Local Variable , Global variable and Scope of the variable

- **Local Variable:**
  - **A variable which is declared with in a function the variable can be used only in that function**

```
#Example Localvariable
def fun1():
   x=100  #local variable
   print("Fun1 x : ",x)

def fun2():
   print("Fun2 x : ",x)

#calling
fun1( )
fun2()
```

Global Variable:

- **A variable which declared outside of all the function called global variable**
- **Global variable we can use any function**

Example:

```
#Global Variable
z=333
def fun1():
   x=111  #local variable
   print("Fun1 x : ",x)
   print("Fun1 z : ",z)
def fun2():
   print("Fun2 :")
   print("z val is :",z)
#calling
fun1()
fun2()
print("From outside of all function")
print("z val is : ",z)
```

Note :

- ***Whenever you want perform any expression on the global variable in function scope , Then we need make the python interpreter to understand it is global variable by using "global" keyword otherwise***

it will be considered as local variable without value, thus it will raise an Error " UnboundLocalError"

```
x=10 #global variable
def fun1():
    global x
    x=x+20
    print("Fun1 ")
    print("Result is : ",x)
#calling
fun1()
```

Note :

- If both local and global declared with the same name then priority is given to local variable only.
- If you want access the global variable then we have to user " globals() -> dict "

```
x=10 #global variable
```

```
def fun1():
    x=20 #local variable
    print("x val is : ",x)
    d=globals()   # d={'x':10}
    print("global is : ",d['x'])
    print("global is : ",globals()['x'])

#calling

fun1()
```

closure

- A Function which is defined inside of outer function simply an inner function

Example:

```
def OuterFunc():
    x=10   #closure variable
    def InnerFunc():
        print("Inner Function ",x)
    InnerFunc()
```

```
#calling
OuterFunc()
```

Note: Whenever you want perform any expression on closure variable in the local scope then we have to make PVM to understand. It is a non local variable by using "nonlocal" keyword. Otherwise we will get "UnboundLocalError"

Example:

```
def OuterFunc():
    x=10   #closure variable
    def InnerFunc():
        x=x+10  #Error
        print("Inner Function ",x)
    InnerFunc()

#calling
OuterFunc()
```

Example 3:

Example:

```
def OuterFunc():
    x=10   #closure variable
    def InnerFunc():
        nonlocal x
        x=x+10
        print("Inner Function ",x)
    InnerFunc()

#calling
OuterFunc()
```

Nested Decorator Example:

```
def dec_split(func):
    def inner():
        return func().split()
    return inner

def dec_upper(func):
    def inner():
        return func().upper()
    return inner

@dec_split
@dec_upper
def greetings():
    return "Hello MyDear "

#calling
s=greetings()
print("Result is : ",s)
```