Python Modules

What is a Module?

Consider a module to be the same as a code library.

A file containing a set of functions you want to include in your application.

Create a Module

To create a module just save the code you want in a file with the file extension `.py`:

Example

Save this code in a file named `mymodule.py`

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

Use a Module

Now we can use the module we just created, by using the `import` statement:

Example

Import the module named mymodule, and call the greeting function:

```
import mymodule

mymodule.greeting("Jonathan")
```

Output:

```
Hello, Jonathan
```

Note: When using a function from a module, use the syntax: *module_name.function_name*.

Variables in Module

The module can contain functions, as already described, but also variables of all types (arrays, dictionaries, objects etc):

Example

Save this code in the file `mymodule.py`

```
person1 = {
  "name": "John",
  "age": 36,
  "country": "Norway"
}
```

Example

Import the module named mymodule, and access the person1 dictionary:

```
import mymodule

a = mymodule.person1["age"]
print(a)
```

Output:

```
36
```

Naming a Module

You can name the module file whatever you like, but it must have the file extension `.py`

Re-naming a Module

You can create an alias when you import a module, by using the `as` keyword:

Example

Create an alias for `mymodule` called `mx`:

```
import mymodule as mx

a = mx.person1["age"]
print(a)
```

Output:

```
36
```

Built-in Modules

There are several built-in modules in Python, which you can import whenever you like.

Example

Import and use the `platform` module:

```
import platform

x = platform.system()
print(x)
```

Output:

```
Windows
```

Using the dir() Function

There is a built-in function to list all the function names (or variable names) in a module. The `dir()` function:

Example

List all the defined names belonging to the platform module:

```
import platform

x = dir(platform)
print(x)
```

Note: The dir() function can be used on *all* modules, also the ones you create yourself.

Import From Module

You can choose to import only parts from a module, by using the `from` keyword.

Example

The module named `mymodule` has one function and one dictionary:

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

person1 = {
  "name": "John",
  "age": 36,
  "country": "Norway"
}
```

Example

Import only the person1 dictionary from the module:

```
from mymodule import person1

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

Output:

```
36
```

Note: When importing using the `from` keyword, do not use the module name when referring to elements in the module. Example: `person1["age"]`, **not** ~~`mymodule.person1["age"]`~~

Import multiple From Module

You can choose to import multiple parts from a module, by using the `from` keyword.

Example

The module named `mymodule` has one function and two dictionaries:

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

person1 = {
  "name": "John",
  "age": 36,
  "country": "Norway"
}


person2 = {
  "name": "Cena",
  "age": 40,
  "country": "Ireland"
}
```

Example

Import only the person1 dictionary from the module:

```
from mymodule import person1,person2

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

Output:

