

# \_\_Import \_\_ () function in Python

\_\_Import \_\_ () is an advanced function called by the import statement in Python

\_\_import \_\_ () is an advanced function called by the import statement in Python.

## The \_\_import \_\_ () syntax in Python

```
__import__(name, globals=None, locals=None, fromlist=(), level=0)
```

**The parameter of \_\_import function \_\_ ():**

1. name : the name of the module to import.
2. globals and locals : used to determine name interpretation.
3. fromlist : names of objects or sub-models should be entered from the provided module by name .
4. level : specifies whether to use absolute or relative import.

**Note :** This is an unnecessary advanced function in using regular Python programming.

This function is called by the import statement. It is used to change the semantics of the import statement (by entering the \_\_builtin\_\_ module and assigning it to \_\_builtin \_\_.\_\_ import\_\_). However, now it is replaced by simpler use of import hooks, which will be introduced in another article. So using \_\_import \_\_ () is very rare, unless you want to enter the module only knowing each parameter name at runtime.

**For example :** The way \_\_import \_\_ () works in Python

```
mathematics = __import__('math', globals(), locals(), [], 0)
print(mathematics.fabs(-2.5))
```

Run the program, the result is:

```
2.5
```

See also: Built-in Python functions

You finished reading the article "**\_\_Import \_\_ () function in Python**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.