

```
/* elice */
```

파이썬 실전 데이터 분석

데이터형 변환

커리큘럼

1 ○

트럼프 대통령 트윗으로 알아보는 미국의 주요 정치 이슈

조건문, 반복문, 리스트를 복합적으로 활용해 트럼프 대통령의 트윗을 분석하고, 알아보기 쉬운 시각화로 표현해 봅니다.

2 ○

영어 단어 모음으로 시작하는 데이터 시각화

파이썬 라이브러리와 파일 불러오기를 이용해 영문 자료에서 가장 많이 사용되는 단어 10,000개를 알아봅니다.

커리큘럼

3 ○

넷플릭스 시청 데이터로 알아 보는 데이터형 변환

넷플릭스 데이터 대회에 사용된 JSON 데이터를 파이썬의 사전으로 변환하여 비슷한 성향의 유저를 찾아 봅니다.

4 ○

TED 강연을 통해 접해 보는 복잡한 형태의 데이터

CSV, JSON, TXT 등 다양한 형태와 앞서 배운 개념을 종합적으로 다루며, TED 강연에서 가장 인기 있는 강연과 핫한 주제를 분석/시각화 해 봅니다.

목차

1. 딕셔너리 (Dictionary)
2. 집합 (Set)
3. 그래프 설정하기

딕셔너리

딕셔너리

```
{ key: value }
```

- **key:** 값을 찾기 위해 넣어 주는 데이터
- **value:** 찾고자 하는 데이터

딕셔너리

```
empty_dict = {}  
new_dict = {  
    'apple': '사과',  
    'book': '책',  
    'human': '사람',  
}
```

딕셔너리

```
dictionary = {}  
dictionary['apple'] = '사과'  
dictionary['book'] = '책'  
dictionary['human'] = '사람'  
print(dictionary['book'])
```


딕셔너리 vs. 리스트

```
# {id: 이름}
accounts = {
    "kdhong.elice": "Kildong Hong",
    ...
}
print(accounts["kdhong.elice"])
```

딕셔너리 vs. 리스트

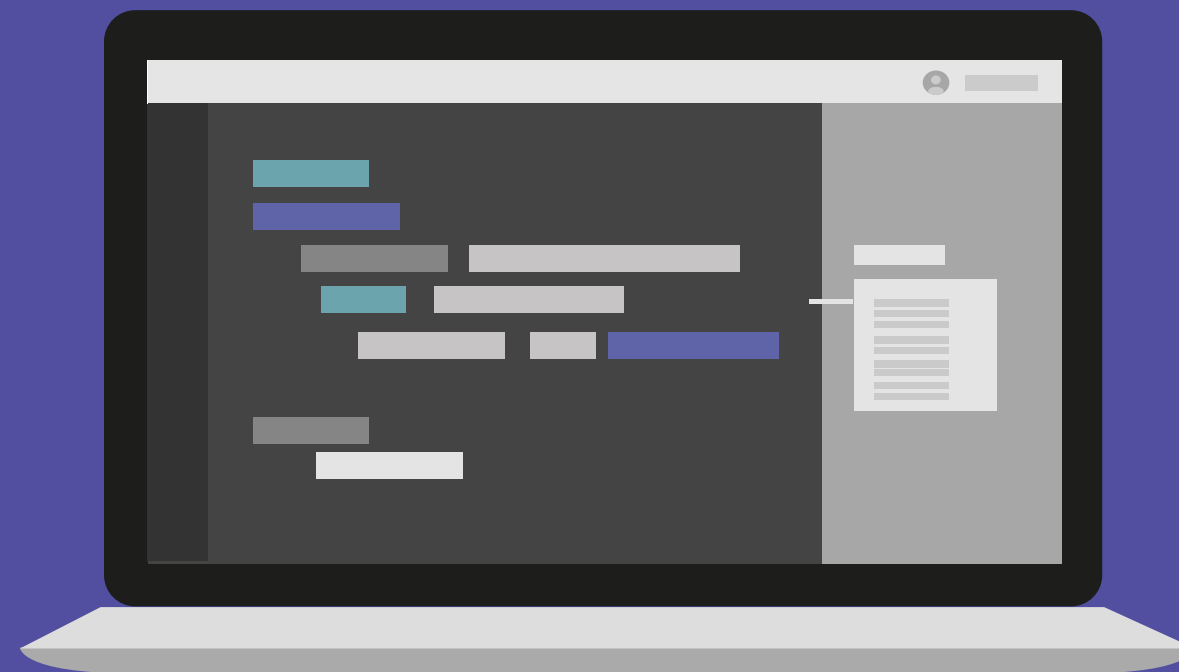
```
# [(id, 이름)]  
accounts = [  
    ("kdhong.elice", "Kildong Hong"),  
    ...  
]
```