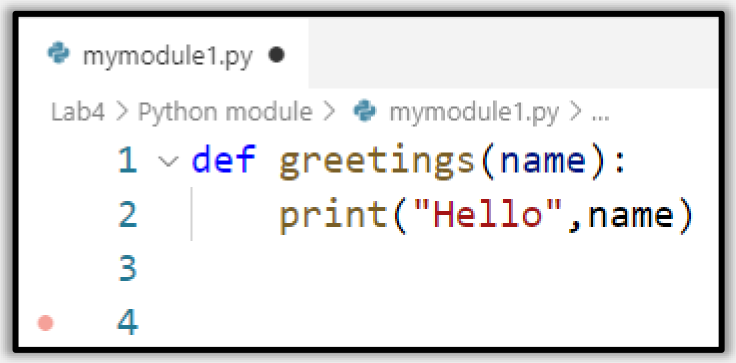
```
36

40
```

Module inside of a module

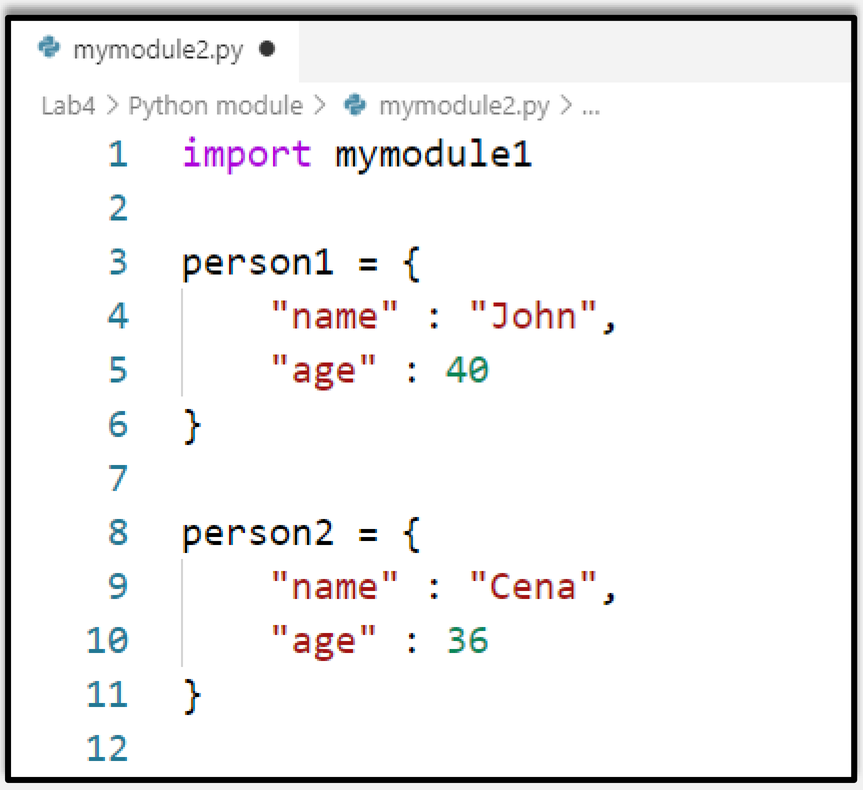
You can use imported modules inside of our modules as follows-.

Example

Suppose you have following two modules and you want to use them in test1.py –



```python
def greetings(name):
    print("Hello",name)


```



```python
import mymodule1

person1 = {
    "name" : "John",
    "age" : 40
}

person2 = {
    "name" : "Cena",
    "age" : 36
}

```

You can use them as follows in test1.py -

```python
import mymodule2

message = mymodule2.mymodule1.greetings("John Cena")

print(mymodule2.person1["age"])
print(mymodule2.person2["age"])
```

Output:

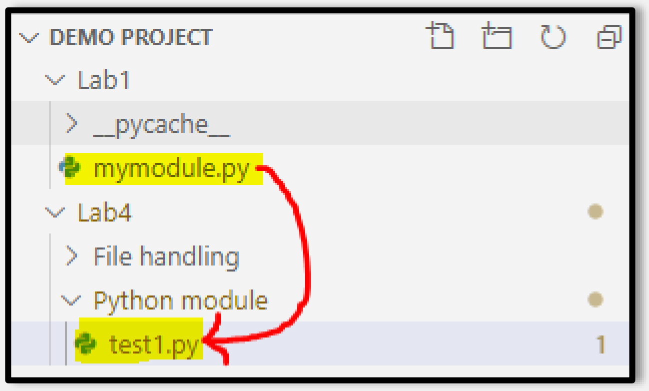
```
Hello John Cena
40
36
```

Import module from a different location

If you want to import module from a different location, you have to add the folder location in sys.path.

Example

Suppose you have a module named "myModule.py" as follows. You have to import it in test1.py –



The module named `mymodule` has one function and two dictionaries:

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

person1 = {
  "name": "John",
  "age": 36,
  "country": "Norway"
}


person2 = {
  "name": "Cena",
  "age": 40,
  "country": "Ireland"
}
```

Example

Import sys and add the path to the module folder in the sys.path.
Then import mymodule:

```
import sys
sys.path.append("../../Lab1")

from mymodule import person1,person2

print (person1["age"])
print (person2["age"])
```

Output:

```
36

40
```