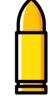
코드로 한 번에 이해되는 파이썬 (7)

실습 코드 : https://github.com/hansikyung/Python_OneTime

Today's point!



- ✔ 리스트 컴프리헨션
- ✓ lambda

리스트 컴프리헨션(Comprehension)

리스트 컴프리헨션

Comprehension(컴프리헨션)

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

- **반복문**을 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)

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)]

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)]
```

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)]
```

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)]
```



0에서 10까지의 리스트 중에서 홀수만 모아서 새 리스트로 만들자.

odds = [x for x in number]

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

0에서 10까지의 리스트 중에서 홀수만 모아서 새 리스트로 만들자.

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

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



0에서 10까지의 리스트 중에서 홀수만 모아서 새 리스트로 만들자.





0에서 10까지의 홀수 중, 5보다 작은 것만 리스트로 만들자.

0에서 10까지의 리스트 중에서 홀수만 모아서 새 리스트로 만들자.





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]



2개라면?

즉, 2중 반복문이라면?



store = [올**영, IseeU, NG25]



product = [휴대폰 충전기, 물티슈]

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

2개라면?

즉, 2중 반복문이라면?



store = [올**영, IseeU, NG25]



product = [휴대폰 충전기, 물티슈]

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

Summary

- ➤ Comprehension(컴프리헨션)은 파이썬 자료형인 set, list, dictionary에 적용하는 문법이다.
- ➤ Comprehension(컴프리헨션)은 반복/조건문을 한 줄로 표현하는 문법이다.
 - ➤ [x for x in <u>(변수)</u>]
 - ➤ [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