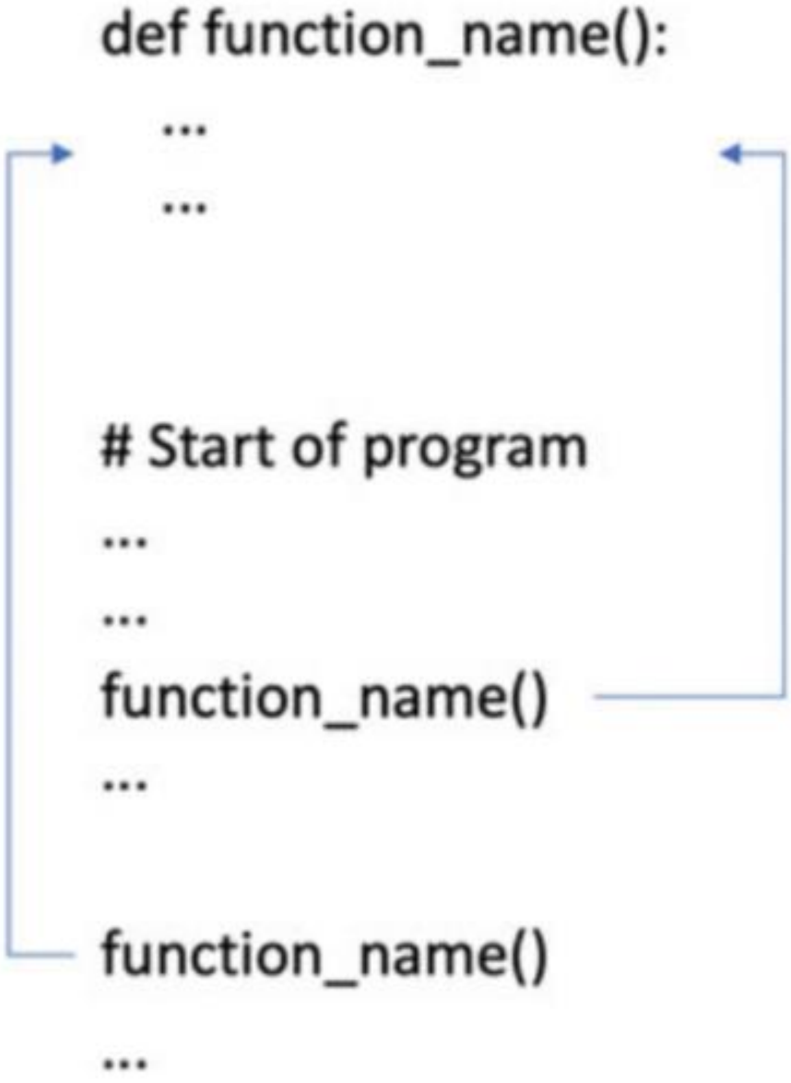
Functions in Python

- By
- Dr. Amer Almahdawi
- Data Structure – Labrotary
- Second stage, College of Science for Women, University of Baghdad

What Are Functions?

- In Python functions are groups of related statements that can be called together, that typically perform a specific task, and which may or may not take a set of parameters or return a value.
- Functions can be defined in one place and called or invoked in another. This helps to make code more modular and easier to understand.
- It also means that the same function can be called multiple times or in multiple locations.

How Functions Work



```
def function_name():
    ...
    ...

# Start of program
...
...
function_name()
...

function_name()
...
```

Types of Functions

- There are two types of functions in Python; ***built-in functions*** and ***user-defined functions***.
- Built-in functions are those provided by the language and we have seen several of these already. For example, both print() and input() are built-in functions.
- In contrast user-defined functions are those written by developers.

Defining Functions

- The basic syntax of a function is illustrated below:
- def function_name(parameter list):
- """docstring""“
- statement
- statement(s)

An Example Function

- def print_msg():
-         print('Hello World!’)
- This function is called print_msg and when called (also known as invoked) it will run the body of the function which will print out the string, for example
- print_msg()
- Will generate the output
- Hello World!

An Example Function

- def print_my_msg(msg):
-         print(msg)

- print_my_msg('Hello World’)
- print_my_msg('Good day’)
- print_my_msg('Welcome’)
- print_my_msg('Ola')

- Hello World
- Good day
- Welcome
- Ola

Returning Values from Functions

- It is very common to want to return a value from a function.
- In Python this can be done using the **return** statement.
- Whenever a return statement is encountered within a function then that function will terminate and return any values following the return keyword.
- def square(n):
-     return n * n
- # Store result from square in a variable
- result = square(4)
- print(result)

Returning Values from Functions

- # Send the result from square immediately to another function
- print(square(5))
- # Use the result returned from square in a conditional expression
- if square(3) < 15:
-     print(' Still less than 15 ‘)
- When this code is run, we get:
- 16
- 25
- Still less than 15

Returning Values from Functions

- It is also possible to return multiple values from a function.
- def swap(a, b):
-         return b, a
- a = 2
- b = 3
- x, y = swap(a, b)
- print(x, ‘ , ', y)
- Which produces
- 3 , 2

Function Parameters

- Multiple Parameter Functions
- def greeter(name, message):
-         print('Welcome', name, '-', message)

- greeter('Eloise', 'Hope you like Rugby’)
- The output is
- Welcome Eloise - Hope you like Rugby

Default Parameter Values

- def greeter(name, message = 'Live Long and Prosper’):
-         print('Welcome', name, '-', message)

- greeter('Eloise’)
- greeter('Eloise', 'Hope you like Python’)
- The output
- Welcome Eloise - Live Long and Prosper
- Welcome Eloise - Hope you like Python