

# Syntax Guide 2.2

## Function Definitions

A function is defined with the “def” keyword. After the “def” keyword, you type a space and what you want the function to be called (here “simpleFunction”). Then, between parentheses you list the parameters of the function which are like variables which hold whatever values are passed to the function. A function can have any number of parameters, including zero parameters. The return statement is responsible for describing what the output of the function will be.

```
def simpleFunction(foo, bar):  
    print("You called simpleFunction")  
    print(foo + bar)  
    return foo * bar
```

If you “call” this function in the “console” (the place to the right of the code area that shows your output text and drawings) with the following two inputs, you will see something like the following two outputs:

```
>>> x = simpleFunction(3, 4)  
You called simpleFunction  
7  
>>> x  
12
```

Here is an example function with no parameters:

```
import random  
def noParameters():  
    print("This function has no parameters")  
    print("It still returns a value, though")  
    return 6 * random.randint(1, 3)
```

And this function outputs the following things when used in the console:

```
>>> x = noParameters()  
This function has no parameters  
It still returns a value, though  
>>> x  
6  
>>> x = noParameters()  
This function has no parameters  
It still returns a value, though  
>>> x  
18
```