

DEFAULT ARGUMENT VALUES

For some functions, you may want to make some parameters **optional** and use default values in case the user does not want to provide values for them. This is done with the help of default argument values. You can specify default argument values for parameters by appending to the parameter name in the function definition the assignment operator (=) followed by the default value.

Note that the default argument value should be a constant. More precisely, the default argument value should be immutable - this is explained in detail in later chapters. For now, just remember this.

Example (save as `function_default.py`):

```
def say(message, times=1):  
    print message * times  
  
say('Hello')  
say('World', 5)
```

Output:

```
$ python function_default.py  
Hello  
WorldWorldWorldWorldWorld
```

How It Works

The function named `say` is used to print a string as many times as specified. If we don't supply a value, then by default, the string is printed just once. We achieve this by specifying a default argument value of 1 to the parameter `times`.

In the first usage of `say`, we supply only the string and it prints the string once. In the second usage of `say`, we supply both the string and an argument 5 stating that we want to **say** the string message 5 times.

CAUTION

Only those parameters which are at the end of the parameter list can be given default argument values i.e. you cannot have a parameter with a default argument value preceding a parameter without a default argument value in the function's parameter list.

This is because the values are assigned to the parameters by position. For example, `def func(a, b=5)` is valid, but `def func(a=5, b)` is **not valid**.