How to write a function

- We can write a function by using the keyword **def** followed by a function name.
- The function name is user define and it is suggested to take a meaningful name while defining a function
- The rules for defining a function is same as giving variable names
- Within the ( ) you can pass parameters to a function these Parameter are called **"Formal parameter"**
- Parameters are called input to a function , a function can take multiple Parameter.
- Parameters can be of any datatype
- Returning values of the statements is Calle **"output"**
- You can call a function by using the function name and pass parameter in ( ) , these Parameter are called **"Actual parameter"**
- The actual parameter values are copied into formal parameter which acts as input to a function they are copied in the same position / order
- When you call a functioning a result is returned you should place that result into another variable (or) print it directly
- If you don't write return in function it'll return **NONE**
- So, every function returns whether you write it or not.

Syntax :

```
def fun_name ( par1 , par2, par3 ) :          #Formal parameter
        Stat1
        Stat2
        ……                                    #Statements of function
        ……
    return result                             #Returning result

fun_name ( par1, par2, par3 )
return_value = fun_name ( apar1, apar2, apar3 )        #Calling a function
```

A simple example to understand function.

```
def add3(a, b, c):  # defining a function
    r = a + b + c  # writing statement inside function
    return r       # returning result
print(add3(10, 15, 5))  # printing the result
r = add3(1, 3, 5)
print(r)  # printing and taking the result in r
```

Output :

```
30
9
```

- The values in formal parameter acts as a pointer to actual parameter
- therefore they'll be referring to the same thing
- Lets understand this with an

```
def add3(a, b, c):
    print('inside function', id(a), id(b), id(c))  # printing id of 3 var
                                                   # i.e, a,b,c

x, y, z = 10, 15, 5  # declaring 3 var i.e, x, y, z
print('outside function', id(x), id(y), id(z))  # printing id of x,y,z
print(add3(x, y, z))  # calling function

# outside function gives id of actual parameter
# Inside function gives id of formal parameter
```

Output:

```
outside function 4446765584 4446765744 4446765424
inside function 4446765584 4446765744 4446765424
None
```

- In python object are always pass just like reference only , copy of an object will never be pass