

# 1. Import Modules

Let's say we have my\_module.py programmed as follows :

```
# Make module in my_module.py
print('Imported my_module ...')

test = 'Test string'

def find_index(to_search, target):
    '''Find the index of a value in a sequence'''
    for i, value in enumerate(to_search):
        if value == target:
            return i
    return -1
```

If we want to apply programmed module on other script, we use import statement.

```
import my_module

courses = ['Math', 'History', Physics, 'CompSci']
index = my_module.find_index(courses, 'Math')
print(index)

# Result will be such as follows
>>> Imported my_module ...
>>> 1
```

## 1.1 Make aliased name of modules

We can make alias name of imported module using as statement.

```
import my_module as mm

courses = ['Math', 'History', Physics, 'CompSci']
index = mm.find_index(courses, 'Math')
print(index)

# Result will be such as follows
>>> Imported my_module ...
>>> 1
```

## 1.2 Import specific method from modules

We can extract specific function from module using from ~ import statement. We can apply as statement also.

If we want to import all things from imported module we can attach \* after import statement, but not recommended.

```
from my_module import find_index

courses = ['Math', 'History', Physics, 'CompSci']
index = find_index(courses, 'Math')
print(index)

# Result will be such as follows
>>> Imported my_module ...
>>> 1
```

## 1.4 Get path of module

We can get path of imported module using sys library. When loading another Python file through an import statement, Python navigates the paths in sys.path and PYTHONPATH to find the file internally. We can easily load Python files in any directory by modifying these two variables appropriately.

```
In [2]: import sys
print(sys.path)

['/home/jeongseokgyu/git_repository/Study_materials/[Personal] Data Engineering Roadmap/01 Python', '/home/jeongseokgyu/anaconda3/lib/python38.zip', '/home/jeongseokgyu/anaconda3/lib/python3.8', '/home/jeongseokgyu/anaconda3/lib/python3.8/lib-dynload', '', '/home/jeongseokgyu/anaconda3/lib/python3.8/site-packages', '/home/jeongseokgyu/anaconda3/lib/python3.8/site-packages/loket-0.2.1-py3.8.egg', '/home/jeongseokgyu/anaconda3/lib/python3.8/site-packages/IPython/extensions', '/home/jeongseokgyu/.ipython']
```

sys.path is a list of strings containing the paths to the directory. If we add a path to this list, we can import the Python file in that path into the import statement.

<https://www.bangseongbeom.com/python-double-sharp-comments.html>

# 2. The standard library

The Python's standard library is very extensive, offering a wide range of facilities as indicated by the long table of contents. The library contains built-in modules (written in C) that provides access to system functionality such as file I/O that would otherwise be inaccessible to Python programmers, as well as modules written in Python that provide standardized solutions for many problems that occur in everyday programming.

```
In [3]: # random library

import random

courses = ['History', 'Math', 'Physics']
random.choice(courses)
```

Out[3]: 'Physics'

```
In [6]: # Get current working directory using os module
import os
print(os.getcwd())

# find the directory of os module
# This will return the location of file in filesystem.
print(os.__file__)

/home/jeongseokgyu/git_repository/Study_materials/[Personal] Data Engineering Roadmap/01 Python
/home/jeongseokgyu/anaconda3/lib/python3.8/os.py
```