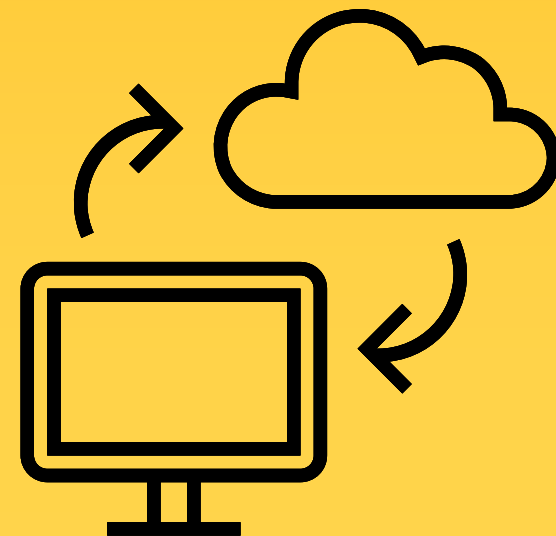


Lesson 7

Collection 자료형 셋(Set)



목차

1. 셋의 생성
2. 셋의 추가, 삭제
3. 셋의 연산

- **중복을 허용하지 않는다.**

중복 제거용으로 사용하거나 집합 연산에 사용.

- **순서X, 중복X, 추가O, 삭제O** *값만 있는 딕셔너리*

- 저장된 값들은 순서가 없다(Unordered). 따라서 인덱싱과 슬라이싱을 할 수 없다.

- **set 자료형에 저장된 값을 인덱싱으로 접근하려면 리스트나 튜플로 변환한 후 해야 한다.**



딕셔너리를 리스트나 튜플로 변경하면 키만 저장됨.

	순서	중복	추가	삭제	수정
리스트	O	O	O	O	O
튜플	O	O	X	X	X
딕셔너리	X	X	O	O	O
셋	X	X	O	O	X

- `s={값1, 값2,...}`
- `s=set(반복가능한객체)`
 - 빈 셋(set)은 반드시 `set()` 함수 사용

코드

```
s=set()
print(s)    # set()
s={1,3,3,4,1,2}
print(s)    # {1, 2, 3, 4} 중복이 제거됨
li=[1,3,3,4,1,2]
s=set(li)   # 리스트를 셋으로 변경
print(s)    # {1, 2, 3, 4}
```

- 셋(set)의 요소 추가 : `add()`, `update()`
 - ※ `add()`는 1개의 요소를 추가, `update()`는 각각의 요소로 추가.
- 셋(set)의 요소 삭제 : `remove()` 또는 `discard()`
 - ※ `remove()`는 없는 값은 오류 처리.

코드

```
s={1,2,3}
s.add(4)          # add()는 하나만 가능.
print(s)          # {1,2,3,4}
s.add(('a','b'))  # {1,2,3,4,('a','b')}
s.update((4,5,6)) # 리스트, 튜플, 셋 사용 가능
print(s)          # {1,2,3,4,5,6,('a','b')}
s.remove(1)        # 셋에서 1을 삭제.
print(s)          # {2,3,('a','b'),4,5,6}
s.remove(7)        # 오류 발생
s.discard(7)       # 요소가 없어도 오류로 처리 안됨
```

- 셋(set) 연산자 : $\&$ (교집합), $|$ (합집합), $-$ (차집합), \wedge (대칭 차집합)
- 셋(set) 함수 : intersection, union, difference, symmetric_difference

코드

```
s1={1,2,3,4}
s2={4,5,6}
print(s1 & s2)      # 교집합. {4}
print(s1 | s2)      # 합집합. {1, 2, 3, 4, 5, 6}
print(s1 - s2)      # 차집합. {1, 2, 3}
print(s1 ^ s2)      # 대칭 차집합(중복제거). {1, 2, 3, 5, 6}
print(s1.union(s2)) # 합집합. {1, 2, 3, 4, 5, 6}
```

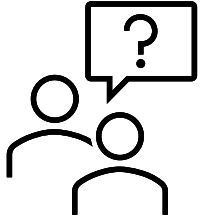
- 셋(set) 함수 : isdisjoint, issubset, issuperset
 - isdisjoint: 교집합이 없으면 True. 중복 확인 가능
 - issubset: 부분집합이면 True
 - issuperset: 포함집합이면 True

코드

```
s1={1,2,3,4,5}
s2={4,5}
print(s1.isdisjoint(s2))    # False   서로 소이어야 True
print(s2.issubset(s1))      # True    s2가 s1의 부분집합인지 확인
print(s1.issuperset(s2))    # True    s1이 s2를 포함하는 집합인지 확인
```


코드

```
li=[1,2,3,4,1,4,6,7]
li_to_set=set(li)
print(li_to_set)           # {1,2,3,4,6,7}
s={1,2,3,4,5,6}
set_to_list=list(s)
set_to_list.append(100)
print(set_to_list)        # [1,2,3,4,6,7,100]
s1=set("abcde")
s2=set("ace")
print(s1-s2)              # {"b","d"}
```



1. 셋은 순서가 (있다 없다).
2. 셋은 요소를 추가할 수 (있다 없다).
3. 셋은 요소를 수정할 수 (있다 없다).
4. 셋은 중복된 요소를 가질 수 (있다 없다).
5. 셋은 생성하려면 ()를 사용하면 된다.
6. 빈 셋을 만들려면?

> { }

> Set()

코드

```
# k는 한국인이 여행하고 싶은 국가
# o는 한국외 지역의 사람들이 선호하는 국가
k={"프랑스","미국","스위스","이탈리아","호주","일본"}
o={"미국","이집트","캐나다","프랑스","영국","중국"}

print("전세계인이 선호하는 곳:",k & o)
print("선호하는 모든 국가:",k | o)
print("한국인만 선호하는 곳:",k - o)
```

전세계인이 선호하는 곳: {"프랑스", "미국"}

선호하는 모든 국가: {"스위스", "호주", "영국", "이집트", "이탈리아", "일본", "중국", "프랑스", "미국", "캐나다"}

한국인만 선호하는 곳: {"호주", "일본", "이탈리아", "스위스"}