Modules

In Python, a module is a file containing Python definitions and statements. The file name is the module name with the suffix `.py` added. Modules allow you to organize your code logically, making it more readable and maintainable. They also help in reusability by allowing you to reuse code in different parts of your program or in different programs altogether.

Here are some key points about Python modules:

### Creating a Module:

To create a module, you can simply write your Python code in a file with a `.py` extension. For example, create a file named `mymodule.py`:

```python
# mymodule.py

def greet(name):
    return f"Hello, {name}!"

def square(x):
    return x ** 2
```

### Importing a Module:

Once you've created a module, you can use the `import` keyword to bring it into another Python script. For example:

```python
# main.py

import mymodule

print(mymodule.greet("Alice"))
print(mymodule.square(5))
```

### Importing Specific Functions:

You can also import specific functions from a module:

```python
# main.py

from mymodule import greet, square

print(greet("Bob"))
print(square(3))
```

### Renaming a Module:

You can use the `as` keyword to give a module a different name:

```python
# main.py

import mymodule as mm

print(mm.greet("Charlie"))
print(mm.square(4))
```

### Executing Modules as Scripts:

You can write code in a module that is meant to be executed when the module is run as the main program:

```python
# mymodule.py

def greet(name):
    return f"Hello, {name}!"

def square(x):
    return x ** 2

if __name__ == "__main__":
    print(greet("Dave"))
    print(square(6))
```

When you run `mymodule.py` directly, the code under `if __name__ == "__main__":` will be executed.

### Standard Python Modules:

Python also comes with a rich standard library that includes many modules for various tasks. For example:

```python
import math

print(math.sqrt(25))  # Square root function from the math module
```

This is a basic overview of Python modules. They play a crucial role in organizing code and promoting code reuse in larger projects.