딕셔너리 vs. 리스트

```
# [(아이디, 이름)]  
for id_, name in accounts:  
    if id_ == "kdhong.elice":  
        print(name)
```

모든 아이디를 확인해야 하므로, 데이터가 많을 경우 수십 배 까지도 성능 차이가 남

[실습1] 딕셔너리



```
/* elice */
```

딕셔너리의 키

```
# {[ID, 비밀번호]: 계정 정보}
kdhong = ["kdhong", "cantcalldad"]
accounts = {
    kdhong: ('Kildong Hong', ...),
}
```

딕셔너리의 키

```
# {[ID, 비밀번호]: 계정 정보}
kdhong = ["kdhong", "cantcalldad"]
accounts = {
    kdhong: ('Kildong Hong', ...),
}
kdhong[0] = "kdhong.elice"
```

딕셔너리의 키

변할 수 없는 값만이
key가 될 수 있다

```
# {(ID, 비밀번호): 계정 정보}
```

```
kdhong = ("kdhong", "cantcalldad")
```

```
accounts = {
```

```
    kdhong: ('Kildong Hong', ...),
```

```
}
```

```
kdhong[0] = "kdhong.elice"    # Error
```

딕셔너리의 키 확인하기

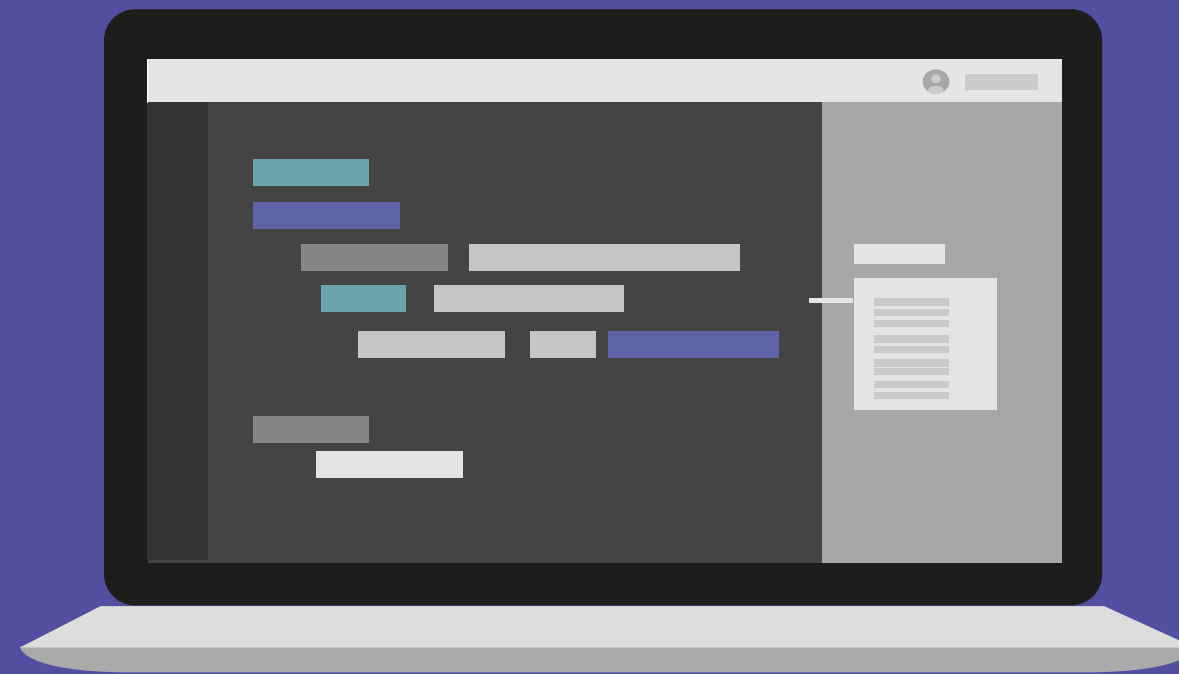
```
# {id: 이름}
accounts = {
    "kdhong": "Kildong Hong",
}

print("kdhong" in accounts)    # True
print("elice" in accounts)    # False
```


