

# Python을 활용한 데이터 분석 강의

## 모듈과 패키지

# 모듈과 패키지

- 모듈

- 서로 연관된 함수들을 담고 있는 파이썬 파일(.py)

- 패키지

- 관련 모듈들을 체계적으로 통합 해놓은 묶음(폴더)

```
import 패키지이름 [as 식별자]  
import 패키지이름.모듈이름  
from 패키지이름 import 모듈이름
```

```
import 모듈 이름 [as 식별자]  
import 모듈이름.함수이름  
from 모듈이름 import 함수이름
```

# 모듈과 패키지

- 모듈 불러오기

```
import math
```

```
x = math.exp(10)  
print(x)
```

```
22026.465794806718
```

```
import math as m
```

```
x = m.exp(10)  
print(x)
```

```
22026.465794806718
```

# 표준 라이브러리

- 표준 라이브러리 공식 문서

<http://docs.python.org/3.7/library/index.html>

- datetime 모듈

```
from datetime import date
```

```
date.today()
```

```
datetime.date(2019, 3, 13)
```

# 표준 라이브러리

- date형 데이터를 문자열로 변환

```
from datetime import date
```

```
today = date.today()  
today.strftime('%Y%m%d')
```

```
'20190313'
```

```
today.strftime('%y/%m/%d')
```

```
'19/03/13'
```

```
today.strftime('%Y년%m월%d일')
```

```
'2019년03월13일'
```

```
today.strftime('%Y %B %d %a')
```

```
'2019 March 13 Wed'
```

기호	표시되는 형식
%Y	연도 네 자릿수
%y	연도 두 자릿수
%m	달 두 자릿수
%B	영어로 달을 표시
%b	영어로 달을 단축하여 표시
%A	영어로 요일을 표시
%a	영어로 요일을 단축하여 표시

# 표준 라이브러리

- 현재 날짜와 시간을 취득

```
datetime.now()
```

```
datetime.datetime(2019, 3, 13, 16, 18, 49, 216166)
```

```
from datetime import datetime as dt
```

```
now = dt.now()  
now.strftime('%Y-%m-%d %H:%M:%S')
```

```
'2019-03-13 16:19:39'
```

기호	표시되는 형식
%H	24시간 표기로 시간을 표시
%I	12시간 표기로 시간을 표시
%p	AM/PM 표시
%M	분을 두 자리수로 표시
%S	초를 두 자리수로 표시
%f	마이크로초를 6자리수로 표시

# 표준 라이브러리

- 일주일 후의 날짜 취득

```
from datetime import date, timedelta
```

```
today = date.today()  
today
```

```
datetime.date(2019, 3, 13)
```

```
one_week = timedelta(days = 7)  
print(today + one_week)  
print(today - one_week)
```

```
2019-03-20
```

```
2019-03-06
```

# 표준 라이브러리

- Collections 모듈의 Counter()

```
from collections import Counter
```

```
L = [1,1,2,3,3,3,3,4,4,5,6,7,7,8,9,9,9,9,9,9]  
counts = Counter(L)  
counts
```

```
Counter({1: 2, 2: 1, 3: 4, 4: 2, 5: 1, 6: 1, 7: 2, 8: 1, 9: 6})
```

```
counts.most_common()
```

```
[(9, 6), (3, 4), (1, 2), (4, 2), (7, 2), (2, 1), (5, 1), (6, 1), (8, 1)]
```

```
counts.most_common(3)
```

```
[(9, 6), (3, 4), (1, 2)]
```



# 표준 라이브러리

- itertools 모듈

```
import itertools
```

```
colors = ['green', 'blue', 'red']  
instruments = ['drum', 'guitar']
```

```
# 항목들의 곱집합
```

```
for e in itertools.product(colors, instruments):  
    print(e)  
print('')
```

```
# 항목들의 조합
```

```
for e in itertools.combinations('xyz', 2):  
    print(e)  
print('')
```

```
# 항목들의 순열
```

```
for e in itertools.permutations('xyz', 2):  
    print(e)
```

```
('green', 'drum')  
( 'green', 'guitar')  
( 'blue', 'drum')  
( 'blue', 'guitar')  
( 'red', 'drum')  
( 'red', 'guitar')
```

```
('x', 'y')  
( 'x', 'z')  
( 'y', 'z')
```

```
('x', 'y')  
( 'x', 'z')  
( 'y', 'x')  
( 'y', 'z')  
( 'z', 'x')  
( 'z', 'y')
```

# Practice 5-1 🧐

- collections.Counter 활용
  - collections.Counter를 사용해서, 아래 리스트에 속한 객체 중 3회 이상 중복된 값을 가진 객체를 출력
    - L = [1, 1, 7, 7, 7, 4, 4, 4, 2, 1, 5, 5, 9, 11, 3, 'a', 'x', 9, 8, 'b', 'b', 'a', 'b']

1  
7  
4  
b

# Practice 5-2 🧐

- 자물쇠의 비밀번호 조합
  - 두 개의 문자와 두 개의 숫자로 이루어져 있는 비밀번호를 가지고 있는 자물쇠
    - ex) AP23
  - 아래 조건을 만족하는 모든 비밀번호 조합의 개수 구하기
    - 첫번째 문자는 모음 대문자 중 하나
    - 두번째 문자는 'a'부터 'e'까지 문자 중 하나 (대소문자 모두 가능)
    - 세번째 숫자는 5 미만의 소수(prime number)
    - 네번째 숫자는 6 이하 홀수

300