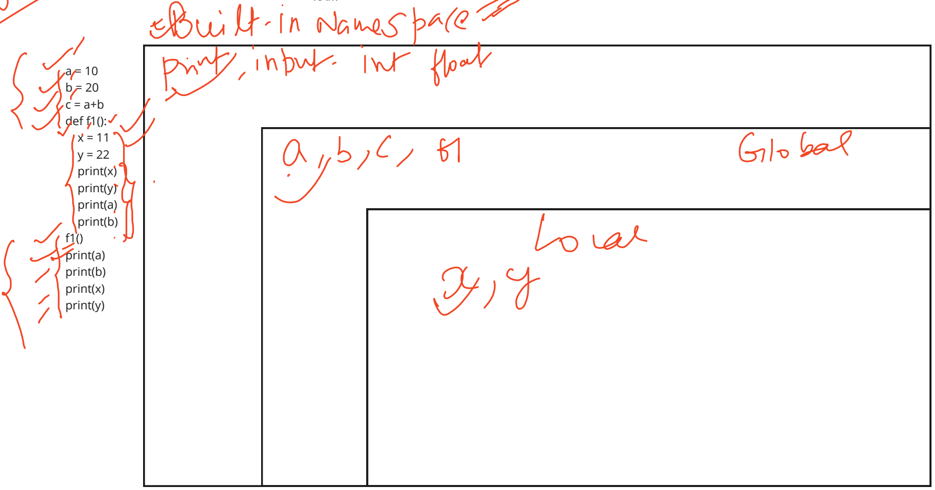

Source code
abc.c
Compiler
executable code
sys
o/p
Memory allocation
Type checking
Source code
abc. Java
Compiler
object code
JIT Compiler
JVM
Machine code
output
Memory allocation
Type checking





Global variable

A variable defined outside of the function



```
a = 10
b = 20
c = a+b
def f1():
    x = 11
    y = 22
    print(x)
    print(y)
    print(a)
    print(b)
f1()
print(a)
print(b)
print(x)
print(y)
```

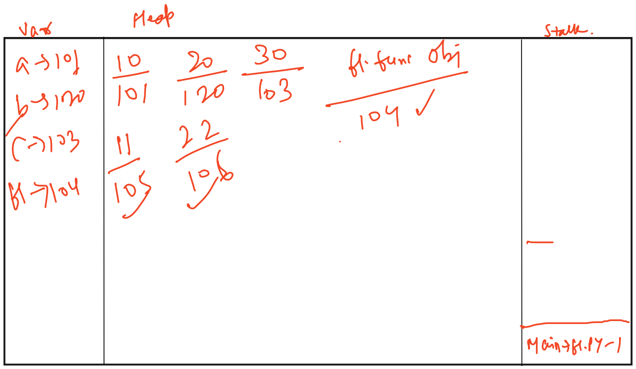
Scope - Rule X <-X 0 ---> ✓

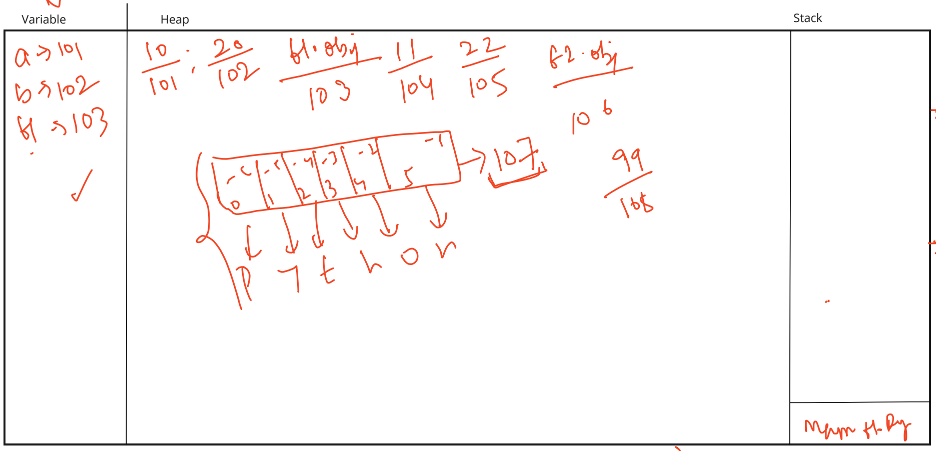
Local → Global → Built-in → Name Error

10+20
101+120

```
f1.py
a = 10
b = 20
c = a+b
def f1():
    x = 11
    y = 22
    print(x)
    print(y)
    print(a)
    print(b)
f1()
print(a)
print(b)
print(x)
print(y)
```

11
22
10
20
10





```
f1.py
a = 10
b = 20
def f1():
    a = 11
    x = 22
    def f2():
        b = 'Python'
        y = 99
        print(y,b,x,a)
    f2()
f1()
print(a,b)
print(x,y)
```

99
Python
22
11

10
20

Error