The `__name__` Variable

The `__name__` variable is a built-in variable which will automatically be assigned a value by Python when a Python code is executed. It contains the name of the module that is being executed.

Main Module and the `__name__` Variable

Let us say that there is a module called `foo.py`, and within it, there is just a single line of code: `print __name__`.

```
# foo.py
print __name__

# ======
# Output
# ======
```

Now, when you execute `foo.py` from the command prompt, the string "`__main__`" will be printed out.

The `__name__` variable contains the string "`__main__`" if you execute the module directly.

Imported Module and the `__name__` Variable

The `__name__` variable will have the string "`__main__`" only if you are executing a module directly from the command prompt. However, if you are importing a module from another module, then that first module's `__name__` variable will have its module's name in string.

```
# =========================
# foo.py
# =========================

print "foo's __name__ :", __name__

# =========================
# bar.py
# =========================

import foo.py

print "bar's __name__ :", __name__
```

If you execute the `bar` module above, you will notice that it will immediately print out "`foo's __name__ : foo`" and then print out "*`bar's __name__ : __main__`*". As you can see, since you directly executed the `bar` module from the command prompt, its `__name__` variable has the string "*`__main__`*", and since the `foo` module was an imported module, its `__name__` variable has the string `foo` .

Main Takeaway

> You **must** write `if __name__ == "__main__":` in every module that you plan to directly execute on CMD.

```python
# module.py

# your import statements go here.
import math

# your global constants go here.
GLOBAL_VAR1 = 3.4123

def foo():
    # function foo's code goes here.

def bar():
    # function bar's code goes here.

if __name__ == "__main__":
    # Your main code goes here.
    # All your module-wide variables should be defined here.
    # All your function calls should be written here.
    # ... etc.
    x = 0
    y = 1
    # ... etc.
    foo()
    bar()
    # ... etc.
```