

PARAMETERS AND ARGUMENT IN PYTHON

Parameters are variables that are defined in the function definition. They are assigned the values which were passed as arguments when the function was called, elsewhere in the code.

5 ARGUMENTS IN PYTHON TO KNOW

default arguments

keyword arguments

positional arguments

arbitrary positional arguments

arbitrary keyword arguments

Default arguments are values that are provided while defining functions.

The assignment operator = is used to assign a default value to the argument.

Default arguments become optional during the function calls.

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

The function can have any number of default arguments.

Default arguments should follow non-default arguments.

Keyword Arguments in Python

Functions can also be called using keyword arguments of the form `kwarg=value`.

During a function call, values passed through arguments don't need to be in the order of parameters in the function definition. This can be achieved by keyword arguments. But all the keyword arguments should match the parameters in the function definition.

```
def add(a,b=5,c=10):  
    return (a+b+c)  
print (add(b=10,c=15,a=20))  
#Output:45
```

Positional Arguments in Python

During a function call, values passed through arguments should be in the order of parameters in the function definition. This is called positional arguments.

Keyword arguments should follow positional arguments only.

```
def add(a,b,c):  
    return (a+b+c)
```


The above function can be called in two ways:

First, during the function call, all arguments are given as positional arguments. Values passed through arguments are

passed to parameters by their position. 10 is assigned to a, 20 is assigned to b and 30 is assigned to c.

```
print (add(10,20,30))  
#Output:60
```

IMPORTANT POINTS TO REMEMBER

- 
- 1.default arguments should follow non-default arguments
 - 2.keyword arguments should follow positional arguments
 - 3.All the keyword arguments passed must match one of the arguments accepted by the function and their order is not important.
 - 4.No argument should receive a value more than once
 - 5.Default arguments are optional arguments.

What Are Variable-Length Arguments in Python?

Variable-length arguments are also known as arbitrary arguments. If we don't know the number of arguments needed for the function in advance, we can use arbitrary arguments

THERE ARE TWO TYPES OF ARBITRARY ARGUMENTS:

1. Arbitrary positional arguments.
2. Arbitrary keyword arguments.

ARBITRARY POSITIONAL ARGUMENTS IN PYTHON

For arbitrary positional argument, an asterisk (*) is placed before a parameter in function definition which can hold non-keyword variable-length arguments. These arguments will be wrapped up in a tuple. Before the variable number of arguments, zero or more normal arguments may occur.

```
def add(*b):  
    result=0  
    for i in b:  
        result=result+i  
    return result
```

```
print (add(1,2,3,4,5))  
#Output:15  
print (add(10,20))  
#Output:30
```

ARBITRARY KEYWORD ARGUMENTS IN PYTHON

For arbitrary positional argument, a double asterisk (**) is placed before a parameter in a function which can hold keyword variable-length arguments.

```
def fn(**a):  
    for i in a.items():  
        print (i)
```

```
fn(numbers=5,colors="blue",fruits="apple")
```

```
'''
```

Output:

```
('numbers', 5)
```

```
('colors', 'blue')
```

```
('fruits', 'apple')
```

```
'''
```