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[Get your own Python Server](https://www.w3schools.com/python/python_server.asp)

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")

[Run Example »](https://www.w3schools.com/python/showpython.asp?filename=demo_module1)

**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)

[Run Example »](https://www.w3schools.com/python/showpython.asp?filename=demo_module2)

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)

[Run Example »](https://www.w3schools.com/python/showpython.asp?filename=demo_module3)

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)

[Try it Yourself »](https://www.w3schools.com/python/trypython.asp?filename=demo_module4)

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)

[Try it Yourself »](https://www.w3schools.com/python/trypython.asp?filename=demo_module5)

**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"])

[Run Example »](https://www.w3schools.com/python/showpython.asp?filename=demo_module6)

**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"]~~