딕셔너리 순회하기

```
accounts = {  
    "kdhong": "Kildong Hong",  
}  
  
for username, name in accounts.items():  
    print(username + " - " + name)
```

[실습 2] items()



```
/* elice */
```

JSON

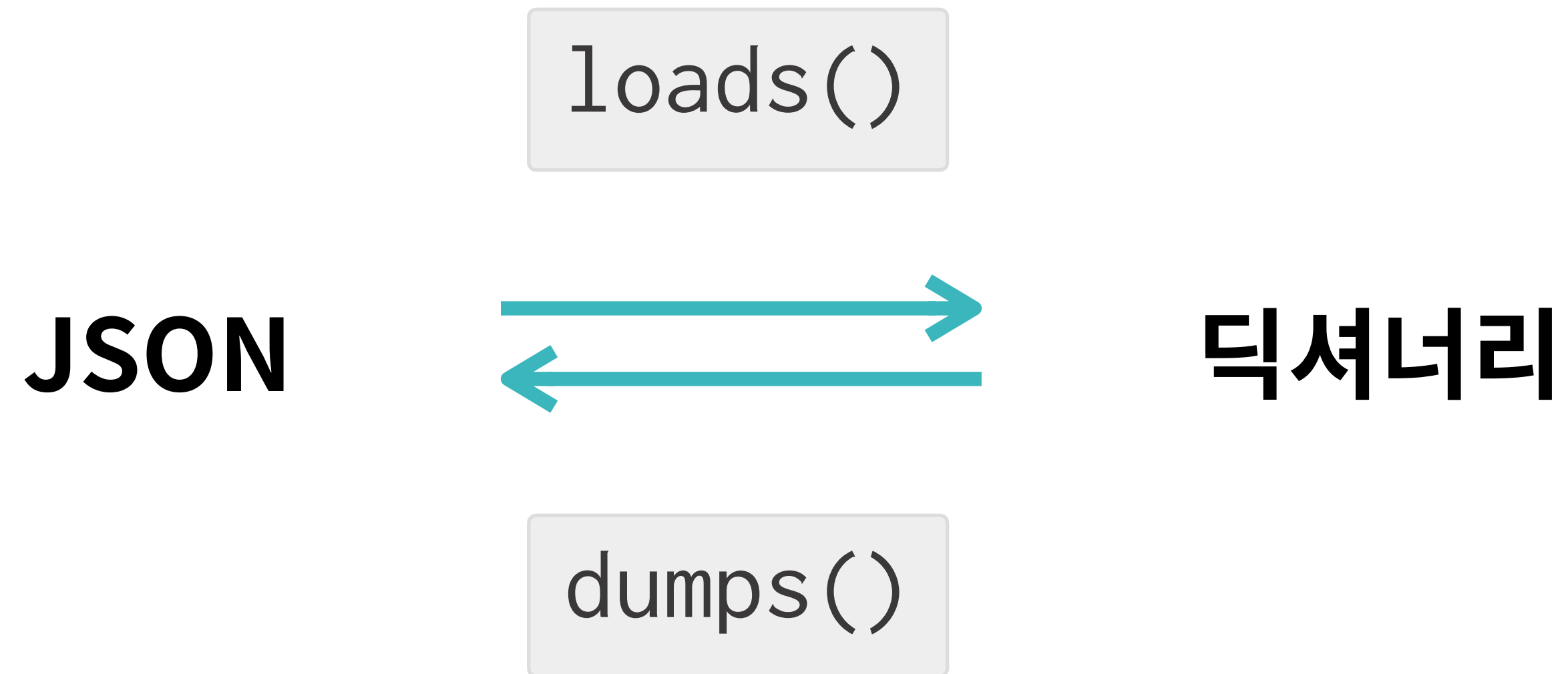
```
{ key: value }
```

"JavaScript Object Notation"

