Python User Created Modules

User Created Python Modules

- Before using, must **import** the module with an **import** statement

User Created Modules

- An **external file** containing a set of code you want to **import** into your current program.
- Modules are created to perform a **special function**.

moduleEx.py

myModule.py

Exercise: Create a Python file

- Save As: **moduleEx.py**

Exercise: Create a Module file

- Save As: **myModule.py**

Exercise: Enter code in myModule.py

```
def greeting(name):
        print(“Hello, “ + name)
```

- This module contains a **function**.
- When the module is called, the **greeting function** inside the module will execute.

How to Use a Module in your Current Program (moduleEx.py)

```
import myModule

myModule.greeting("Your Name")
```

1. In current program, **IMPORT** the **myModule** module.
2. **Call** the greeting function inside the module.
   - **Syntax: modulename.functionname**
3. ("Your Name") argument will be passed to the **greeting function.**

Exercise: Add to myModule.py

```
def greeting(name):
        print(“Hello, “ + name)

#Part of the module, not the function
person1 = {
  "name": "John",
  "age": 36,
  "country": "Norway"
}
```

- This module contains a **function**.
- and person1 is a **Dictionary**

Accessing information in the person1 dictionary in myModule

```
import myModule

print(myModule.person1["age"])
```

- **Syntax: modulename.dictionary[key]**

Import **parts** from Module

You can choose only parts from a module using the **from** keyword

Import parts from Module

```
from myModule import person1

print (person1["age"])
```