

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.

파일 다루기

파일 열기/닫기

```
file = open('data.txt')  
content = file.read()  
file.close()
```

with

```
file = open('data.txt')  
content = file.read()  
file.close()
```

for 반복문

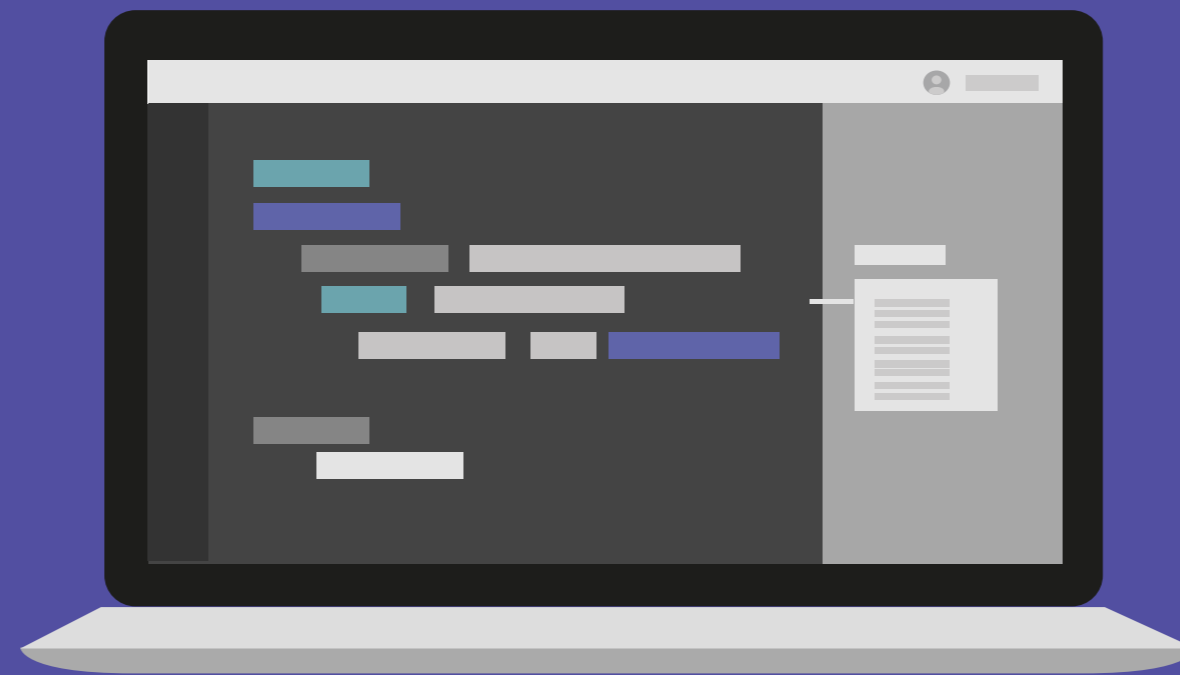
0부터 9까지 한 줄 씩 출력합니다.

```
for num in range(10):  
    print(num)
```

for 반복문

```
fruits = ["사과", "바나나", "키위"]  
  
# 과일의 이름과 번호를 함께 출력합니다.  
for i in range(len(fruits)):  
    print("과일" + str(i+1) + ": fruits[i]")
```

[실습1] for, in



```
/* elice */
```

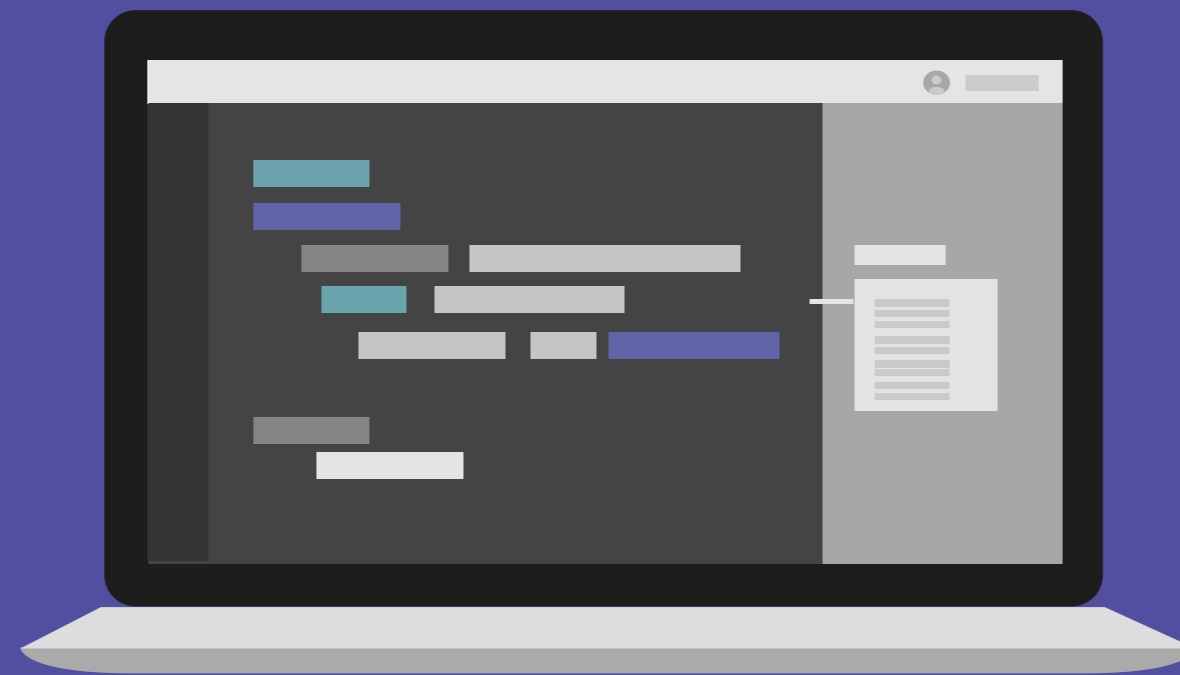

인덱싱

```
fruits = ["사과", "바나나", "키위", "배"]  
last_fruit = fruits[-1]  
tropical_fruits = fruits[1:3]  
no_apple = fruits[1:]  
no_pear = fruits[:3]
```

