파이썬 프로그래밍

모듈의 활용과 패키지

한국기술교육대학교
온라인평생교육원

▣ 모듈의 실행과 테스트 코드

•__name__
- 현재의 모듈이 최상위 모듈로서 수행되는지, 아니면 다른 모듈에 의해 import 되어 수행되는지를 구별하기 위해 주로 활용

•prname.py를 직접 수행할 때의 출력 내용: __main__
- >>> ipython prname.py◦__main__
- prname.py가 최상위 모듈로서 수행됨을 의미

```
#FILE : prname.py
print __name__
```

```
__main__
```

▪__name__ 이 값이 디폴트로 존재
▪__main__ 값이 __name__에 할당되어 있음
▪모듈을 만드는 것 → .py를 만들어서 바로 수행하는 것
▪__name__은 바로 수행해서 사용하고 있으므로 최상위 모듈

•prname.py가 모듈로서 다른 이름 공간으로 import 되어질 때의 출력 내용: prname

```
import prname
print prname.__name__
```

```
prname
prname
```

▪test가 최상위 모듈 → 이 모듈 내 prname의 네임은 prname
▪prname 안 print name = test의 prname name
▪최상위 모듈 → 바로 prname을 정의해서 수행하는 것
▪다른 모듈에 의해 import 되어짐 → name 값은 모듈 이름 그대로 출력

▣ 모듈의 실행과 테스트 코드

```
import string
print string.__name__

import re
print re.__name__

import mimetools
print mimetools.__name__

import os
print os.__name__
```

```
string
re
mimetools
os
```

- string, re, mimetools, os → 표준 모듈
- __name__을 쓰면 전부 다 모듈의 이름과 동일 내용 출력

▣ 모듈의 실행과 테스트 코드

•아래 코드는 최상위 모듈로서 수행될 때에만 test() 함수 호출이 일어난다.
•보통 파이썬 모듈을 개발할 때에는 마지막 부분에 if __name__ == "__main__": 과 같은 코드를 추가하여 테스트 코드를 삽입한다.

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

def f():
    print "Python is becoming popular."

if __name__ == "__main__":
    print add(1, 10)
    f()
```

```
11
Python is becoming popular.
```

▪새로운 .py = 새로운 모듈
▪모듈이 최상위 모듈로 활용된다면 name에 main값이 들어가 있음
▪다른 모듈에서 import 하는 순간 name 값은 모듈 이름 그대로 출력
▪모듈 개발 시 if절 삽입 → 현재 개발 중인 모듈의 test 가능

▣ 모듈의 실행과 테스트 코드

•정의된 모듈이 다른 모듈로 import되어질 때에는 __name__은 모듈 이름 자체이므로 위 코드에서 if 문이 수행되지 않는다.
- 즉, test() 함수 호출이 곧바로 되지 않는다.

```
import module_test
```

▪모듈 개발 시 if절이 많이 활용됨