SNS COLLEGE OF ENGINEERING
Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with ‘A’ Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF CSE

19IT103 – COMPUTATIONAL THINKING AND PYTHON PROGRAMMING

- **A readable, dynamic, pleasant, flexible, fast and powerful language**

Scope of Variables

Local and global Scope

- All variables in a program may not be accessible at all locations in that program. This depends on where you have declared a variable.
- The scope of a variable determines the portion of the program where you can access a particular identifier.
- There are two basic scopes of variables in Python
  1. Global variables
  2. Local variables

Scope of Variables

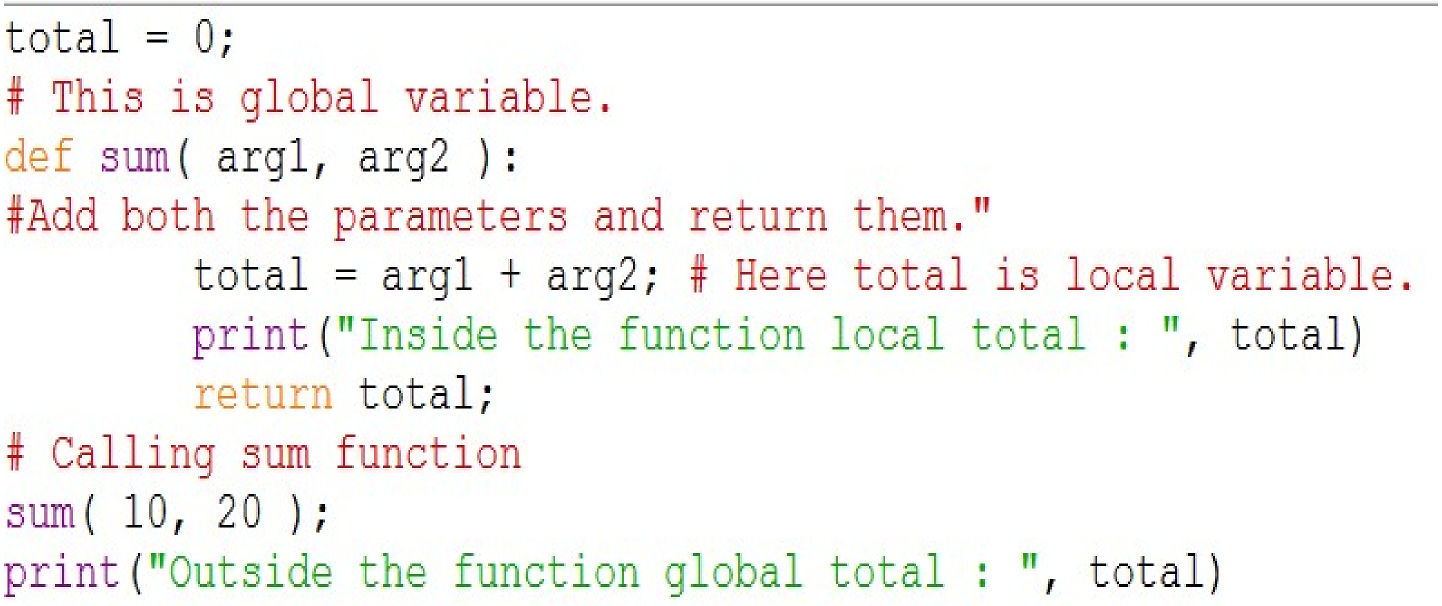
Local and global Scope

- Variables that are defined inside a function body have a local scope, and those defined outside have a global scope
- This means that local variables can be accessed only inside the function in which they are declared, whereas global variables can be accessed throughout the program body by all functions.
- When you call a function, the variables declared inside it are brought into scope.

