Python Main function

What is the main() function in Python?

Some programming languages have a special function called `main()` which is the execution point for a program file. Python interpreter, however, runs each line serially from the top of the file and has no explicit `main()` function.

Python offers other conventions to define the execution point. One of them is using the `main()` function and the *__name__* property of a python file.

What is __name__ in Python?

The *__name__* variable is a special builtin Python variable that shows the name of the current module.

It has different values depending on where we execute the Python file. Let's look at an example.

Running Python File as a Script

Suppose we have a Python file called **helloworld.py** with the following content:

```
print(__name__)
```

If we run **helloworld.py** from the command line, then it runs as a Python script. We can run the Python program using the following command:

```
$ python helloworld.py
```

When we run the program as a script, the value of the variable *__name__* is set to `__main__`. So the output of the following program will be:

```
__main__
```

Running Python file as a Module

We can also run a Python file as a module. For this, we have to import this file into another Python program. Let's look at an example.

Suppose we have a Python file called **main.py** in the same directory as the heloworld.py file. It has the following content:

```
import helloworld
```

When we run this file, we will have the following output:

```
helloworld
```

Here, we can see that importing a module also runs all the code in the module file.

But, we can see that instead of displaying `__main__`, the program displays the name of the module i.e. `helloworld`.

It is because, in the context of running a Python file as a module, the name of the module itself is assigned to the *__name__* variable.

Using if conditional with __name__

Now that we have understood how *__name__* variable is assigned values, we can use the `if` conditional clause to run the same Python file differently in different contexts.

Let's look at an example.

Suppose we change the content of the **helloworld.py** file to the following:

```
def main():
    print("Hello World")

if __name__=="__main__":
    main()
```

Now, when we run it as a script via the command line, the output will be:

```
Hello World
```

However, when we run it as a module by importing it in the **main.py** file, no output is displayed since the `main()` function is not called.

Here, we have created a custom `main()` function in the **helloworld.py** file. It is executed only when the program is run as a standalone script and not as an imported module.

This is the standard way to explicitly define the `main()` function in Python. It is one of the most popular use cases of *__name__* variable of a Python file.

Also Read:

- Python Functions

Table of Contents