Built-In Variables in Python

By Bhimashankar Takalki

In Python, `__name__` and `__doc__` are special built-in variables used for specific purposes. Let's explore each of them along with examples:

`__name__`:

The `__name__` variable is a special built-in variable in Python that represents the name of the current module. However, when a Python script is executed directly, the value of `__name__` is set to `'__main__'`. This allows you to determine whether a script is being run as the main program or if it's being imported as a module into another script.

Example:

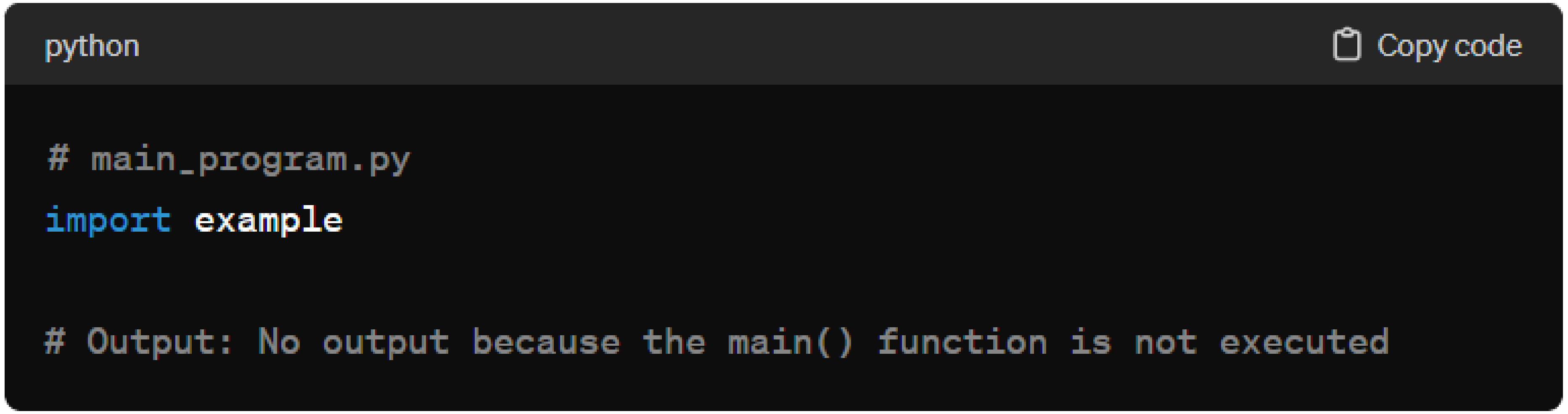
Consider the following Python script named `example.py`:

python Copy code

```python
# example.py
def main():
    print("This is the main function.")

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

When you run `example.py` directly, the `main()` function will be executed because `__name__` is set to `'__main__'`. However, if you import `example.py` as a module into another script, the `main()` function won't be executed automatically.



```python
# main_program.py
import example

# Output: No output because the main() function is not executed
```

`__doc__`:

The `__doc__` variable is a special built-in variable in Python that contains the docstring (documentation string) of a module, class, function, or method. It allows you to access and display the documentation string associated with an object.

Example:

Consider the following Python script with a docstring:

```python
def square(x):
    """Return the square of a number."""
    return x ** 2

print(square.__doc__)
```

When you run this script, it will output the docstring associated with the `square` function:

```css
Return the square of a number.
```

Built-in Variables:

Python also has several other built-in variables that provide useful information or perform special tasks. Some common built-in variables include:

- `__file__`: Contains the filename of the current module.
- `__package__`: Contains the name of the current package.
- `__all__`: Specifies the symbols to be exported when using `from module import *`.
- `__annotations__`: Contains the variable annotations of a function or method.

```
# example.py
print("__file__:", __file__)
print("__package__:", __package__)
print("__all__:", __all__)
print("__annotations__:", square.__annotations__)
```