문자열 인덱싱

```
word = "superman"
print(word[3])           # 'e'
print(word[-2])         # 'a'
print(word[5:])         # 'man'
print(word[:5])         # 'super'
```

[실습 2] 인덱싱



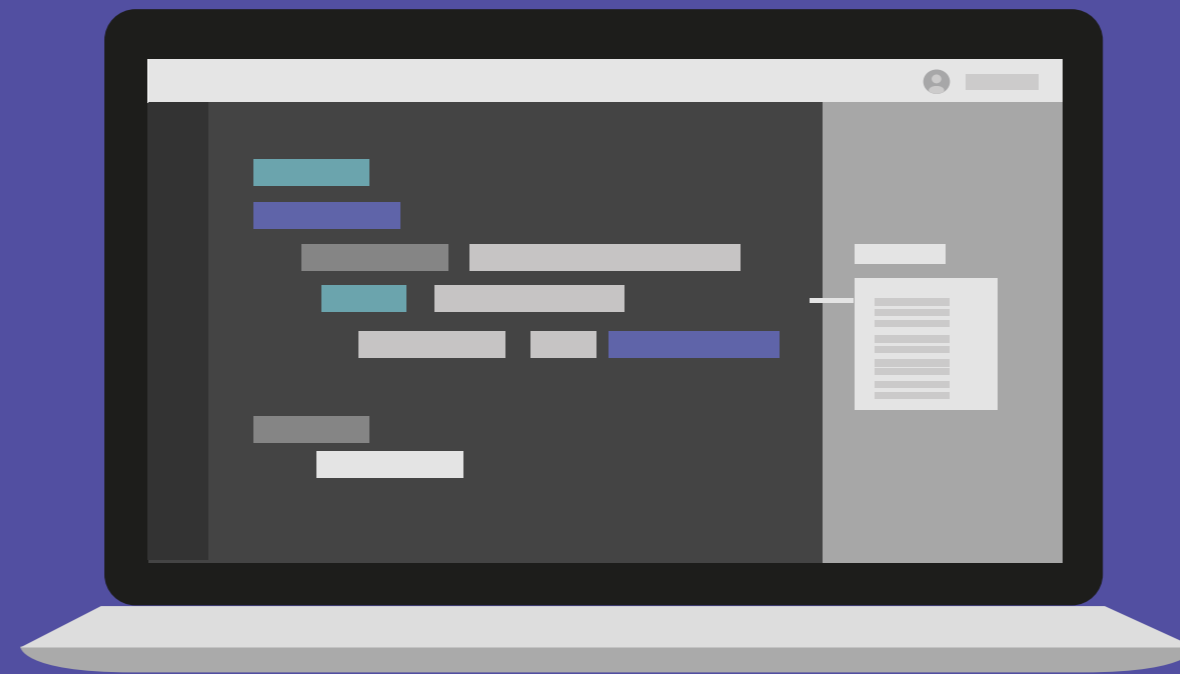
`/* elice */`

문자열 함수

.startswith()

```
word = "superman"  
print(word.startswith('s'))      # True  
  
if word.startswith('a'):  
    print("a로 시작하는 단어입니다.")
```

[실습3] startswith()



```
/* elice */
```

.split()

```
intro = "제 이름은 엘리스입니다."  
print(intro.split())  
>>> ["제", "이름은", "엘리스입니다."]
```

```
fruits = "사과,귤,배,바나나"  
print(fruits.split(','))  
>>> ["사과", "귤", "배", "바나나"]
```

.split()

```
numbers = " 1 2 3 "
```

```
print(numbers.split())
```

```
>>> ['1', '2', '3']
```

```
print(numbers.split(' '))
```

```
>>> ['', '', '1', '', '2', '', '3', '', '']
```


대표적인 공백 문자

' '

빈칸
(스페이스바)

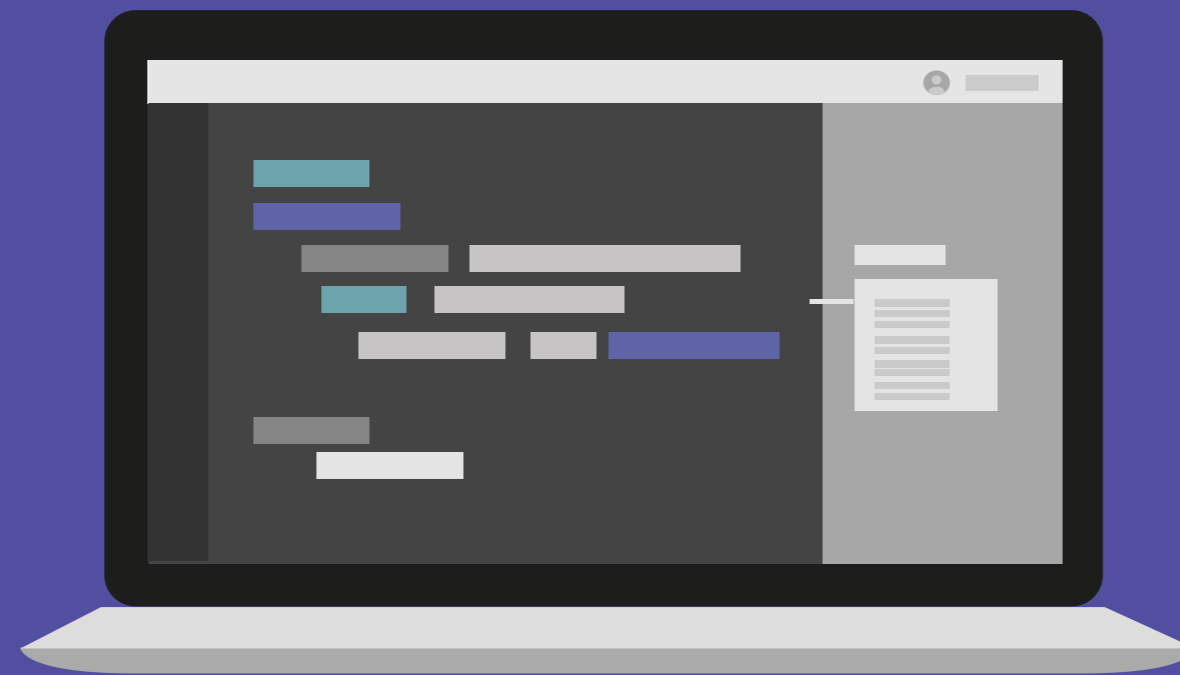
'\t'

