# FUNCTIONS

Functions are chunks of code that tell Python to do something. They are one way to reuse code – you can use the functions in your programs again and again.

Functions help to make your program more efficient, easier to debug (fix errors) and modify.

### Parts of a Function

A function has three parts: a name, parameters and a body.

### Example

```
def testfunc(myname):
            print('hello %s' % myname)
```

The name of this function is *testfunc*. It has a single parameter, *myname*, and its body is the block of code immediately following the line beginning with *def* (short for define). A parameter is a variable that exists only while a function is being used.

You can run a function by calling its name, using parentheses around the parameter value:

```
testfunc('Juan')
>>>
hello Juan
```

The function can take two, three, or any number of parameters, instead of just one. The two values for these parameters are separated by a comma:

```
def testfunc(fname, lname):
    print('Hello %s %s' % (fname, lname))
testfunc('Mia, 'Lee')
>>>
```

```
Hello Mia Lee
```

You could also create some variables first and then call the function with them:

```
firstname = 'Joe'
lastname = 'Smith'
testfunc(firstname, lastname)
>>>
Hello Joe Smith
```

A function is often used to return a value using a return statement. For example, you could write a function to calculate how much money you were saving:

```
def savings(pocket_money, paper_route, spending):
    return pocket_money + paper_route - spending
```


This function takes three parameters. It adds the first two (pocket_money and paper_route) and subtracts the last (spending). The result is returned and can be assigned to a variable, balance, which can then be printed.

```
balance = savings(10, 10, 5)
print(balance)
>>>
15
```

## Variables and Scope

A variable that is inside the body of a function can't be used again when the function has finished running because it exists only inside the function. In the world of programming, this is called *scope*.

## Example

```
def variable_test():
    first_variable = 10
    second_variable = 20
    return first_variable * second_variable


print(variable_test())
>>>
200
```

If we call this function using print, we get the result: 200. However, if we try to print the contents of first_variable or second_variable outsie of the block of the function, we get an error message:

```
print(first_variable)
>>>
```

Traceback (most recent call last):

File "<pyshell#50>", line 1, in <module>

    print(first_variable)

NameError: name 'first_variable' is not defined

If a variable is defined outside the function, it has a different scope. For example, let's define a variable before we create our function and then try using it inside the function:

```
another variable = 100

def variable_test2():

   first_variable = 10

   second_variable = 20

   return first_variable * second_variable * another_variable


print(variable_test2())

>>>

20000
```

In this code, eventhough the variables first_variable and second_variable can't be used outside the function, the variable, another_variable (which was created outside the function) can be used inside it.

**Practice Questions**

1. Write a program that calls a function num() which will output the numbers from 1 to 20.

2. Write a program to input two numbers. The program will then call a function, max_num, and pass the two numbers to the function. The function will compare the two numbers and return the larger of the two numbers.

3. Write a program that calls a function sum_average(). The function, sum_average will input three numbers from the user and return the sum and average of the three numbers.

4. Write a program to output a student's grade based on their score. The student can score anywhere from 0 to 100. The program will input the score and call a function, grade(), to calculate and output the grade. The grade is determined based on the following table:

| Score | Grade |
| --- | --- |
| 90 and above | A |
| 80 to below 90 | B |
| 70 to below 80 | C |
| 60 to below 69 | D |
| Below 60 | F |