7. SCOPE OF VARIABLES, CONSTANTS & FUNCTIONS

The scope of a variable, constant or function in a computer program is the place where it can be used.

There are two different types of scope that you need to know about:

- **Local**
  This means the variable, constant or function is only in scope and available for use inside the block of code where it is defined.

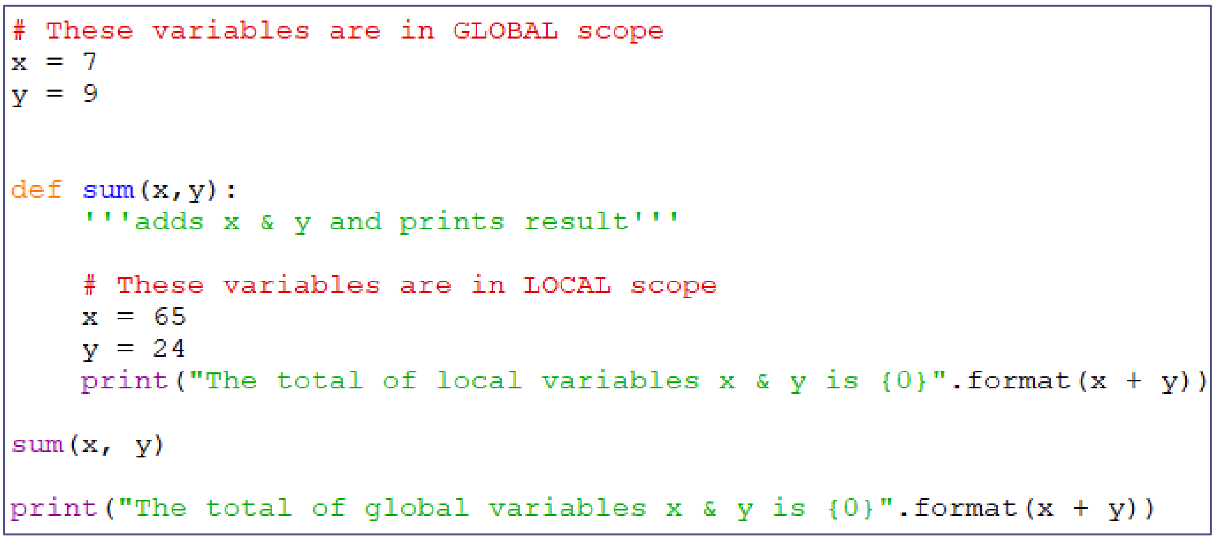
- **Global**
  This means the variable, constant or function is in scope and available for use throughout the program.

When you are writing programming code you should try to avoid using global variables as it is not good programming practice.

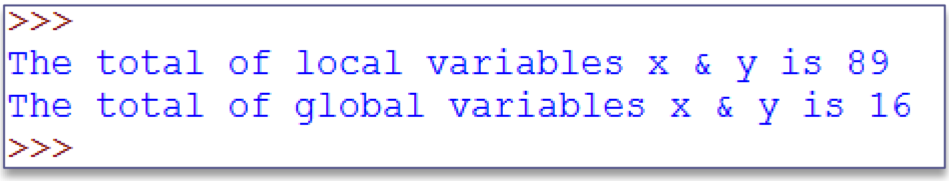
The first two variables assignments, x = 7 and y = 9, are in global scope for the whole of the program as they have been created outside of a function or procedure.

The second set of variables, x and y, are assigned values inside the function and have local scope in that block of code.

When we run the code, Python will always look to see whether there are local variables before using global variables.

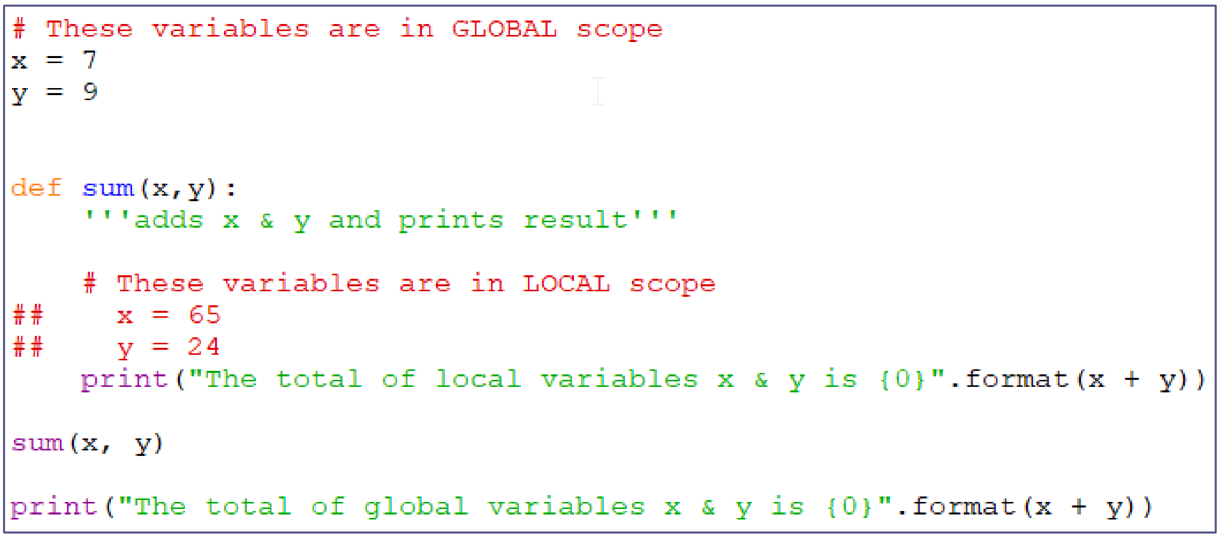


```python
# These variables are in GLOBAL scope
x = 7
y = 9


def sum(x,y):
    '''adds x & y and prints result'''

    # These variables are in LOCAL scope
    x = 65
    y = 24
    print("The total of local variables x & y is {0}".format(x + y))

sum(x, y)

print("The total of global variables x & y is {0}".format(x + y))
```



```
>>>
The total of local variables x & y is 89
The total of global variables x & y is 16
>>>
```

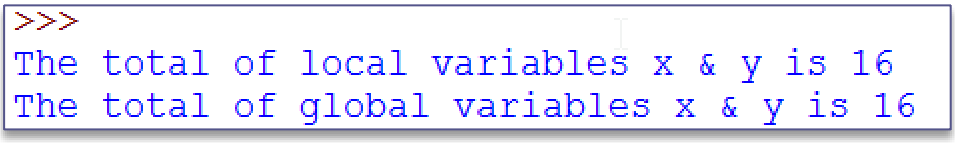
If I change the code to remove the local variable assignments for x and y, the answers will be the same:



```
# These variables are in GLOBAL scope
x = 7
y = 9


def sum(x,y):
    '''adds x & y and prints result'''

    # These variables are in LOCAL scope
##    x = 65
##    y = 24
    print("The total of local variables x & y is {0}".format(x + y))

sum(x, y)

print("The total of global variables x & y is {0}".format(x + y))
```

The local variable assignment is now commented out so when the code is executed, Python will not find any local variables inside the function. The function parameters, x and y, will be the global variables assigned at the start of the code.

The results for local and global scope are now the same, as the function uses the global variables:



```
>>>
The total of local variables x & y is 16
The total of global variables x & y is 16
```