Scope of Variables

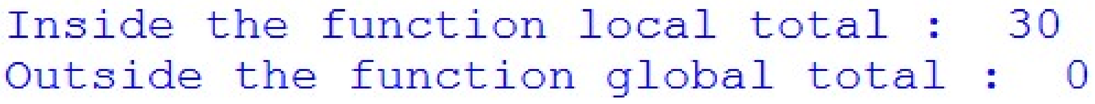
Local and global Scope

Example:



```
total = 0;
# This is global variable.
def sum( arg1, arg2 ):
#Add both the parameters and return them."
        total = arg1 + arg2; # Here total is local variable.
        print("Inside the function local total : ", total)
        return total;
# Calling sum function
sum( 10, 20 );
print("Outside the function global total : ", total)
```

Output:



```
Inside the function local total :  30
Outside the function global total :  0
```

Scope of Variables

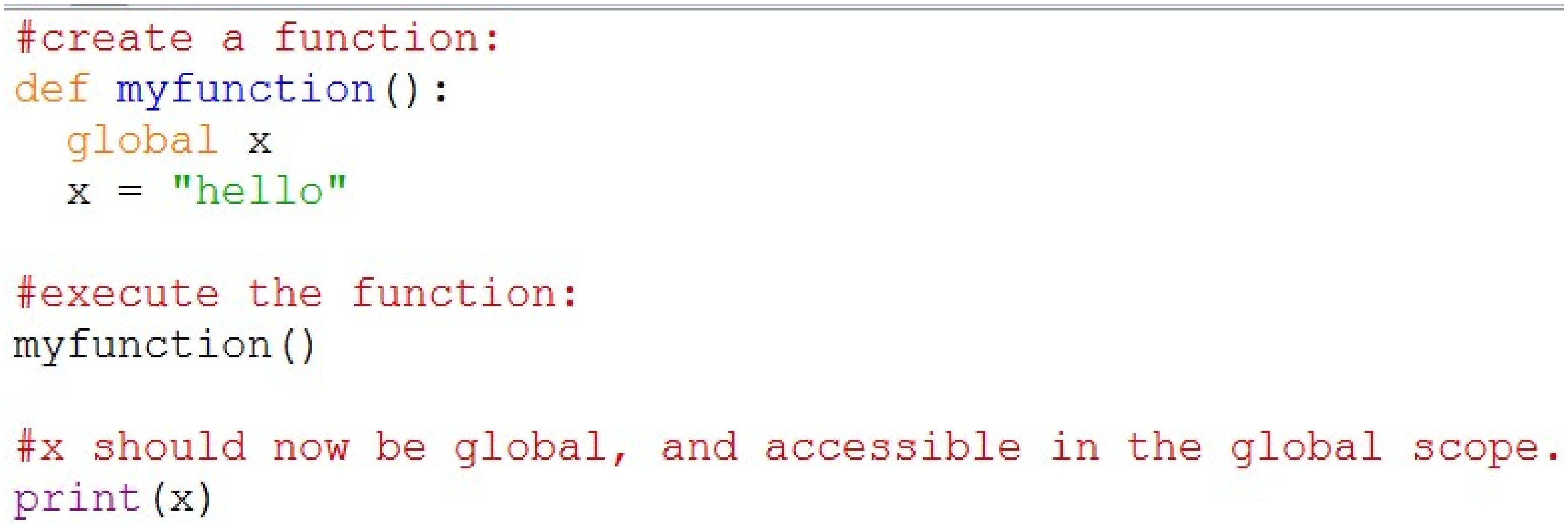
Local and global Scope

- In Python, global keyword allows you to modify the variable outside of the current scope.
- It is used to create a global variable and make changes to the variable in a local context.

Scope of Variables

“global” Keyword:

Example:

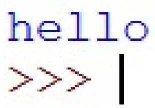


```
#create a function:
def myfunction():
  global x
  x = "hello"

#execute the function:
myfunction()

#x should now be global, and accessible in the global scope.
print(x)
```

Output:



```
hello
>>> |
```

Summary

- Values present in the function calling statement are called arguments
- Variables used in the function header are called parameters
- Required, keyword, default and variable-length are types of arguments
- Variable can be created with local and global scopes
- Global keyword creates a global variable inside a block

THANK YOU