# **Python \_\_import\_\_()**

The \_\_import\_\_() is an advanced function that is called by the import statement.

Syntax:

\_\_import\_\_(name, globals=None, locals=None, fromlist=(), level=0)

* **name** - name of the module you want to import
* **globals** and **locals** - determines how to interpert name
* **fromlist** - objects or submodules that should be imported by name
* **level** - specifies whether to use absolute or relative imports

This \_\_import\_\_() function is not necessary in everyday Python programming. It is rarely used and often discouraged.

This function can be used to changed the semantics of [import statement](https://www.programiz.com/python-programming/modules) as importstatement calls this function. However, it is always better to use import hooks.

If you want to import a module by name, use[importlib.import\_module()](https://docs.python.org/3/library/importlib.html#importlib.import_module) instead.

Ex: mathematics = \_\_import\_\_('math', globals(), locals(), [], 0)

print(mathematics.fabs(-2.5))

output: 2.5

# **Python help()**

**The help() method calls the built-in Python help system.**

The syntax of help() is:

help(object)

The help() method takes maximum of one parameter.

* **object** (optional) - you want to generate the help of the given object

The help() method is used for interactive use. It's recommenced to try it in your interpreter when you need help to write Python program and use [Python modules](https://www.programiz.com/python-programming/modules).

Ex: help(list)

Output: Help on class list in module builtins:

class list(object)

| list() -> new empty list

| list(iterable) -> new list initialized from iterable's items

|

| Methods defined here:

|

| \_\_add\_\_(self, value, /)

| Return self+value.

|

| \_\_contains\_\_(self, key, /)

| Return key in self………..

………………..

All methods will be displayed

# **Python dir()**

**The dir() method tries to return a list of valid attributes of the object.**

The syntax of dir() is:

dir([object])

The dir() takes maximum of one object.

* **object** (optional) - dir() attempts to return all attributes of this object.

The dir() tries to return a list of valid attributes of the object.

* If the object has \_\_dir\_\_() method, the method will be called and must return the list of attributes.
* If the object doesn't have \_\_dir\_\_() method, this method tries to find information from the \_\_dict\_\_ attribute (if defined), and from type object. In this case, the list returned from dir() may not be complete.

If object is not passed to the dir() method, it returns the list of names in the current local scope.

Ex: number = [1, 2, 3]

print(dir(number))

print('\nReturn Value from empty dir()')

print(dir())

Output : ['\_\_add\_\_', '\_\_class\_\_', '\_\_contains\_\_', '\_\_delattr\_\_', '\_\_delitem\_\_', '\_\_dir\_\_', '\_\_doc\_\_', '\_\_eq\_\_', '\_\_format\_\_', '\_\_ge\_\_', '\_\_getattribute\_\_', '\_\_getitem\_\_', '\_\_gt\_\_', '\_\_hash\_\_', '\_\_iadd\_\_', '\_\_imul\_\_', '\_\_init\_\_', '\_\_init\_subclass\_\_', '\_\_iter\_\_', '\_\_le\_\_', '\_\_len\_\_', '\_\_lt\_\_', '\_\_mul\_\_', '\_\_ne\_\_', '\_\_new\_\_', '\_\_reduce\_\_', '\_\_reduce\_ex\_\_', '\_\_repr\_\_', '\_\_reversed\_\_', '\_\_rmul\_\_', '\_\_setattr\_\_', '\_\_setitem\_\_', '\_\_sizeof\_\_', '\_\_str\_\_', '\_\_subclasshook\_\_', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']

Return Value from empty dir()

['\_\_annotations\_\_', '\_\_builtins\_\_', '\_\_doc\_\_', '\_\_file\_\_', '\_\_loader\_\_', '\_\_name\_\_', '\_\_package\_\_', '\_\_spec\_\_', 'number']