Tab
(Tab 키)

'\n'

Newline
(엔터 키)

[실습4] split()



```
/* elice */
```

.append()

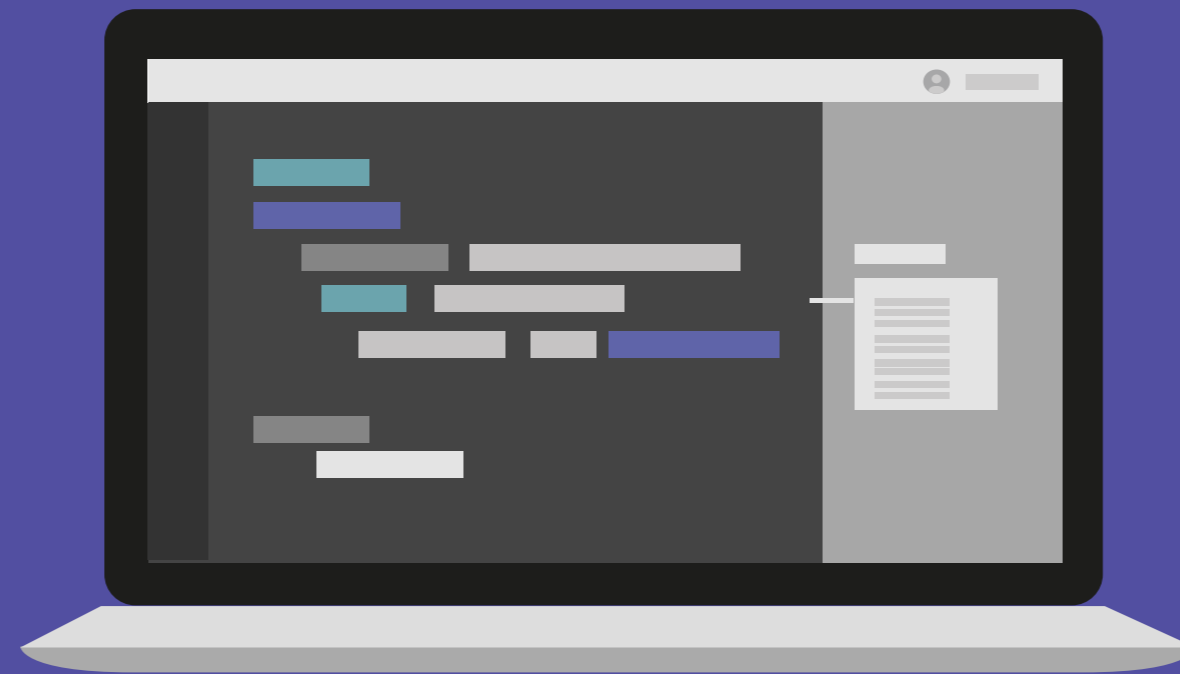
```
numbers = []  
numbers.append(1)  
print(numbers)           # [1]  
numbers.append(2)  
print(numbers)           # [1, 2]
```

.append()

```
numbers = [1, 2, 10, 17]
small_numbers = []

for num in numbers:
    if number < 10:
        small_numbers.append(num)
```

[실습5] append()



```
/* elice */
```

대소문자 변환

```
intro = "My name is Elice!"  
print(intro.upper())  
>>> "MY NAME IS ELICE!"  
print(intro.lower())  
>>> "my name is elice!"
```

append()와 lower()의 차이

```
words = ['hello']  
words.append('elice')  
print(words)  
>>> ['hello', 'elice']
```

append()와 lower()의 차이

```
intro = "My name is Elice"  
intro.lower()  
print(intro)  
>>> ???
```


append()와 lower()의 차이

```
intro = "My name is Elice"
```

```
intro.lower()
```

원래 문자열을 직접 수정하지 않는다!

