- **Default Value Parameters**

```
def printInfo(name, age=35):
    print("Name:", name)
    print("Age:", age)

printInfo("Ali", 50)
printInfo("Ali")
#printInfo()  #Error in Name (without value)
```

- **Types of Function Arguments**
  1. Required arguments.
  2. Keyword arguments:

     With keyword arguments in python, we can change the order of passing the arguments without any consequences.
  3. Variable-length arguments:

     Sometimes you might want to define a function that can take any number of parameters.

Required arguments	Keyword arguments	Variable-length arguments

def sum(a, b):
  total = a + b
  print(total)
 	
sum(10, 20) ✓
sum(10, 20, 20) **Error**
sum(10) **Error**
	def subtract(a, b, c):
	  print(a – b – c)
	
	subtract(1, 2, 3) # Output: -4
	subtract(b=3, c=1, a=2) # Output: -2
	subtract(3, c=2, b=1) # Output: 0
		def add(*numbers):
		  result = 0
		  for i in numbers:
		  result += i
		  print(result)
		
		add(1, 2, 3)
		# Output: 6
		add(1, 2, 3, 4, 5)
		# Output: 15

- **Returning Multiple Value**

```
def square(x, y, z):
    return x*x, y*y, z*z
a, b, c = square(2, 3, 4)
print(a, b, c)
# Output: 4   9   16
```

- **Functions can Call Other Functions**

```
def square(x):
    return x * x
def sum_of_squares(x, y, z):
    a = square(x)
    b = square(y)
    c = square(z)
    return a + b + c
result = sum_of_squares(-5, 2, 10)
print(result)  # Output: 129
```

- **Functions insides functions:**

```
def a ( ):
    def b ( ):
        print("Hello")
    print("CS Level1")  # print this then calling b
    b ( )
a ( )   #start
```

- **scopes of variables (Identifier Scope):**

1. Global variables : defined outside a function.
2. Local variables : defined inside a function.

X = 100 **# X is global**
def sum(Y):
  Z = X + Y **# Z , Y local**
  print(Z)
sum(5)
 	
#105
	def sum(Y):
	  Z = X + Y
	  print(Z)
	X = 100
	sum(5)
	
	#105
		def sum(Y):
		  Z = X + Y **#'X' is not defined**
		  print(Z)
		sum(5)
		X = 100
		
		#Error