

Python Programming

Presented by

Mariena A. A.

**Asst. Professor and Head, Department of Computer Science
Little Flower College, Guruvayoor.**

Global variables and local variables

The scope of Local variables are within the function

Scope of global variable are in entire program.

Variables declared above the user defined function is treated as global variables

a can be used in the entire program and outside the function call.

```
a=20
```

```
def display():
```

```
    b=30
```

```
    print("inside user defined function:" , a)
```

```
    display()
```

```
    print("outside the function:",a)
```

```
    print("the value is ", b) #
```

Local variables:

Variables used inside a user defined function is called local variables.

Example

```
a=10
def display():
    b=20
    print("global value", a) # 10
    print("local value", b) # 20
    display():
        c=40 # local variable of main function
        print("local", c) # 40
        print("global", a) # 10
```

Example

If local variable and global variables are having the same name then first preference should be given to local variables.

```
a=10
```

```
def display():
```

```
    a=20                # local variable of display function
```

```
    print("global value", a)    # 20
```

```
display():
```

```
    a=40                # local variable of main function
```

```
    print("local", a)        # 40
```

Example

```
a=10
c=20
def display():
    a=20                # local variable of display
function
    print("global value", a)    # 20
display():
    b=40                # local variable of main function
    print("local", b) #40
    d=a+c
    print(d)
```

what is the output? 30

Formatted output

We can customize the view of output as our wish

The important format strings are

% d integer

%f float

%s string

Example:

a= 4567 %4d output :4567

b=2.567 %3.2f output: 2.56

c="meena" %5s output: meena

Arguments

Arguments are four types

1.Required arguments : Number of arguments should be same in both function call and function definition. Position or order of arguments should be same.

```
def display(a,b):# function definition
```

```
print(a,b)
```

```
display(10,20) # function call
```

If you write in the different order

```
display(20,10) # a=20,b=10
```

2.Keyword arguments: Here order or position is not required. Initialization can be done based on keywords that means name of variables given in function call.

```
def display(a,b):
```

```
print(a,b)
```

```
display(b=20, a=10) # b is mapped with b only no position important
```

Default Arguments

3. Default Arguments: Number of arguments need not be same.

Some of the arguments will be considered as default arguments

```
def display(name, course =“Btech”)  
print(name)  
print(course)  
display( name=“abc”, course=“Mtech”)  
display(name=“pqr”)
```

Output:

abc, Mtech

pqr, Btech

Argument value given in the function definition is called default argument.

If it is not given in the function call the value given in the function definition is mapped with it.

Variable length Arguments

Variable length arguments:

The function definition access arbitrary number of arguments by placing * as prefix to the argument in the function definition.

```
def display(* courses) # it will be treated as number of values presented in courses variable
for i in courses
    print(i)
display("BCA","MCA","MBA","MBA")
```

Thank you!