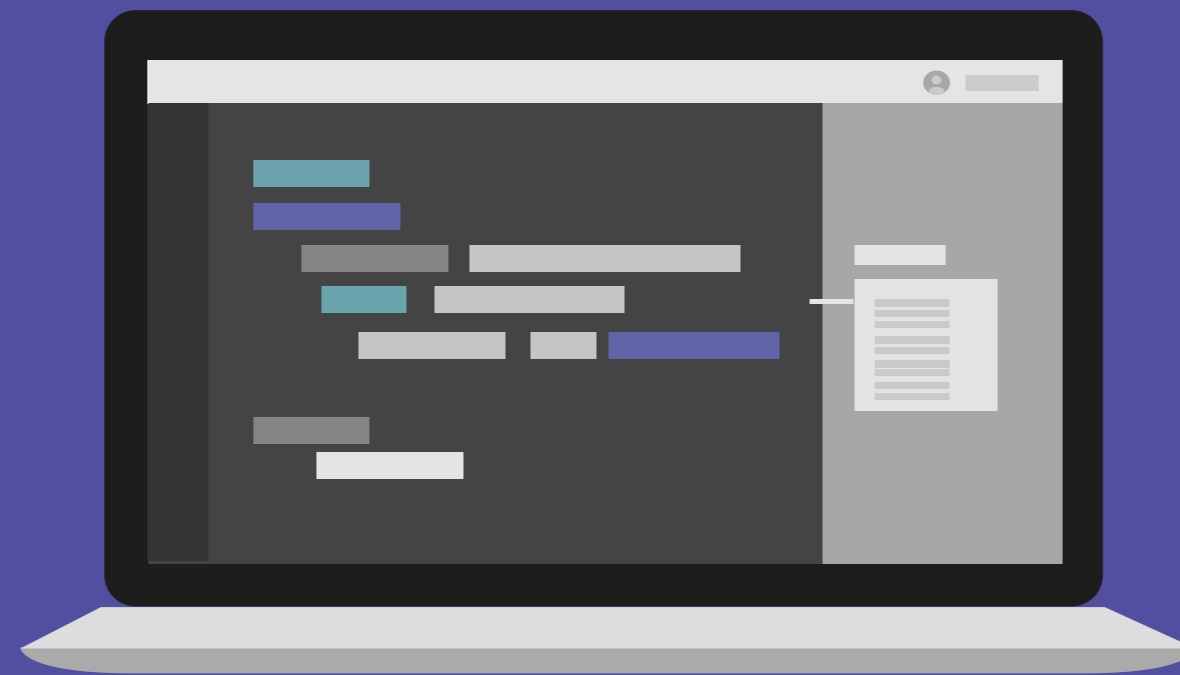
```
print(intro)
```

```
>>> "My name is Elice"
```

append()와 lower()의 차이

```
intro = "My name is Elice"  
lower_intro = intro.lower()  
print(lower_intro)  
>>> "my name is elice"
```

[실습6] lower()



```
/* elice */
```

.replace()

```
intro = "제 이름은 Elice입니다."  
print(intro.replace('Elice', '엘리스'))  
>>> "제 이름은 엘리스입니다."
```

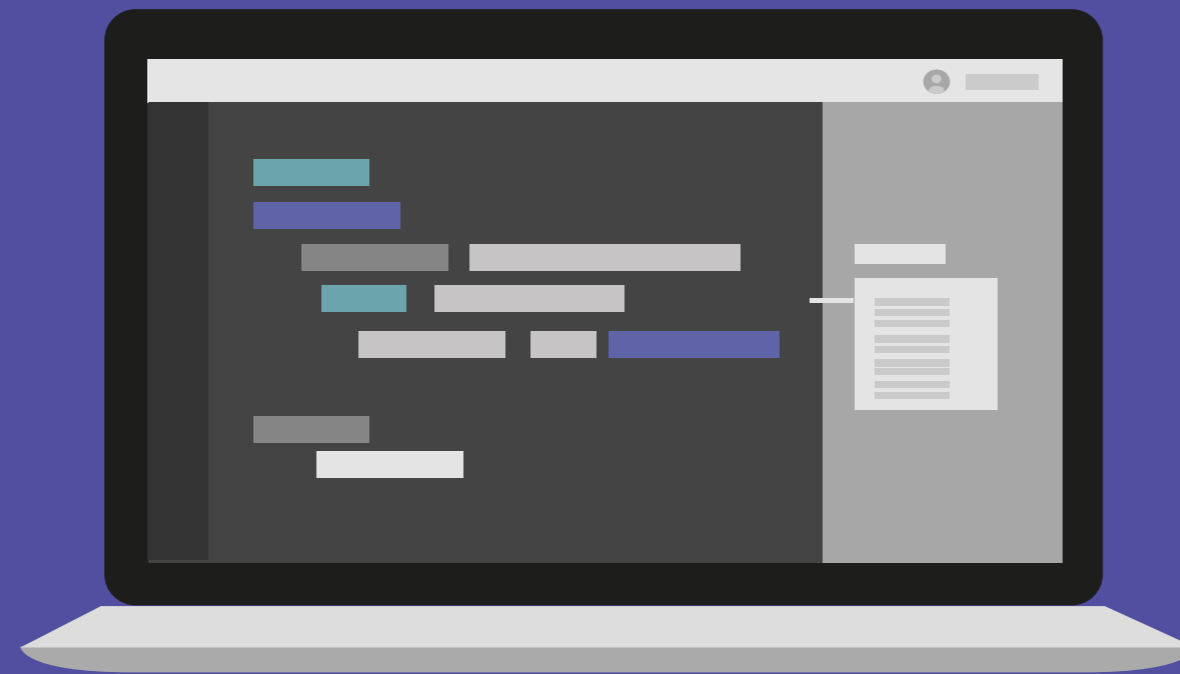
.replace()

```
intro = "제 이름은 Elice입니다."  
print(intro.replace(' ', ''))  
>>> "제이름은Elice입니다."
```

.replace()

```
intro = "제 이름은 Elice입니다."  
intro.replace(' ', '')  
print(intro)  
>>> "제 이름은 Elice입니다."
```

[실습7] replace()



```
/* elice */
```

`/* elice */`

문의 및 연락처

academy.elice.io

contact@elice.io

facebook.com/elice.io

medium.com/elice

/* elice */

파이썬 실전 데이터 분석

텍스트 파일 분석



김건우 선생님

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.

