

# 비시퀀스형 데이터 구조

🕒 작성일시	@2022년 7월 25일 오후 10:49
☰ 내용	<span>문법</span> <span>자료구조</span> <span>제어문</span> <span>파이썬</span>
▼ 주차	2주차
📎 자료	

## 셋(Set)

문법	설명
s.copy()	셋의 얇은 복사본을 반환
s.add()	항목 x가 셋 s에 없다면 추가
s.pop()	셋 s에서 랜덤하게 항목을 반환하고, 해당 항목을 제거 set이 비어있을 경우, KeyError
s.remove(s)	항목 x를 셋 s에서 삭제 함축이 존재하지 않을 경우, KeyError
s.discard(x)	항목 x가 셋 s에 있는 경우, 항목 x를 셋 s에서 삭제
s.update(t)	셋 t에 있는 모든 항목 중 셋 s에 없는 항목을 추가
s.clear()	모든 항목을 제거
s.isdisjoint(t)	셋 s가 셋 t의 서로 같은 항목을 하나라도 갖고 있지 않은 경우, True반환
s.issubset(t)	셋 s가 셋 t의 하위 셋인 경우, True 반환
s.issuperset(t)	셋 s가 셋 t의 상위 셋인 경우, True 반환

- set 원소 제거 및 추가

```
a = {'apple', 'banana', 'grape'}

a.update(['apple', 'berry', 'melon'])
print(a) # {'berry', 'apple', 'grape', 'melon', 'banana'}

a.remove('apple')
print(a) # {'melon', 'banana', 'berry', 'grape'}

a.remove('apple')
print(a) # KeyError: 'apple'

a.discard('apple')
print(a) # {'grape', 'melon', 'berry', 'banana'}
```

- set 집합 관련 메서드

```
a = {'apple', 'banana', 'grape'}
b = {'berry', 'melon'}
c = {'apple', 'banana', 'grape', 'berry', 'melon'}

print(a.isdisjoint(b)) # True
print(a.isdisjoint(c)) # False
print(a.issubset(c)) # True
print(c.issuperset(a)) # True
```

## 딕셔너리(Dictionary)

문법	설명
d.clear()	모든 항목을 제거
d.copy()	딕셔너리 d의 얇은 복사본을 반환
d.keys()	딕셔너리 d의 모든 키를 담은 뷰를 반환
d.values()	딕셔너리 d의 모든 값을 담은 뷰를 반환
d.items()	딕셔너리 d의 모든 키-값의 쌍을 담은 뷰를 반환
d.get(k)	키 k의 값을 반환하는데, 키 k가 딕셔너리 d에 없을 경우 None을 반환
d.get(k, v)	키 k의 값을 반환하는데, 키 k가 딕셔너리 d에 없을 경우 v를 반환
d.pop(k)	키 k의 값을 반환하고 키 k인 항목을 딕셔너리 d에서 삭제하는데, 키 k가 딕셔너리 d에 없을 경우 KeyError를 발생
d.pop(k, v)	키 k의 값을 반환하고 키 k인 항목을 딕셔너리 d에서 삭제하는데, 키 k가 딕셔너리 d에 없을 경우 v를 발생
d.update([other])	딕셔너리 d의 값을 매핑하여 업데이트