

# 컴프리핸션

프리젠테이션 사용 불가

## 컴프리헨션(Comprehension)

- 컴프리헨션(Comprehension)은 파이썬에서 list, dict, set과 같은 자료 구조를 생성, 추출, 검색할 때 간결하고 좀 더 이해하기 쉽게 표현하기 위한 문법이다.
- 컴프리헨션(Comprehension)은 list, dict, set 객체에 대하여 사용할 수 있으며 객체에 따라 다음과 같이 일컫는다.
  - list : list Comprehension
  - dict : dict Comprehension
  - set : set Comprehension

# List Comprehension

- 1부터 10까지의 값으로 구성된 리스트를 생성하고자 한다면 다음의 코드를 사용할 수 있다.

```
li = []
for i in range(1, 10):
    li.append(i)
print(li)
```

- 위의 코드를 Comprehension을 사용하여 기술한다면 다음과 같이 기술할 수 있다.

```
li = [ i for i in range(1, 11 )
print(li)                                     # [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

- 만약 값을 가공하여 리스트를 생성하고 싶다면 식을 사용할 수 도 있다. 아래는 1부터 10까지의 값에 10을 곱한 값으로 리스트를 생성한다.

```
li = [ i * 10 for i in range(1, 11 )
print(li)                                     # [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
```

## 조건 필터 사용하기

- 다음은 1부터 10까지의 숫자 중 짝수 만으로 리스트는 생성하는 예제이다.

```
even = []  
for i in range(1, 11):  
    if i % 2 == 0:  
        even.append(i)
```

- Comprehension에서 조건은 for ... in 뒤에 if 구문을 사용하여 지정한다.

```
even = [ i for i in range(1, 11) if i % 2 == 0 ]  
print(event)
```

실행결과>  
[2, 4, 6, 8, 10]

# Set Comprehension

- Set Comprehension은 set 타입의 객체를 생성하기 위한 문법이다.
- []를 사용하면 List Comprehension이며, {}를 사용하면 Set Comprehension이다.
- 다음은 1부터 10까지의 홀수 값 만으로 구성된 set 객체를 생성하는 예제이다.

```
odd = { i for i in range(1, 11) if i % 2 = 1 }  
print(odd)
```

실행결과>

{1, 3, 5, 7, 9}

- Set Comprehension의 사용법은 List Comprehension과 동일하다.

## Dict Comprehension

- Dict Comprehension은 사전 객체를 만들어 내기 위한 문법이다.
- {}를 사용하여 생성하며 set과는 달리 요소의 구성 형식을 key:value 형식으로 기술한다.
- 다음의 예제는 키 값만을 가지고 있는 keys 리스트와 값만을 가지는 values 리스트를 조합하여 하나의 사전 객체를 생성하는 예제이다.

```
keys = ['name', 'age', 'job']  
values = ['김기희', 20, '강사']  
d = { keys[i]:values[i] for i in range(len(names)) }  
print(d)
```

실행결과>

```
{'name': '김기희', 'age': 20, 'job': '강사'}
```

## list를 dict로 변환하기

- 다음의 코드는 list 객체 내의 요소의 인덱스를 key로 하고 값을 value로 하는 dict 객체로 변환하는 예제이다.

```
names = ['홍길동', '이순신', '김기희']  
dictionary = { i:names[i] for i in range(len(names)) }  
print(dictionary)
```

실행결과

```
{0: '김기희', 1: '이순신', 2: '홍길동'}
```

- tuple 또한 list와 동일한 방법으로 dict 객체로 변환할 수 있다.

```
names = ('홍길동', '이순신', '김기희')  
dictionary = ( i:names[i] for i in range(len(names)) )  
print(dictionary)
```

실행결과

```
{0: '김기희', 1: '이순신', 2: '홍길동'}
```