목차

1. 파일 다루기
2. 데이터 구조 다루기
3. 그래프 다루기

파일 다루기

파일 열기/닫기

```
file = open('data.txt')  
content = file.read()  
file.close()
```

파일 자동으로 닫기

```
with open('data.txt') as file:  
    content = file.read()  
# file.close() - 필요 없음
```

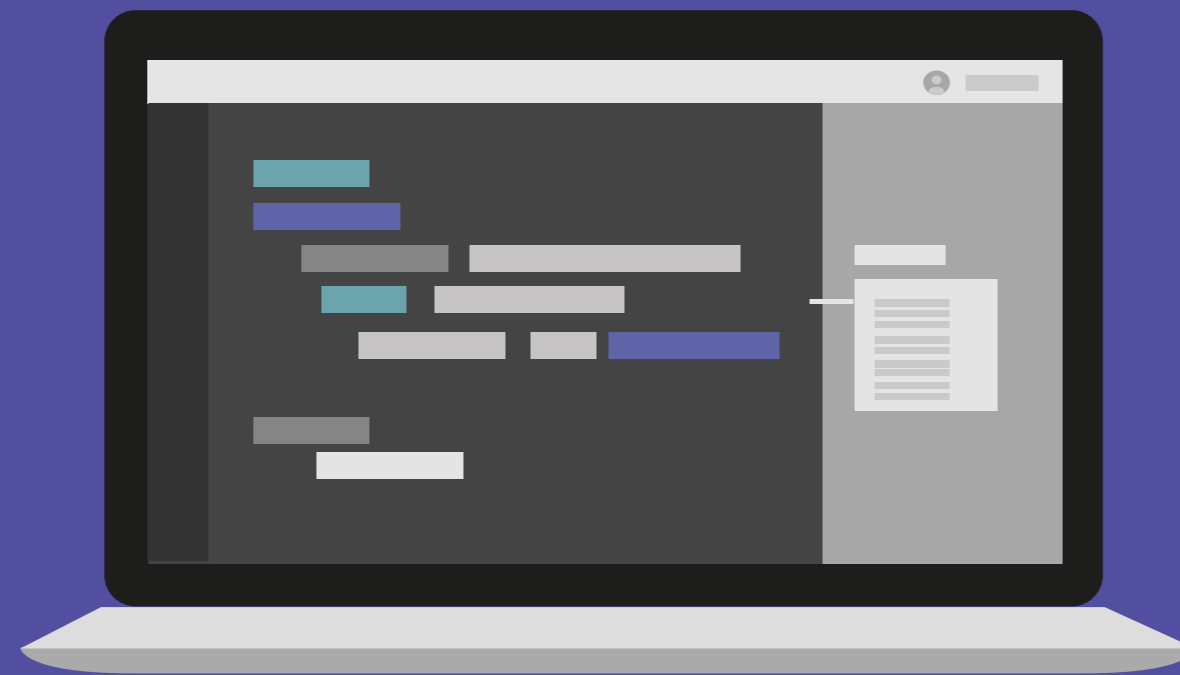
줄 단위로 읽기

```
contents = []  
with open('data.txt') as file:  
    for line in file:  
        contents.append(line)
```

파일의 모드

```
# 쓰기 (Write) 모드로 파일을 연다  
with open('data.txt', 'w') as file:  
    file.write('Hello')
```


[실습1] 파일 열고 읽기



```
/* elice */
```

데이터 구조 다루기

튜플 (Tuple)

```
hello = ('안녕하세요', 'hello', 'bonjour')
```

```
apple = ('사과', 'apple', 'pomme')
```

```
red = ('빨갱다', 'red', 'rouge')
```

튜플 vs. 리스트

공통점

- 순서가 있는 원소들의 집합

차이점

- 각 원소의 값을 수정할 수 없음
- 원소의 개수를 바꿀 수 없음

리스트 변형

```
hello = ['안녕하세요', 'hello', 'bonjour']
```

```
hello[0] = '안녕'
```

```
# ['안녕', 'hello', 'bonjour']
```

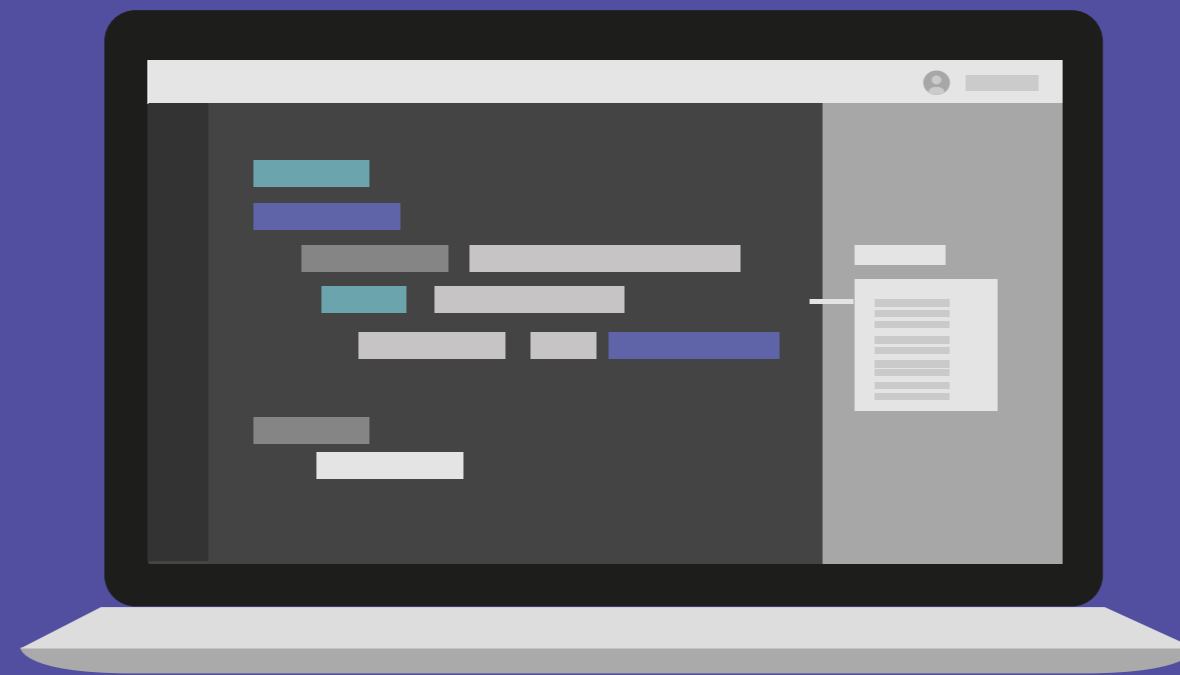
```
hello.append('ni hao')
```

```
# ['안녕', 'hello', 'bonjour', 'ni hao']
```

