FUNCTION ARGUMENTS

- The argument is a value that is passed to the function when it's called.
- Parameters are useful to receive values from outside of the function. They are called **'formal parameters'**.
- When we call the function, we should pass data or values to the function. These values are called **'actual arguments'.**

Example:

```
def sum (a, b) #formal arguments
        c=a+b
        print(c)

x=10; y=15
sum(x,y) #actual arguments
```

- The actual arguments used in a function call are of four types:

1. Positional Arguments
2. Keyword Arguments
3. Default Arguments
4. Variable Length Arguments

1. Positional Arguments

- Passed to a function in correct positional order.

Example:

```
def match(s1, s2):
        s3=s1+s2
        print ("Total String:"+s3)
match ('Good', 'Morning') #Positional Arguments
```

2. Keyword Arguments

- Arguments that identify the parameters by their names.
- We can change the order of the arguments.
- Keyword arguments must follow positional arguments.

Example:

```
def grocery (item, price):
        print ('Item = %s' %item)
        print ('Price = %f' %price)

grocery (item='Milk', price=28.50) #keyword Arguments
grocery (price=90.0, item='sugar')
```

3. Default Arguments

- Some default value for the function parameters in the definition.
- Assignment operator (=).
- Non-default arguments cannot follow default arguments.

Example: Def grocery(item, price=30.0):

```
def greet (name, msg='Good Morning'):
        print ("Hello", name + ',' +msg)
greet ("Pooja")
greet ("Ram", "How do you do? ")

def greet (msg='Good Morning', name): # Error
```

4. Variable Length Arguments or Python Arbitrary Arguments

- We don’t know in advance the number of arguments that will be passed into a function.
- That can accept any number of values.
- In the function definition, we can use asterisk (*) before the parameter name to denote this kind of argument.

```
Def add(fargs, *args)
```

Here, ‘fargs’ is the formal argument and ‘*args’ represents variable length argument.

Example:

```
def greet(*names):
        #names is a tuple with argumnets
        for name in names:
                print("Hello",name)

greet("Monica","Naman","Dhruv","Vrushika","Pooja")
```