When we define and call a Python function, the terms parameter and argument are used to pass on information to the function.

parameter: It is the variable that is mentioned in the function definition's parentheses.

argument: It is a value that is passed to the function when it is called. It is the data regarding variables on which a function performs and returns a result.

Types of function arguments.

- Default argument
- Keyword arguments (named arguments)
- Positional arguments
- Arbitrary arguments (variable-length arguments *args and **kwargs)

Positional arguments:

Positional arguments are those arguments where values get assigned to the arguments by their position when the function is called.

By default, Python functions are called using the positional arguments.

Try it yourself:

Example 1:

```
def student_info(name,age,marks):
    print(" name : {} \n age : {} \n marks {}".format(name,age,marks))


student_info("pratik",20,90)
```

Output:

```
name : pratik
 age : 20
 marks 90
```

```
def is_even(x):
```

```
    if x%2==0:
        return True
    else:
        return False


print(is_even(10))
```

Output:

True

Default argument:

Arguments in a function can have default values. At the time of function definition, we use the '=' operator to assign default values to the arguments. A function can have an n number of default arguments.

If we don't provide a value to an argument at the time of the function call, the default value will be used inside the function. As a result, the default arguments are not required for the function call.

If we provide a value to the default parameters during function calls, it overrides the default value.

Try it yourself:

```
def student_info(name,age,marks,city="Mumbai",institute="xyz"):
     print(" name : {} \n age : {} \n marks {} \n city :{} \n
institute:{}".format(name,age,marks,city,institute))

# without passing city and institute
# Passing only the mandatory arguments
student_info("pratik",20,80)

#assign all arguments
student_info("Meera",20,95,"Pune","Itvedant")
```

Output:

name : pratik

age : 20

marsk 80

city :Mumbai

institute:xyz

name : Meera

age : 20

marks 95

city :Pune

institute:Itvedant

Keyword arguments (named arguments):

Keyword arguments are those arguments where values get assigned to the arguments by their keyword (name) when the function is called. It is followed by the variable name and an assignment operator (=). The Keyword Argument is also known as a named argument.

Note: Keyword arguments will be linked to dictionaries in that they map a value to a keyword.

Try it yourself:

```
def student(name, age):
    print('Student Details:', name, age)

# default function call
student('Shikha', 18)

# both keyword arguments
student(name='meera', age=17)

# 1 positional and 1 keyword
student('rohit', age=19)
```

Output:

Student Details: Shikha 18

Student Details: meera 17

Student Details: rohit 19

Arbitrary arguments (variable-length arguments *args and **kwargs):

In Python, we may need to pass multiple arguments to a function at times. Such arguments are known as arbitrary arguments or variable-length arguments.

If we don't know how many arguments the function will require in advance, we use variable-length parameters.

Types of Arbitrary Arguments:

arbitrary positional arguments (*args):

Example:

Try it yourself:

```
def percentage(*args):
    sum = 0
    for i in args:
        # get total
        sum = sum + i
    # calculate average
    avg = sum / len(args)
    print('Average =', avg)

percentage(56, 61, 73)
```

Output:

Average = 63.33

arbitrary keyword arguments (**kwargs)

Try it yourself:

```
def percentage(**kwargs):
    sum = 0
    for sub in kwargs:
        # get argument name
        sub_name = sub
```

```
        # get argument value
        sub_marks = kwargs[sub]
        print(sub_name, "=", sub_marks)

# pass multiple keyword arguments
percentage(math=56, english=61, science=73)
```

Output:

```
math = 56
english = 61
science = 73
```