JSON

- 웹 환경에서 데이터를 주고 받는 가장 표준적인 방식
- 키를 이용하여 원하는 데이터만 **빠르게** 추출 가능
- 데이터가 쉽게 **오염되지 않음**
- 다른 포맷에 비해 **용량이 조금 큰 편**

JSON과 딕셔너리 변환



[실습 3] JSON 데이터 다루기



`/* elice */`

집합

집합

중복이 없다

순서가 없다

집합 만들기

셋 다 같은 값

set1 = {1, 2, 3}

set2 = set([1, 2, 3])

set3 = {3, 2, 3, 1}

원소 추가/삭제

```
num_set = {1, 3, 5, 7}
```

```
num_set.add(9)
```

```
num_set.update([3, 15, 4])
```

```
num_set.remove(7)
```

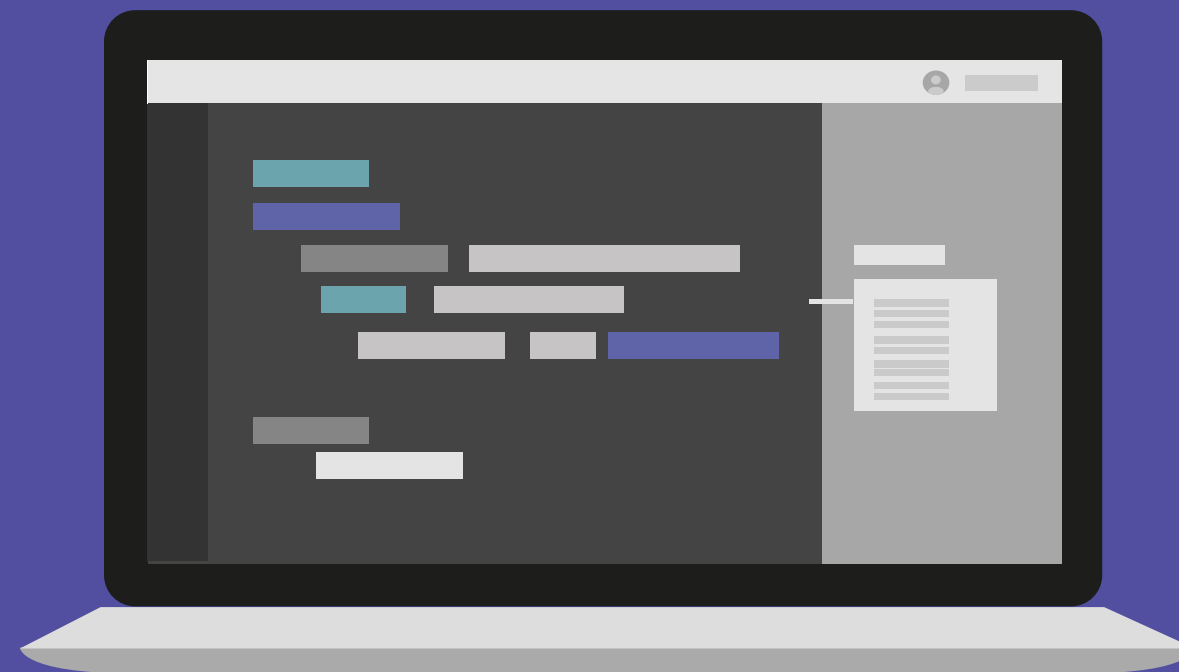
```
num_set.discard(13)
```

집합 다루기

```
num_set = {1, 3, 5, 7}

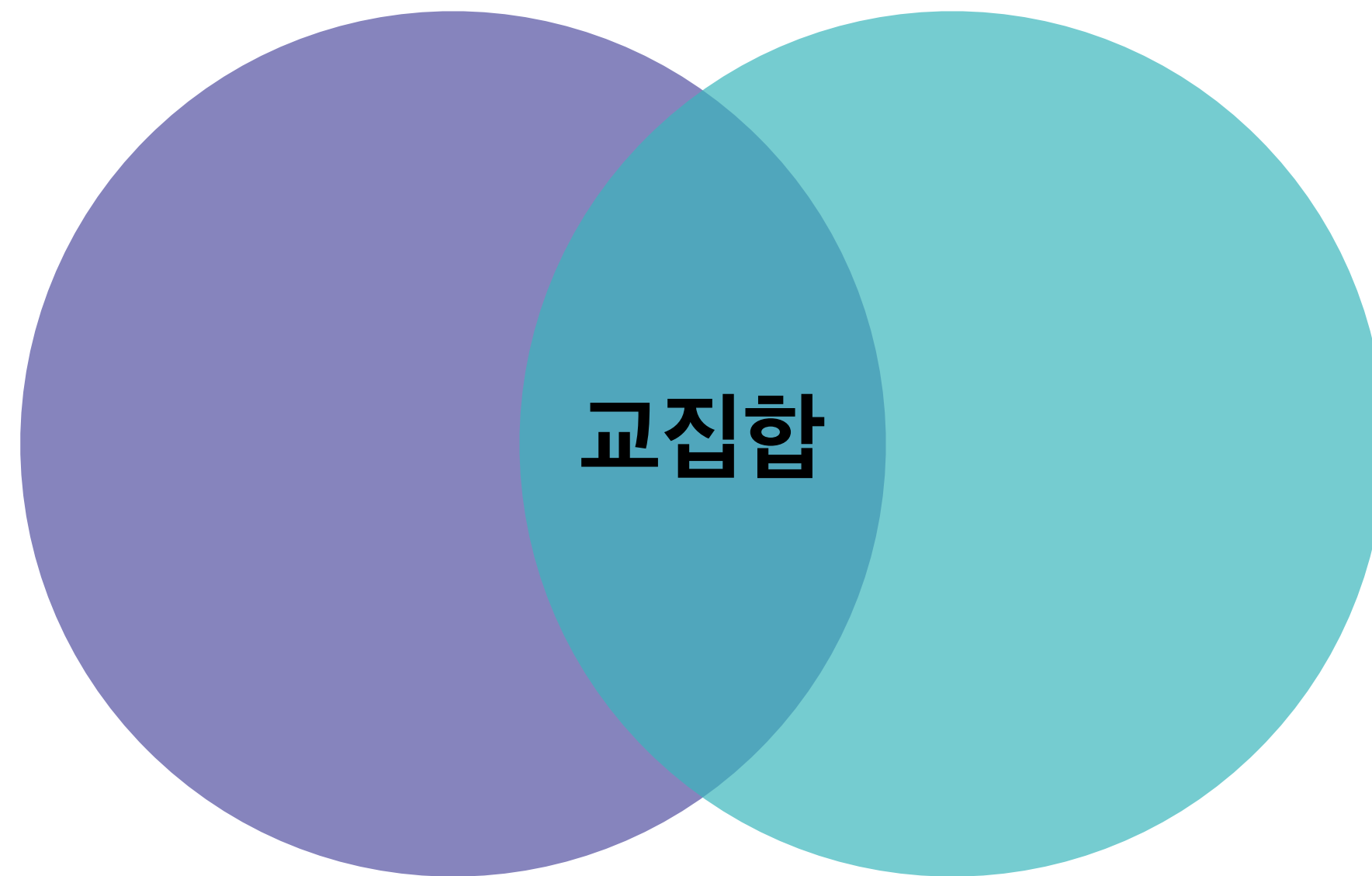
print(6 in num_set)          # False
print(len(num_set))          # 4
```

[실습 4] set()

