# What does the if __name__ == “__main__”: do?

Difficulty Level : Medium • Last Updated : 11 Dec, 2020

Before executing code, Python interpreter reads source file and define few special variables/global variables.
If the python interpreter is running that module (the source file) as the main program, it sets the special __name__ variable to have a value **“__main__”**. If this file is being imported from another module, __name__ will be set to the **module’s name.** Module’s name is available as value to __name__ global variable.

A module is a file containing Python definitions and statements. The file name is the module name with the suffix .py appended.

When we execute file as command to the python interpreter,

```
python script.py
```

**Python3**

```python
# Python program to execute
# main directly
print ("Always executed")

if __name__ == "__main__":
    print ("Executed when invoked directly")
else:
    print ("Executed when imported")
```

- All of the code that is at indentation level 0 [Block 1] gets executed. Functions and classes that are defined are, well, defined, but none of their code runs.
- Here, as we executed script.py directly __name__ variable will be **__main__**. So, code in this if block[Block 2] will only run if that module is the entry point to your program.

- Thus, you can test whether your script is being run directly or being imported by something else by testing __name__ variable.

- If script is getting imported by some other module at that time **__name__** will be module name.

### Why Do we need it?

For example we are developing script which is designed to be used as module:

### Python3

```python
# Python program to execute
# function directly
def my_function():
    print ("I am inside function")

# We can test function by calling it.
my_function()
```

```
I am inside function
```

Now if we want to use that module by importing we have to comment out our call. Rather than that approach best approach is to use following code:

### Python3

```python
# Python program to use
# main for function call.
if __name__ == "__main__":
    my_function()
```

```
import myscript

myscript.my_function()
```

**Advantages :**

1. Every Python module has it's __name__ defined and if this is '__main__', it implies that the module is being run standalone by the user and we can do corresponding appropriate actions.
2. If you import this script as a module in another script, the __name__ is set to the name of the script/module.
3. Python files can act as either reusable modules, or as standalone programs.
4. if __name__ == "main": is used to execute some code **only** if the file was run directly, and not imported.

This article is contributed by **Nirmi Shah**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.
Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

**Like** 49

Next

__name__ (A Special variable) in Python