

오늘의 강의 목표

- Module에 대한 이해

Modules

- 연관된 function들의 집합

```
>>> import math ← math module import
>>> dir(math)      math module에 속하는 function들
['__doc__', '__loader__', '__name__', ↙
['__package__', '__spec__', 'acos', 'acosh',
'asin', 'asinh', 'atan', 'atan2', 'atanh',
'ceil', 'copysign', 'cos', 'cosh', 'degrees',
'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs',
'factorial', 'floor', 'fmod', 'frexp', 'fsum',
'gamma', 'hypot', 'isfinite', 'isinf', 'isnan',
'ldexp', 'lgamma', 'log', 'log10', 'log1p',
'log2', 'modf', 'pi', 'pow', 'radians', 'sin',
'sinh', 'sqrt', 'tan', 'tanh', 'trunc']
```

import

- 특정 module을 사용할 수 있도록 loading

```
import module1[, module2[, ... moduleN]
```

```
>>> import turtle, random, math  
>>>
```

- Module의 nickname 지정

```
import module as nickname
```

```
>>> import math as m  
>>> m.sqrt(16)  
4.0
```

Alias 혹은 nickname

dir

- 특정 모듈의 function과 variable list

```
dir(module)
```

```
>>> dir(turtle)
['__doc__', '__loader__', '__name__', '__package__',
 '__spec__', 'acos', 'acosh', 'asin', 'asinh', 'atan',
 'atan2', 'atanh', 'ceil', 'copysign', 'cos', 'cosh',
 'degrees', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs',
 'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma',
 'hypot', 'isfinite', 'isinf', 'isnan', 'ldexp', 'lgamma',
 'log', 'log10', 'log1p', 'log2', 'modf', 'pi', 'pow',
 'radians', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'trunc']
```

from ... import ...

- 특정 module에서 특정 기능만 loading

```
from module import name
```

```
>>> from math import sqrt
>>> sqrt(16)
4.0
```

Module loading과
function loading의 차이
→ namespace



```
>>> import math
>>> sqrt(16)
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    sqrt(16)
NameError: name 'sqrt' is not defined
>>> math.sqrt(16)
4.0
```

from ... import ...

```
>>> from math import *  
>>> sin(radians(90))  
1.0  
>>> cos(radians(180))  
-1.0
```

← math module에 속하는
모든 function import
주의: 권장하지 않음

```
>>> pi = "Apple Pie"  
>>> from math import *  
>>> pi  
3.141592653589793
```

math module의 pi 변수가
내 pi 변수를 overwrite

Python Modules

- Python은 수많은 module들을 기본적으로 제공

<https://docs.python.org/3/py-modindex.html>

- 지금까지 사용해본 module list
 - sys
 - turtle
 - msvcrt import 안해도 전부 다 미리 loading 해주면 안되나?
→ 메모리 소모량을 고려해서 필요한 것만 loading
 - math
 - statistics
 - random

Writing Modules

- 모듈명.py를 C:\Python34 밑에 생성

```
def print_hello():  
    print("Hello")
```

hello.py

```
def print_bye():  
    print("Bye")
```

```
def print_hi():  
    print("Hi")
```

```
print("Thank you for loading hello")
```


Using Modules

- import 하여 일반 module 처럼 사용

```
>>> import hello
Thank you for loading hello
>>> dir(hello)
['__builtins__', '__cached__', '__doc__', '__file__',
 '__loader__', '__name__', '__package__', '__spec__',
 'print_bye', 'print_hello', 'print_hi']
>>> hello.print_hello()
Hello
>>> hello.print_bye()
Bye
>>> hello.print_hi()
Hi
>>> hello.__file__ # 파일명
'C:\\Python34\\hello.py'
>>> hello.__name__ # 모듈명
'hello'
```

Using Modules


- from ... import ... 사용

idle 재시작 필요

```
>>> from hello import print_hello, print_bye, print_hi
Thank you for loading hello
>>> print_hello()
Hello
>>> print_bye()
Bye
>>> print_hi()
Hi
```

Locating Modules

- import 할때 해당 py 파일을 어디서 찾나?
 - 현재 디렉토리
 - PYTHONPATH 환경변수
 - 디폴트 path

```
>>> import sys Module search path  
>>> sys.path   
['', 'C:\\Python34\\Lib\\idlelib',  
'C:\\Windows\\system32\\python34.zip',  
'C:\\Python34\\DLLs', 'C:\\Python34\\lib',  
'C:\\Python34', 'C:\\Python34\\lib\\site-packages']
```

Locating Modules

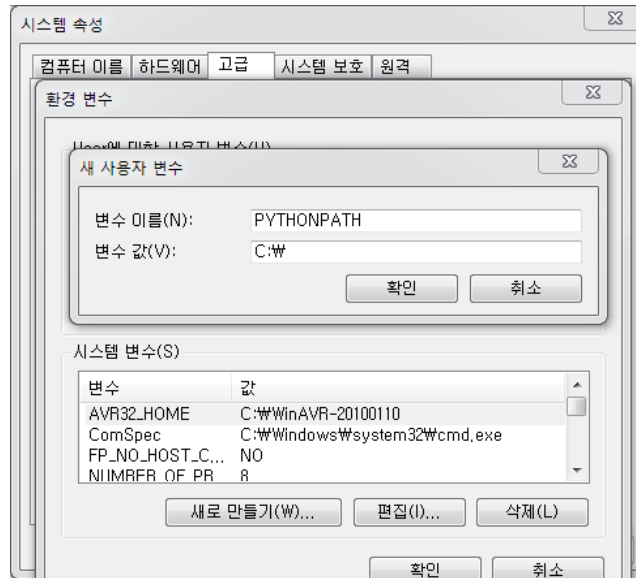
- C:\Python34\hello.py 를 C:\ 로 옮기면

idle 재시작 필요

```
>>> import hello
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    import hello
ImportError: No module named 'hello'
>>>
```

Locating Modules

- 환경 변수 PYTHONPATH에 C:\를 추가



idle 재시작 필요

```
>>> import sys
>>> sys.path
['', 'C:\\Python34\\Lib\\idlelib', 'C:\\',
'C:\\Windows\\system32\\python34.zip',
'C:\\Python34\\DLLs', 'C:\\Python34\\lib',
'C:\\Python34', 'C:\\Python34\\lib\\site-packages']
>>> import hello
>>>
```

Module vs Script

```
def print_hello():  
    print("Hello")
```

```
def print_bye():  
    print("Bye")
```

```
def print_hi():  
    print("Hi")
```

```
if __name__ == "__main__":
```

```
    print_hello()
```

```
    print_bye()
```

```
    print_hi()
```

```
else:
```

```
    print("Thank you for importing hello")
```

} hello.py 를 스크립트로
실행한 경우 실행

} hello.py 를
import한
경우 실행

Advanced Topics

- Package

Questions

