**SCOPES**

**Defination:-** Scope is the area in a program where a variable is accessible or visible.

Python uses the LEGB Rule to decide the scope of variables:

L → Local → Inside the current function  
E → Enclosing → In the outer function (for nested functions)  
G → Global → At the top level of the script/module  
B → Built-in → Python’s reserved names (like len, print)

**1.Local Scope:-** Variables declared inside a function can be accessed only inside that function.

**Example:-**

def demo():

y = 20 # Local variable

print("Inside function:", y)

demo()

# print(y) # Error: y is not defined outside function

**2.Global Scope:-** Variables declared outside all functions.

**Example:-**

x = 10 # Global variable

def show():

print("Inside function:", x)

show()

print("Outside function:", x)

**3. Enclosing Scope (Nested Functions**):- When a function is inside another function, the inner function can access outer function variables.

**Example:-**

def outer():

a = 30

def inner():

print("Inner accessing outer variable:", a)

inner()

outer()

**4.** **Global Keyword:-** Use global to modify global variables inside a function.

**Example:-**

count = 0

def increment():

global count

count += 1

print("Inside function:", count)

increment()

increment()

print("Outside function:", count)