

FUNCTION PARAMETERS

A function can take parameters, which are values you supply to the function so that the function can **do** something utilising those values. These parameters are just like variables except that the values of these variables are defined when we call the function and are already assigned values when the function runs.

Parameters are specified within the pair of parentheses in the function definition, separated by commas. When we call the function, we supply the values in the same way. Note the terminology used - the names given in the function definition are called **parameters** whereas the values you supply in the function call are called **arguments**.

Example (save as function_param.py):

```
def print_max(a, b):  
    if a > b:  
        print a, 'is maximum'  
    elif a == b:  
        print a, 'is equal to', b  
    else:
```

```
print b, 'is maximum'
```

```
# directly pass literal values
```

```
print_max(3, 4)
```

```
x = 5
```

```
y = 7
```

```
# pass variables as arguments
```

```
print_max(x, y)
```

Output:

```
$ python function_param.py
```

```
4 is maximum
```

```
7 is maximum
```

How It Works

Here, we define a function called `print_max` that uses two parameters called `a` and `b`.

We find out the greater number using a simple `if..else` statement and then print the bigger number.

The first time we call the function `print_max`, we directly supply the numbers as arguments. In the second case, we call the function with variables as arguments. `print_max(x, y)` causes the value of argument `x` to be assigned to parameter `a` and the value of argument `y` to be assigned to parameter `b`. The `printMax` function works the same way in both cases.

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