

모듈

백석대학교 강윤희



모듈의 필요성

- 비슷한 함수와 클래스를 작성한다면 코드가 길어지고 중복되는 부분이 생김
 - 공통되는 부분을 빼내서 모듈(module)과 패키지(package)로 구성함
 - 코드를 다시 만들지 않고 모듈과 패키지만 가져와서 사용하면 편리함
 - 모듈은 간단한 기능을 담을 때 사용하며, 패키지는 코드가 많고 복잡할 때 사용
- ❖ 패키지(package)는 여러 모듈을 묶은 것



모듈

- 모듈(Module)은 파이썬 코드를 논리적으로 묶어서 관리하고 사용할 수 있도록 하는 것
- 보통 하나의 파이썬.py 파일이 하나의 모듈이 됨
- import 문은 하나 혹은 복수의 모듈을 불러들일 수 있음
- 표준 라이브러리 중 수학과 관련된 함수들을 모아 놓은 "math" 모듈을 import 하여 그 모듈 안에 있는 factorial() 함수를 사용

```
import 모듈1[, 모듈2[,... 모듈N]
```

```
import math
```

```
n = math.factorial(5)
```



모듈 작성

- 변수, 함수, 클래스 등을 모아 놓은 스크립트 파일(파이썬 파일)
- 다른 파이썬 프로그램에서 불러와 사용할 수 있는 파이썬 파일
- import 사용 이미 만들어 놓은 모듈을 사용할 수 있음

import 모듈이름

```
# mod.py
def add(a, b):
    return a + b
```

```
def sub(a, b):
    return a-b
```

```
# test_mod.py
import mod

print(mod.add(3, 4))
```



모듈 사용하기

- from import로 변수와 함수를 가져온 뒤 모듈 이름을 붙이지 않고 사용(모듈함수 활용)
 - mod.add 대신 add 사용

from 모듈이름 import 모듈함수

```
# mod.py
def add(a, b):
    return a + b
```

```
def sub(a, b):
    return a-b
```

```
# test_mod.py
from mod import add
```

```
print(add(3, 4))
```



(실습)모듈 사용하기

```
$ cat mymodule.py
person1 = {
    "name": "John",
    "age": 36,
    "country": "Norway"
}
```

```
def greeting(name):
    print("Hello, " + name)
```

```
$ cat test_mymodule.py
import mymodule

a = mymodule.person1["age"]
print(a)
mymodule.greeting("안녕하세요")
```



(실습)모듈 사용하기

- 작성 모듈 테스트 하기 (test_mod.py를 작성함)

```
# mod.py
def add(a, b):
    return a + b
```

```
def sub(a, b):
    return a-b
```

```
print(add(1, 4))
print(sub(4, 2))
```

두개의 값 5와 2를 입력 받아 add를 호출하여 값을 출력함

```
$ python test_mod.py
```

5

2

7

- 테스트 문장 수행을 제외하기 위한 방법 활용
if __name__ == "__main__":



(실습)모듈 사용하기

- `__name__` 변수는 파이썬이 내부적으로 사용하는 특별한 변수
 - 모듈 단독 수행시 `__main__` 값이 지정
 - 아래에서 `add()`와 `sub()`가 실행됨
- `import` 시에는 모듈 파일명이 값으로 지정되어 테스트 문장 실행을 막기

```
# mod.py
def add(a, b):
    return a + b
```

```
def sub(a, b):
    return a-b
```

```
if __name__ == "__main__":
    print(add(1, 4))
    print(sub(4, 2))
```

두수의 값 5와 2를 더하기 위해 add를 호출함
\$ python test_mod.py
7



*모듈 사용하기

- 모듈에서 정의된 클래스 사용하기
 - 모듈.클래스명() 로 객체 생성

```
# mod.py
```

```
PI = 3.141592
```

```
class Circle:  
    def area(self, r):  
        return PI * (r ** 2)
```

```
# test_mod.py  
import mod
```

```
print (mod.PI)
```

```
a = mod.Circle()  
print (a.area(2))
```

