

---

# 모듈과 패키지

# 모듈

---

## ❖ 모듈

- 변수와 함수를 모두 한 파일에 정의하면 관리가 힘들어짐
- 비슷한 성격의 변수, 함수들을 파일 별로 나눠 정의
- 이렇게 정의한 파일을 모듈이라고 함
- 파일명이 모듈명이 됨

# 모듈

---

## ❖ 모듈의 정의 (util.py)

```
INCH = 2.54
```

```
def calcsun(n):  
    sum = 0  
    for num in range(n+1):  
        sum += num  
    return sum
```

## ❖ 모듈의 사용

```
import util  
  
print("1inch = ", util.INCH)  
print("~10 = ", util.calcsun(10))
```

```
1inch = 2.54  
~10 = 55
```

# 모듈

---

## ❖ 모듈의 사용

- `from 모듈명 import *`
- 지정한 모듈의 모든 요소를 임포트함
- 바로 사용 가능하지만 이름 충돌 발생 가능성이 높아짐

```
from util import *  
  
print("1inch = ", INCH)  
print("~10 = ", calcsun(10))
```

# 모듈

---

## ❖ 모듈 테스트

- `__name__` 변수에 모듈명이 저장됨
- 단독으로 실행된 경우(모듈로 사용된 것이 아님)
  - `"__main__"` 으로 지정됨
- 모듈로 사용된 경우(import에 의해 실행된 경우)
  - 파일명이 지정됨
- 모듈 개발시 테스트 할 수 있도록 지원

# 모듈 정의

```
if __name__ == "__main__":  
    # 모듈 테스트 코드
```

# 모듈

---

## ❖ 모듈 테스트

```
INCH = 2.54
```

```
def calcsun(n):  
    sum = 0  
    for num in range(n+1):  
        sum += num  
    return sum  
  
if __name__ == "__main__":  
    print("1inch = ", INCH)  
    print("~10 = ", calcsun(10))
```

# 모듈

---

## ❖ 모듈 경로

- sys 모듈의 path 리스트를 먼저 조사
  - 현재 워킹 디렉토리가 제일 먼저 조사됨
- 그 다음 PYTHONPATH 환경 변수의 디렉토리 검색

```
import sys

for path in sys.path:
    print(path)
```

