

KEYWORD ARGUMENTS

If you have some functions with many parameters and you want to specify only some of them, then you can give values for such parameters by naming them - this is called **keyword arguments** - we use the name (keyword) instead of the position (which we have been using all along) to specify the arguments to the function.

There are two advantages - one, using the function is easier since we do not need to worry about the order of the arguments. Two, we can give values to only those parameters to which we want to, provided that the other parameters have default argument values.

Example (save as `function_keyword.py`):

```
def func(a, b=5, c=10):  
    print 'a is', a, 'and b is', b, 'and c is', c  
  
func(3, 7)  
func(25, c=24)  
func(c=50, a=100)
```

Output:

```
$ python function_keyword.py  
a is 3 and b is 7 and c is 10  
a is 25 and b is 5 and c is 24  
a is 100 and b is 5 and c is 50
```

How It Works

The function named `func` has one parameter without a default argument value, followed by two parameters with default argument values.

In the first usage, `func(3, 7)`, the parameter `a` gets the value 3, the parameter `b` gets the value 7 and `c` gets the default value of 10.

In the second usage `func(25, c=24)`, the variable `a` gets the value of 25 due to the position of the argument. Then, the parameter `c` gets the value of 24 due to naming i.e. keyword arguments. The variable `b` gets the default value of 5.

In the third usage `func(c=50, a=100)`, we use keyword arguments for all specified values. Notice that we are specifying the value for parameter `c` before that for `a` even though `a` is defined before `c` in the function definition.

Source: <http://www.swaroopch.com/notes/python/>