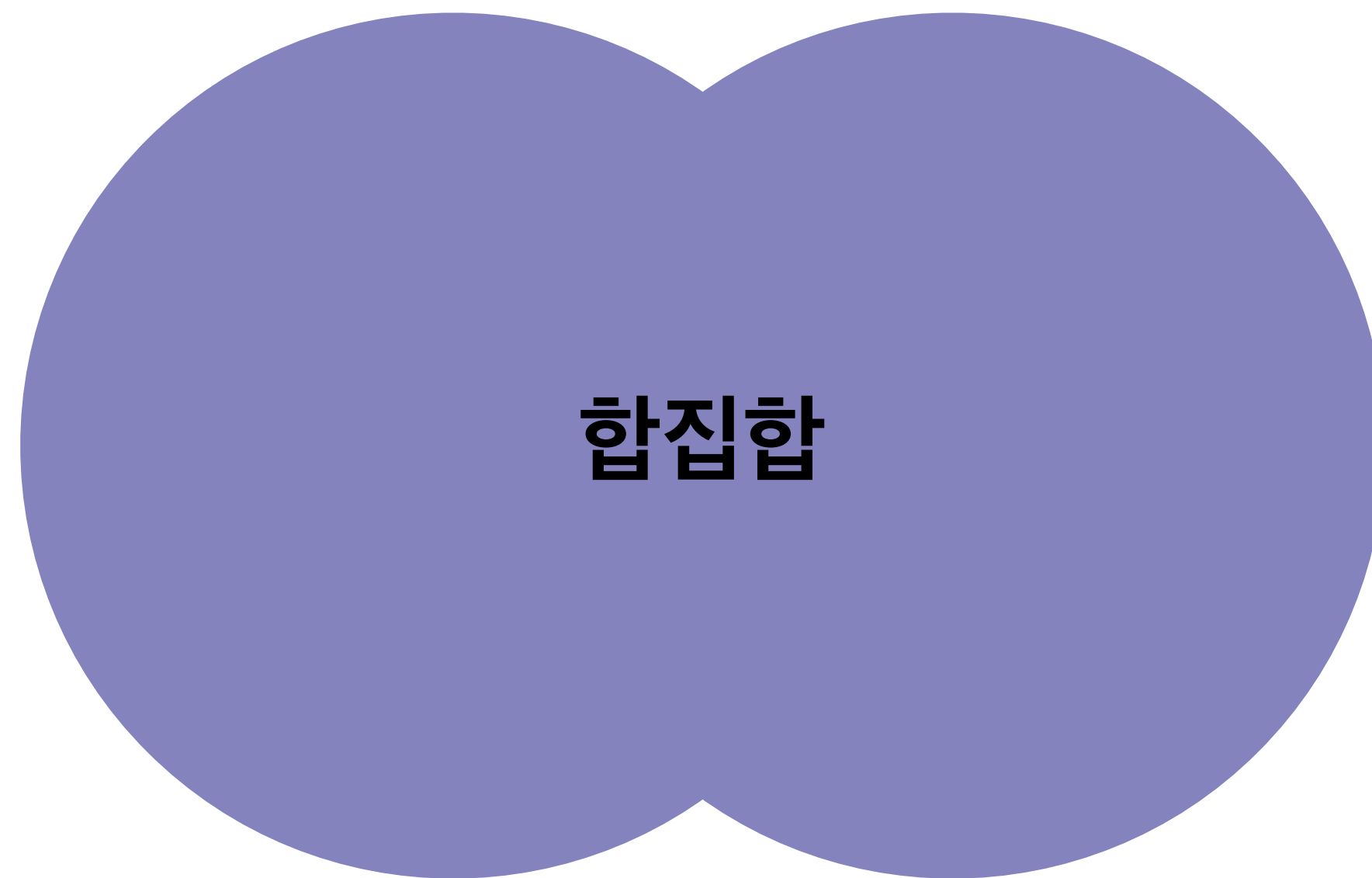


```
/* elice */
```

집합 연산

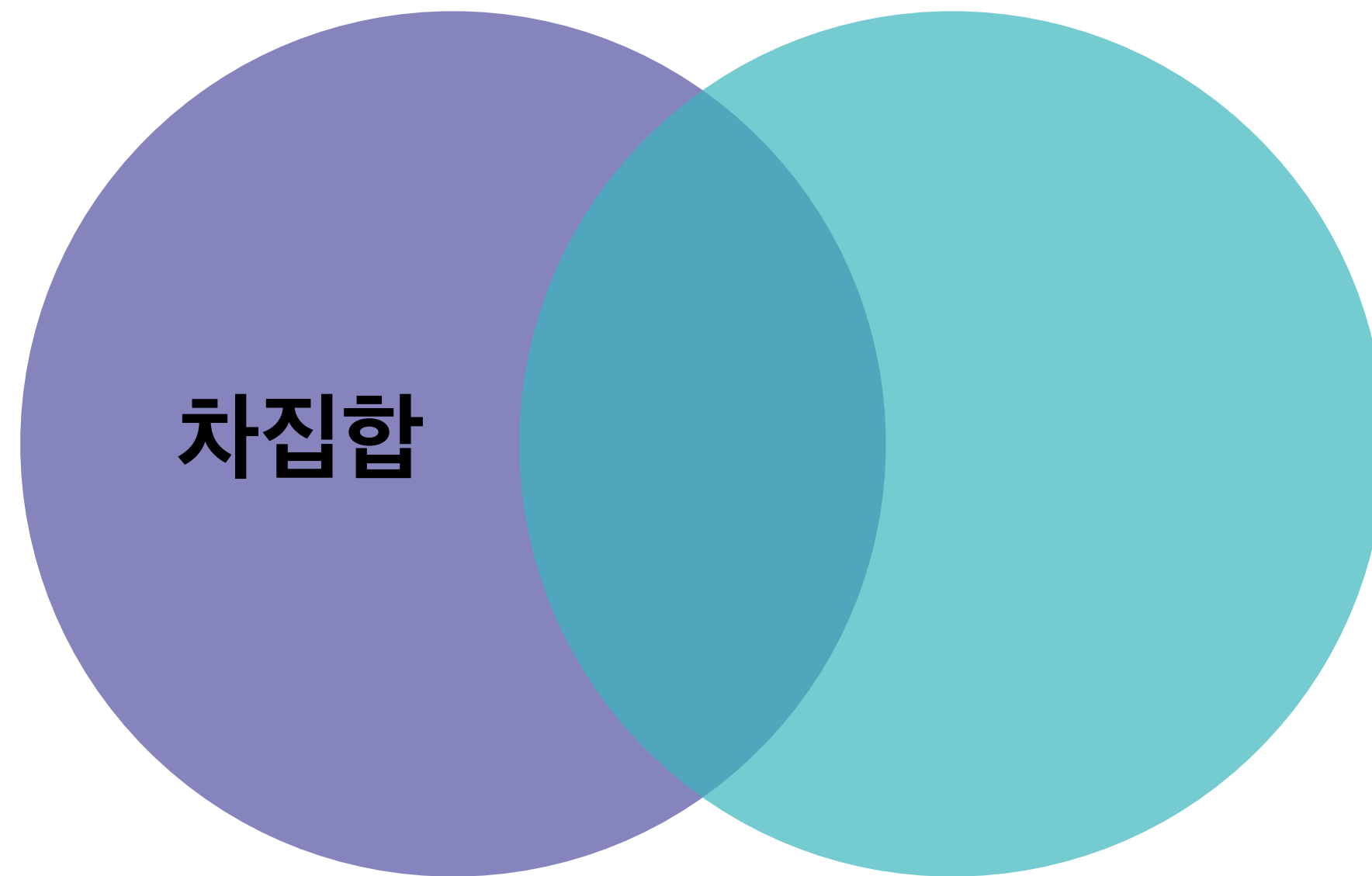


집합 연산

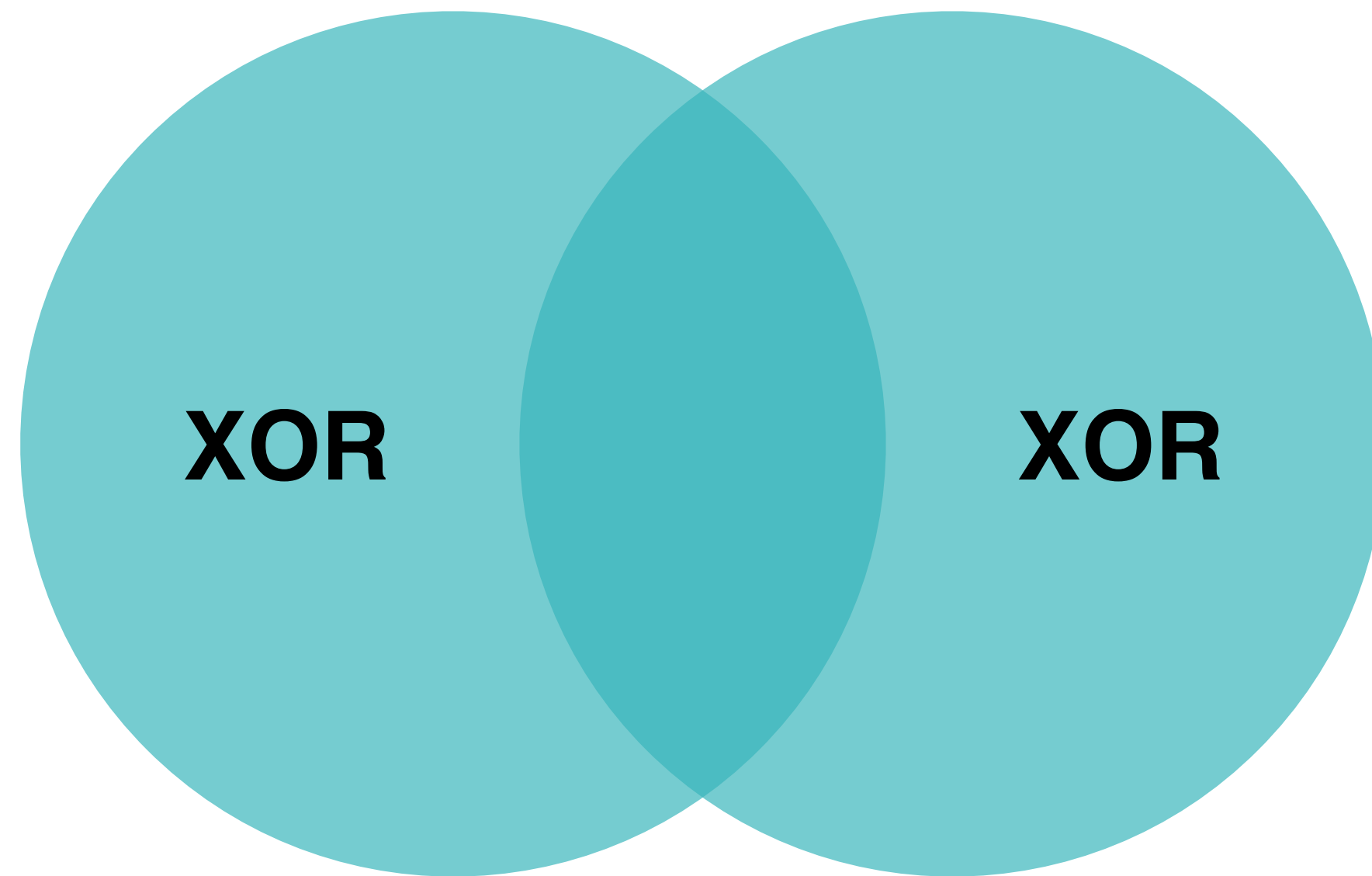


합집합

집합 연산



집합 연산



집합 연산

set1 = {1, 3, 5, 7}

set2 = {1, 3, 9, 27}

집합 연산

```
union = set1 | set2          # 합집합
```

```
intersection = set1 & set2    # 교집합
```

```
diff = set1 - set2           # 차집합
```

```
xor = set1 ^ set2            # XOR
```

[실습5] 집합 연산자



```
/* elice */
```

그래프 다루기

matplotlib으로 그래프 설정

- 제목 달기
- 축 별로 **라벨** 추가하기
- 막대 그래프의 **tick** 예쁘게 그리기
- **여백** 조정하기

[실습6] 차트 설정하기



`/* elice */`

`/* elice */`

문의 및 연락처

academy.elice.io

contact@elice.io

facebook.com/elice.io

medium.com/elice