튜플 변형

```
hello = ('안녕하세요', 'hello', 'bonjour')  
hello[0] = '안녕' # 에러  
hello.append('ni hao') # 그런 함수 없음
```

[실습2] 튜플



`/* elice */`

리스트로 리스트 만들기

```
words = ['life', 'love', 'faith']  
first_letters = []  
for word in words:  
    first_letters.append(word[0])
```


리스트로 리스트 만들기

```
words = ['life', 'love', 'faith']  
first_letters = [word[0] for word in words]
```

리스트로 리스트 만들기

```
numbers = [1, 3, 5, 7]
new_numbers = []
for n in numbers:
    new_numbers.append(n + 1)
```

리스트로 리스트 만들기

```
numbers = [1, 3, 5, 7]
```

```
new_numbers = [n + 1 for n in numbers]
```

특정 원소 걸러내기

```
numbers = [1, 3, 4, 5, 6, 7]
even = []
for n in numbers:
    if n % 2 == 0:
        even.append(n)
```

특정 원소 걸러내기

```
numbers = [1, 3, 4, 5, 6, 7]
```

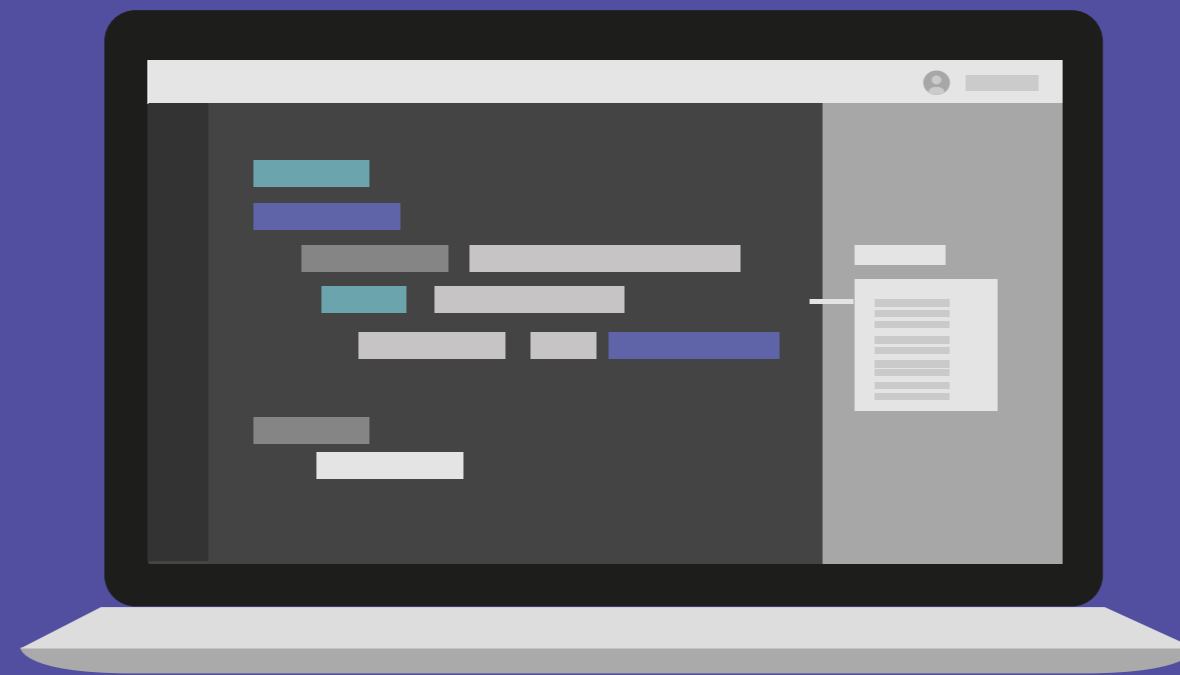
```
even = [n for n in numbers if n % 2 == 0]
```

특정 원소 걸러내기

```
numbers = [1, 3, 4, 5, 6, 7]
```

```
odd = [n + 1 for n in numbers if n % 2 == 0]
```

[실습3] 한 줄로 코드 짧게 쓰기



```
/* elice */
```

데이터 정렬하기

```
numbers = [-1, 3, -4, 5, 6, 100]  
sort_by_abs = sorted(numbers, key=abs)
```

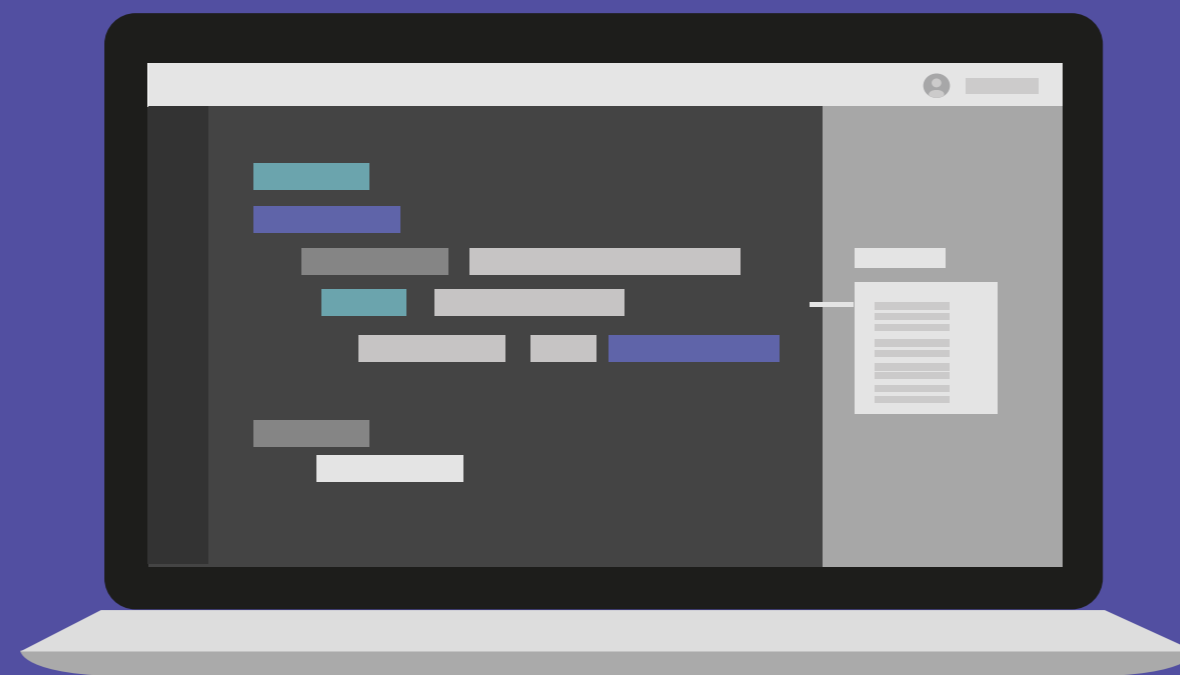

데이터 정렬하기

```
fruits = ['cherry', 'apple', 'banana']  
sort_by_alphabet = sorted(fruits)  
# ['apple', 'banana', 'cherry']
```

데이터 정렬하기

```
def reverse(word):  
    return str(reversed(word))  
  
fruits = ['cherry', 'apple', 'banana']  
sort_by_last = sorted(fruits, key=reverse)  
# ['banana', 'apple', 'cherry']
```

[실습4] 데이터 정렬하기



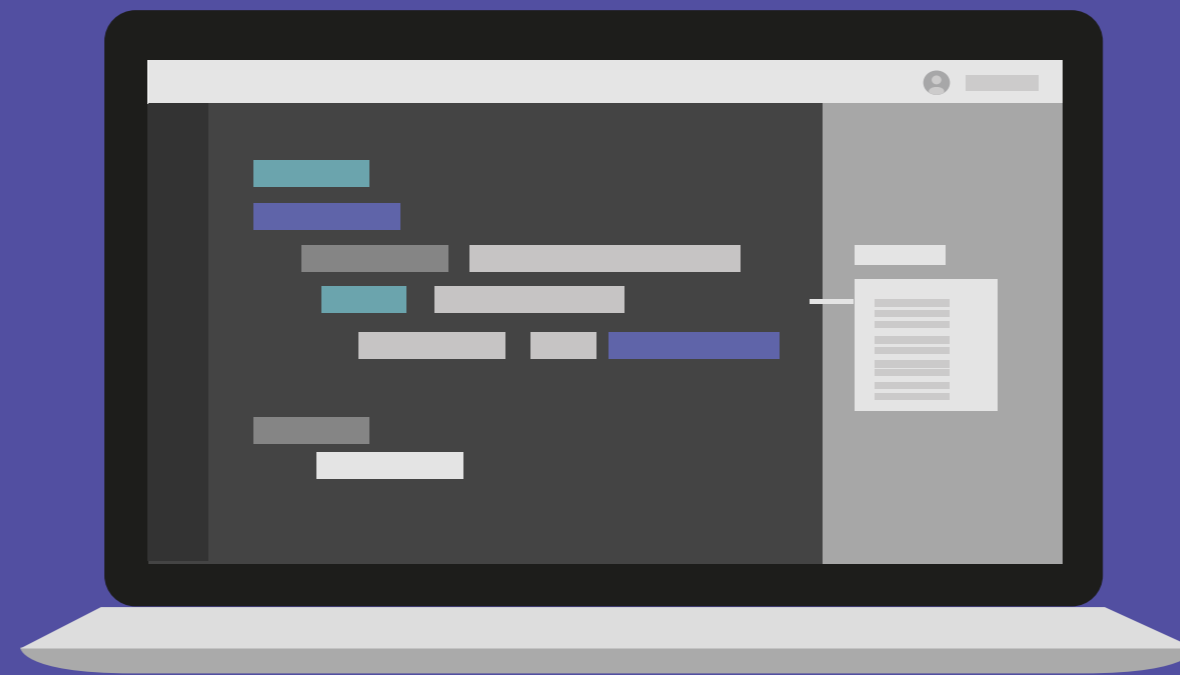
`/* elice */`

그래프 다루기

matplotlib

- Mathematical Plot Library
- 파이썬에서 그래프를 그릴 수 있게 하는 라이브러리
- 꺾은선 그래프, 막대 그래프 등을 모두 지원

