

베이직 파이썬 (6)

Today's point!



- ✓ 리스트 컴프리헨션
- ✓ lambda

리스트 컴프리헨션 (Comprehension)

1

리스트 컴프리헨션

Comprehension(컴프리헨션)

set, list, dictionary 등 파이썬 컬렉션타입(여러 개의 변수를 담는 자료형)에 활용하는 문법이다.

- 반복문을 한 줄로 표현한다.
- 조건문을 한 줄로 표현한다.
- 한 줄로 표현한 반복문과 조건문을 set, list, dictionary 형태로 만들어준다.
- 주로 list 형식에 많이 쓰여서, 이번 장의 주제는 리스트 컴프리헨션!

0에서 10까지의 반복문을 number라는 이름의 리스트로 만들자.

```
number = [ ]  
for i in range(10+1):  
    number.append(i)
```



3줄이나!

리스트 컴프리헨션 - 반복문

0에서 10까지의 반복문을 number라는 이름의 리스트로 만들자.

```
number = [ ]  
for i in range(10+1):  
    number.append(i)
```

A red starburst graphic with a white outline, containing the text '1줄로!' in white Korean characters.

1줄로!

```
number = [ x for x in range(10+1) ]
```

리스트 컴프리헨션 - 반복문

0에서 10까지의 반복문을 number라는 이름의 리스트로 만들자.

```
number = []  
for i in range(10+1):  
    number.append(i)
```

```
number = [ x for x in range(10+1) ]
```

리스트 컴프리헨션 - 반복문

0에서 10까지의 반복문을 number라는 이름의 리스트로 만들자.

```
number = [ ]  
for i in range(10+1):  
    number.append(i)
```

```
number = [ x for x in range(10+1) ]
```

리스트 컴프리헨션 - 반복문

0에서 10까지의 반복문을 number라는 이름의 리스트로 만들자.

```
number = []  
for i in range(10+1):  
    number.append(i)
```

```
number = [ x for x in range(10+1) ]
```




리스트 컴프리헨션 - 조건문

0에서 10까지의 리스트 중에서 홀수만 모아서 새 리스트로 만들자.

```
odds = [ ]  
for i in number :  
    if i%2 == 1 :  
        odds.append(i)
```

어디로
가야 하나요

```
odds = [ x for x in number ]
```

리스트 컴프리헨션 - 조건문

0에서 10까지의 리스트 중에서 홀수만 모아서 새 리스트로 만들자.

```
odds = [ ]  
for i in number :  
    if i%2 == 1 :  
        odds.append(i)
```

어디로
가야 하나요

```
odds = [ x for x in number if x%2 == 1 ]
```



리스트 컴프리헨션 - 조건문

0에서 10까지의 리스트 중에서 홀수만 모아서 새 리스트로 만들자.

2개라면?

+

0에서 10까지의 홀수 중, 5보다 작은 것만 리스트로 만들자.

리스트 컴프리헨션 - 조건문

0에서 10까지의 리스트 중에서 홀수만 모아서 새 리스트로 만들자.

2개라면?

+

0에서 10까지의 홀수 중, 5보다 작은 것만 리스트로 만들자.

```
odds = [ x for x in number if x%2 == 1 ]
```

```
odds = [ x for x in number if x%2 == 1 if x < 5 ]
```



리스트 컴프리헨션 - 반복문

2개라면?

즉, 2중 반복문이라면?



store =
[올**영, IseeU, PG25]



product =
[휴대폰 충전기, 물티슈]

```
for x in store:  
    for y in product:  
        print(x, '에 가서 ', y, '를 사자.')
```

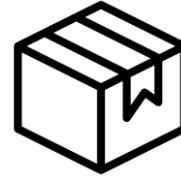

리스트 컴프리헨션 - 반복문

2개라면?

즉, 2중 반복문이라면?



store =
[올**영, IseeU, PG25]



product =
[휴대폰 충전기, 물티슈]

```
[ ( x , y ) for x in store for y product ]
```

람다식 (lambda)

2

람다식

Lambda(람다) 식

매개변수와 매개변수가 들어갈 식을 활용하여 코드를 한 줄로 표현한다.

- 단, 한 번만 쓸 수 있는 일회용이다.
- 정의와 동시에 사용할 수 있다.

```
x = 4  
y = 7  
sum = x + y
```

람다식

Lambda(람다) 식

매개변수와 매개변수가 들어갈 식을 활용하여 코드를 한 줄로 표현한다.

- 단, 한 번만 쓸 수 있는 일회용이다.
- 정의와 동시에 사용할 수 있다.

```
x = 4  
y = 7  
sum = x + y
```

어떻게 한 줄로?

람다식

Lambda(람다) 식

매개변수와 매개변수가 들어갈 식을 활용하여 코드를 한 줄로 표현한다.

- 단, 한 번만 쓸 수 있는 일회용이다.
- 정의와 동시에 사용할 수 있다.

`lambda` 매개변수 : 매개변수가 들어간 식

람다식

Lambda(람다) 식

매개변수와 매개변수가 들어갈 식을 활용하여 코드를 한 줄로 표현한다.

- 단, 한 번만 쓸 수 있는 일회용이다.
- 정의와 동시에 사용할 수 있다.

정의

```
lambda x , y : x + y
```

람다식

Lambda(람다) 식

매개변수와 매개변수가 들어갈 식을 활용하여 코드를 한 줄로 표현한다.

- 단, 한 번만 쓸 수 있는 일회용이다.
- 정의와 동시에 사용할 수 있다.

사용

```
(lambda x , y : x + y) (매개변수1, 매개변수2)
```

Summary



- Comprehension(컴프리헨션)은 파이썬 자료형인 set, list, dictionary에 적용하는 문법이다.
- Comprehension(컴프리헨션)은 반복/조건문을 한 줄로 표현하는 문법이다.
 - [x for x in (변수)]
 - [x for x in (변수) if (조건1)]
- 컴프리헨션 문법에서는 반복/조건문을 여러 번 사용할 수 있다.
 - 조건문은 여러 개의 조건을 모두 적용하는 and 의 역할을 한다.
 - 반복문은 여러 개의 반복을 겹쳐 수행하는 n중 반복문의 역할을 한다.
- Lambda(람다)식은 매개 변수와 매개 변수가 들어간 식을 활용하여 짧게 표현한 식이다.
 - lambda 매개변수 : 매개변수가 들어간 식
 - (lambda 매개변수 : 매개변수가 들어간 식)(실제로 매개변수에 들어갈 값)
 - 람다식은 일회용이다.