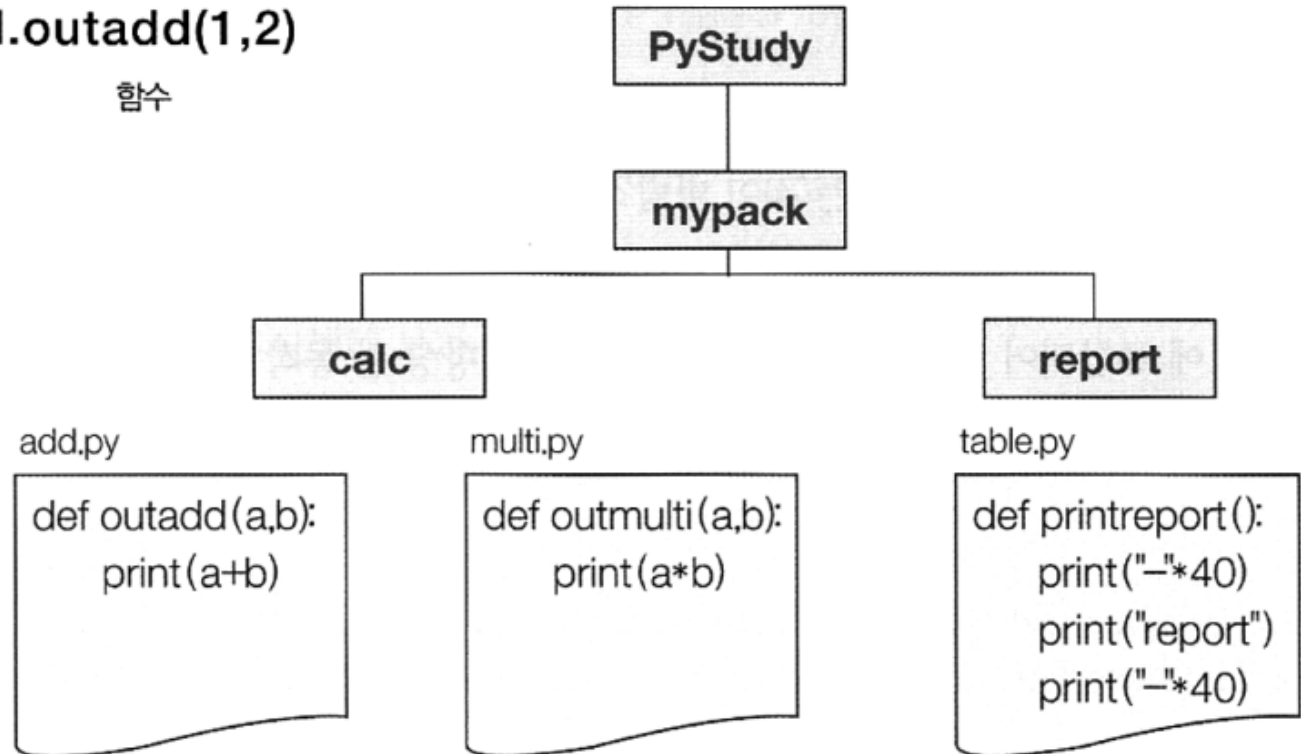
# 패키지

## ❖ 패키지

- 모듈들을 모아 놓은 디렉토리
- 반드시 `__init__.py`가 존재해야 함
  - 일반적으로 내용은 없음

`mypack.calc.add.outadd(1,2)`

패키지    디렉토리    모듈    함수





# 패키지

---

## ❖ `__init__.py`

- `from 패키지 import *`
  - \*에 의해 임포트될 모듈 목록을 `__all__` 리스트로 지정
  - 지정하지 않으면 모든 모듈이 임포트됨

```
from mypack.calc import *
```

```
add.outadd(1,2)  
multi.outmulti(1,2)
```

# 패키지

---

## ❖ \_\_init\_\_.py

```
__all__ = ["add", "multi"]  
print("add module imported")
```

# 써드 파티 모듈

---

## ❖ 모듈의 내부

```
import sys

print(sys.builtin_module_names)
```

```
('_abc',
 '_ast',
 '_codecs',
 '_collections',
 '_functools',
 '_imp',
 '_io',
 '_locale',
 '_operator',
 '_signal',
  :
)
```

# 써드 파티 모듈

---

## ❖ dir() 함수

- 모듈내 함수 목록 출력

```
import math
```

```
dir(math)
```

```
['__doc__',  
 '__file__',  
 '__loader__',  
 '__name__',  
 '__package__',  
 '__spec__',  
 'acos',  
 'acosh',  
 'asin',  
 :  
 ]
```

# 써드 파티 모듈

---

## ❖ `help()` 함수

- 함수의 docstring 출력

```
help(math.hypot)
```

Help on built-in function hypot in module math:

```
hypot(x, y, /)
```

Return the Euclidean distance, `sqrt(x*x + y*y)`.

# 써드 파티 모듈

---

## ❖ 외부 모듈 관리 pip

- `install` : 패키지를 설치
- `uninstall` : 설치한 패키지를 삭제한다.
- `freeze` : 설치한 패키지의 목록을 보여준다.
- `show` : 패키지의 정보를 보여준다.
- `search` : pyPI에서 패키지를 검색한다.

```
$ pip install beautifulsoup4
```

# 써드 파티 모듈

---

## ❖ BeautifulSoup을 이용한 날씨정보 얻기

```
from urllib import request
import bs4

target = request.urlopen("http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp?stnId=108")
soup = bs4.BeautifulSoup(target)

for city in soup.select("location"):
    name = city.select_one("city").string
    wf = city.select_one("wf").string
    tmn = city.select_one("tmn").string
    tmx = city.select_one("tmx").string
    print("%s : %s(%s ~ %s)"%(name, wf, tmn, tmx))
```

서울 : 구름많음(-1 ~ 6)

인천 : 구름많음(-1 ~ 5)

: