

# 데이터처리프로그래밍

## list comprehension



강원대학교 교육혁신원 송혜정

<hjsong@kangwon.ac.kr>



# list comprehension

## ✓ 학습목표

- 제어문을 활용한 자료구조 생성을 이해하고 활용한다.

## ✓ 학습내용

- list comprehension
  - 반복문을 사용한 리스트 생성
  - 조건문을 사용한 필터링
- dictionary comprehension



# 강의에 앞서서..

- 본 강의자료는 아래의 자료들을 참고하여 만들어 졌음을 알립니다
  1. 데이터과학을 위한 파이썬 프로그래밍, 최성철, 한빛아카데미,2019
  2. Python (<https://docs.python.org>)
  3. 실용 파이썬 프로그래밍(Practical Python Programming)  
(<https://wikidocs.net/84426>)
  4. <https://www.w3schools.com/>

# list comprehension

- 시퀀스의 각 원소에 연산을 적용하여 새로운 리스트를 생성
- 리스트 안에 반복문, 조건문을 사용하여 리스트를 생성
- 구문  
[ <표현식> for <변수명> in <시퀀스> if <조건> ]
- 의미

```
result = []  
for 변수명 in 시퀀스:  
    if 조건:  
        result.append(표현식)
```

# list comprehension

- 리스트 생성 예

```
#1~4 연속값으로 리스트를 생성하는 방법
#(1)#4개의 수치 나열하여 리스트로 정의
l1 = [1, 2, 3, 4]
print(l1)
#(2)빈 리스트에 반복문으로 생성된 데이터를 추가
l2 = []
for i in range(1,5) :
    l2.append(i)
print(l2)
#(3)list comprehension 사용
#리스트 안에서 반복문으로 값을 생성
l = [i for i in range(1,5)]
print(l)
```

```
[1, 2, 3, 4]
[1, 2, 3, 4]
[1, 2, 3, 4]
```

# list comprehension

- 리스트 생성 예

```
#list comprehension 사용 예
l = [2*i for i in range(1,5)]      #반복문으로 생성된 값을 연산한 결과로 리스트 생성
print(l)

nums = [1,2,3,5]
squares = [ x ** 2 for x in nums ]
print(squares)

colors = ['red', 'blue', 'green']
ucolors = [c.upper() for c in colors]
print(ucolors)
```

```
[2, 4, 6, 8]
[1, 4, 9, 25]
['RED', 'BLUE', 'GREEN']
```

# list comprehension

- 조건에 따른 리스트 생성 예

```
#list comprehension 사용 예
#반복문으로 생성된 값을 조건으로 필터링한 결과로 리스트 생성

#'a'글자가 들어간 과일명으로 리스트 생성
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]
newlist = []

#반복문, 조건문 사용
for x in fruits:
    if "a" in x:
        newlist.append(x)
print(newlist)

#list comprehension 사용
newlist = [x for x in fruits if "a" in x]
print(newlist)
```

```
['apple', 'banana', 'mango']
['apple', 'banana', 'mango']
```

# list comprehension

- 조건에 따른 리스트 생성 예

*#(1) 1에서 100사이의 10의 배수만 리스트로 생성*

```
list10 = [i for i in range(1, 101) if i % 10 == 0]
print(list10)
```

*#(2) 1~5 사이의 짝수만 2배하여 리스트 생성*

```
elist = [2*i for i in range(1,6) if i % 2 == 0]
print(elist)
```

*#(3) 1~10 사이의 3의 배수만 제외시켜 3배한 값을 리스트 생성*

```
list3 = [3*i for i in range(1,11) if i % 3 != 0]
print(list3)
```

*#(4) fruits에서 "banna가 아니면 그대로, banna이면 orange로 리스트 생성*

```
flist = [x if x != "banana" else "orange" for x in fruits]
print(flist)
```

[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

[4, 8]

[3, 6, 12, 15, 21, 24, 30]

['apple', 'orange', 'cherry', 'kiwi', 'mango']



# dictionary comprehension

- 조건에 따른 딕셔너리 생성 예

```
#dictionary comprehension
#두 리스트로 딕셔너리 생성
t = ['year', 'month', 'day', 'season']
d = [2021, 5, 12, 'spring']
date1 = dict(zip(t,d))
print(date1)
#두 리스트로 딕셔너리 생성시 조건에 따라 처리 (season 제외)
date2 = {k:v for k,v in zip(t,d) if k != 'season'}
print(date2)
```

```
{'year': 2021, 'month': 5, 'day': 12, 'season': 'spring'}
{'year': 2021, 'month': 5, 'day': 12}
```

# dictionary comprehension

- 과일 수량에 따른 딕셔너리 생성 예

```
#fruits 딕셔너리에서 수량이 3개이상인 과일만 이름을 첫글자만 대문자로 변경한 딕셔너리 생성
f = [("apple", 3), ("banana", 5), ("cherry", 4), ("kiwi", 2), ("mango", 1)]
fruits = dict(f)
ufruits = {k.title():v for k, v in fruits.items() if v >= 3}
print(fruits)
print(ufruits)
```

```
{'apple': 3, 'banana': 5, 'cherry': 4, 'kiwi': 2, 'mango': 1}
{'Apple': 3, 'Banana': 5, 'Cherry': 4}
```