

# How to import a Python module given the full path?

## - GeeksforGeeks

deepanshu\_rustagi@deepanshu\_rustagi

The python module is a file consisting of Python code with a set of functions, classes, and variables definitions. The module makes the code reusable and easy to understand. The program which needs to use the module should import that particular module. In this article, we will discuss how to import a Python module given its full path.

There are various methods that can be used to import the module by using its full path:

- Using `sys.path.append()` Function
- Using `importlib` Package
- Using `SourceFileLoader` Class

Consider the following file arrangement and let's see how the above-listed methods can be used to import *gfg.py* module in *main.py*:

```
python
|--main.py
|articles
|--gfg.py
```

Below is the code for **gfg.py**:

- Python3

## Python3

```
class GFG:
```

```
    def method_in():
```

```
        print("Inside Class method")
```

```
def method_out():
```

```
    print("Inside explicit method")
```

## Using `sys.path.append()` Function

This is the easiest way to import a Python module by adding the module path to the path variable. The path variable contains the directories Python interpreter looks in for finding modules that were imported in the source files.

### Syntax :

```
sys.path.append("module_path")
```

### Example :

- Python3

## Python3

```
import sys

sys.path.append('articles')

import gfg

from gfg import GFG

GFG.method_in()

gfg.method_out()
```

### Output:

```
Inside Class method
Inside explicit method
```

## Using importlib Package

The importlib package provides the implementation of the import statement in Python source code portable to any Python interpreter. This enables users to create their custom objects which helps them to use the import process according to their needs. The importlib.util is one of the modules included in this package that can be used to import the module from the given path.

### Syntax :

```
module = importlib.util.spec_from_file_location("module_name", "module_path")
```

### Example:

- Python3

## Python3

```
import importlib.util

spec=importlib.util.spec_from_file_location("gfg","articles/gfg.py")

foo = importlib.util.module_from_spec(spec)

spec.loader.exec_module(foo)

foo.GFG.method_in()

foo.method_out()
```

### Output:

```
Inside Class method
Inside explicit method
```

## Using SourceFileLoader Class

SourceFileLoader class is an abstract base class that is used to implement source file loading with help of load\_module() function which actually imports the module.

**Syntax :**

```
module = SourceFileLoader("module_name","module_path").load_module()
```

**Example :**

- Python3

## Python3

```
from importlib.machinery import SourceFileLoader

foo = SourceFileLoader("gfg","articles/gfg.py").load_module()

foo.GFG.method_in()

foo.method_out()
```

**Output:**

```
Inside Class method
Inside explicit method
```