

21.(심화)컴프리헨션

2018.12

일병 김재형

컴프리헨션(comprehension)

리스트 컴프리헨션(지능형 리스트, 리스트 내포)

- 새로운 리스트를 만들 때, 보기 좋게 만드는 방법
- [표현식 for 표현식변수 in 순환 가능 객체]
- 예시

```
>>> # 10개의 3의 배수 요소를 가진 리스트 만들기
...
>>> result = [num * 3 for num in range(10)]
>>> result
[0, 3, 6, 9, 12, 15, 18, 21, 24, 27]
```

컴프리헨션(comprehension)

리스트 컴프리헨션(지능형 리스트, 리스트 내포)

- 조건을 추가하기
- [표현식 for 표현식변수 in 순환 가능 객체 if 조건]
- 조건에 맞는 시퀀스 자료형 내의 요소가 표현식 변수로 넘어간다.

```
>>> # 짝수에만 3을 곱해 리스트 만들기
...
>>> result = [num * 3 for num in range(10) if num % 2 == 0]
>>> result
[0, 6, 12, 18, 24]
```

컴프리헨션(comprehension)

리스트 컴프리헨션(지능형 리스트, 리스트 내포)

- 표현식변수를 추가하기
- [표현식 for 표현식변수1 in 순환 가능 객체
for 표현식변수2 in 순환 가능 객체]

```
>>> colors = ['black', 'blue']
>>> sizes = [80, 82, 84]
>>> pants = [(color, size) for color in colors for size in sizes]
>>> pants
[('black', 80), ('black', 82), ('black', 84), ('blue', 80), ('blue', 82), ('blue', 84)]
```

- for color in colors:
for size in sizes:

컴프리헨션(comprehension)

리스트 컴프리헨션(지능형 리스트, 리스트 내포)

—[표현식 for 표현식변수1 in 순환 가능 객체
for 표현식변수2 in 순환 가능 객체]

—데카르트 곱

		S		
		80	82	84
R	'black'	('black', 80)	('black', 82)	('black', 84)
	'blue'	('blue', 80)	('blue', 82)	('blue', 84)
R X S				

컴프리헨션(comprehension)

리스트 컴프리헨션(지능형 리스트, 리스트 내포)

- [표현식 for 표현식변수1 in 순환 가능 객체
for 표현식변수2 in 순환 가능 객체]
- 다음도 가능

```
>>> matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
>>> flat = [x for row in matrix for x in row]
>>> flat
[1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> flat = [x for x in row for row in matrix]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'row' is not defined
```

컴프리헨션(comprehension)

딕셔너리 컴프리헨션(지능형 딕셔너리)

- {키_표현식: 값_표현식 for 표현식 변수 in 순환 가능 객체}
- 예시

```
>>> [("key1", "value1"), ("key2", "value2"), ("key3", "value3")]  
[('key1', 'value1'), ('key2', 'value2'), ('key3', 'value3')]  
>>> dict_list = [("key1", "value1"), ("key2", "value2"), ("key3", "value3")]  
>>> test_dict = {key: value for key, value in dict_list}  
>>> test_dict  
{'key1': 'value1', 'key2': 'value2', 'key3': 'value3'}
```

- 리스트와 같게 for의 중첩과 조건도 넣을 수 있다.

컴프리헨션(comprehension)

집합 컴프리헨션(지능형 집합)

—{표현식 for 표현식변수 in 순환 가능 객체}

```
>>> numbers = [1, 2, 1, 2, 3, 4]
>>> set_numbers = { num * 2 for num in numbers }
>>> set_numbers
{8, 2, 4, 6}
```


컴프리헨션(comprehension)

컴프리헨션

- 다중 if와 다중 for도 가능하다.
- 단, 매우 복잡해지기 때문에 일반적으로
 - 조건 두 개
 - 루프 두 개
 - 조건 한 개와 루프 한 개 정도만 쓰는 것을 추천한다.

```
>>> # 행렬의 행의 합이 10이상이고, 3으로 나뉘지는 셀을 구한다.
...
>>> matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
>>> filter = [x for row in matrix if sum(row) >= 10
...           for x in row if x % 3 == 0]
>>> filter
[6, 9]
```

컴프리헨션(comprehension)

튜플의 컴프리헨션?

- 제너레이터가 생성된다.
- (심화) 제너레이터에서 설명한다.