[실습5] 차트 그리기



`/* elice */`

`/* elice */`

문의 및 연락처

academy.elice.io

contact@elice.io

facebook.com/elice.io

medium.com/elice

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.alice": "Kildong Hong",
    ...
}
print(accounts["kdhong.alice"])
```


딕셔너리 vs. 리스트

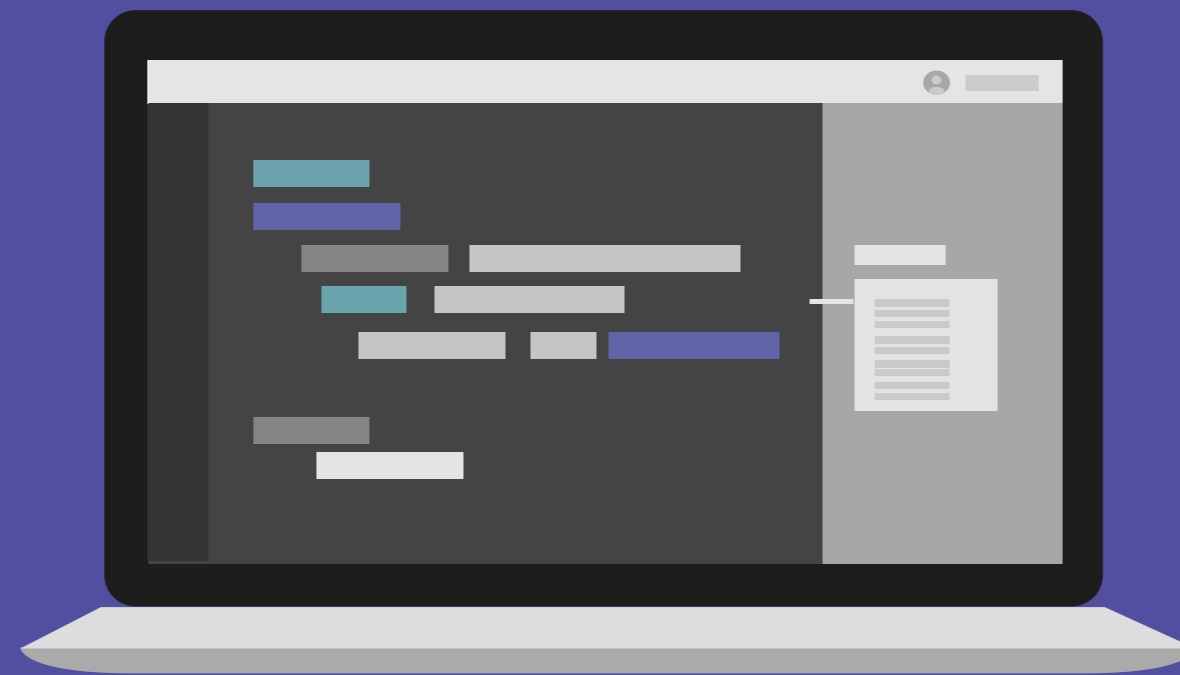
```
# [(id, 이름)]  
accounts = [  
    ("kdhong.elice", "Kildong Hong"),  
    ...  
]
```

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

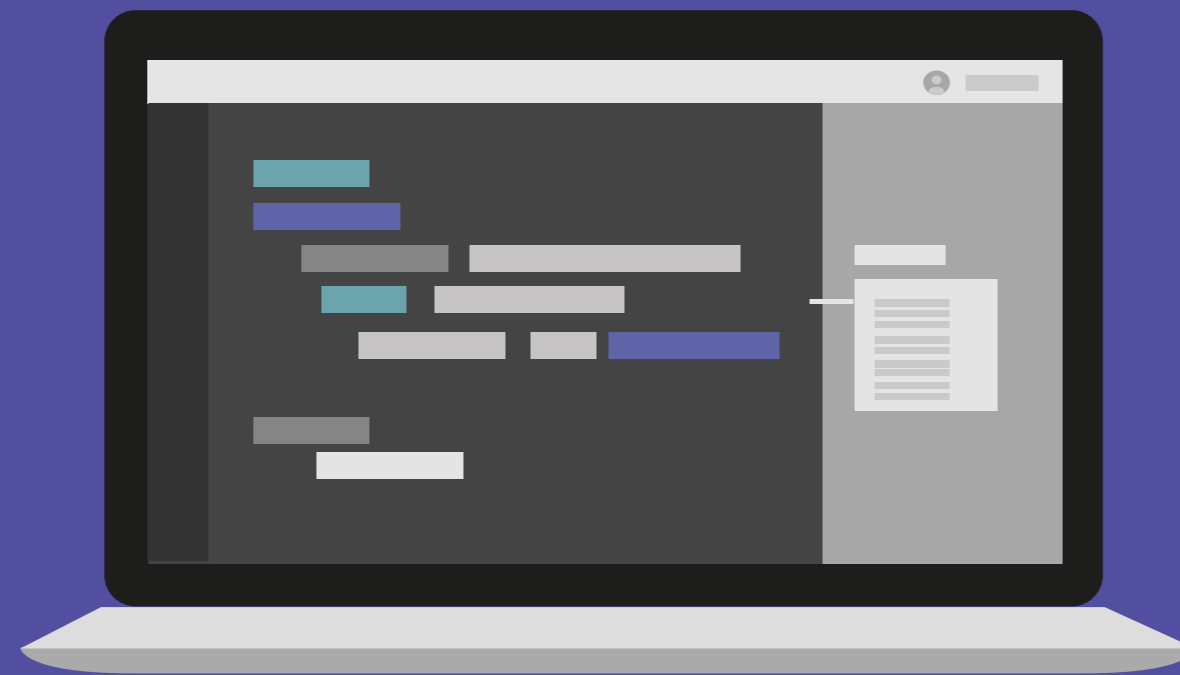
```
# {id: 이름}
accounts = {
    "kdhong": "Kildong Hong",
}

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

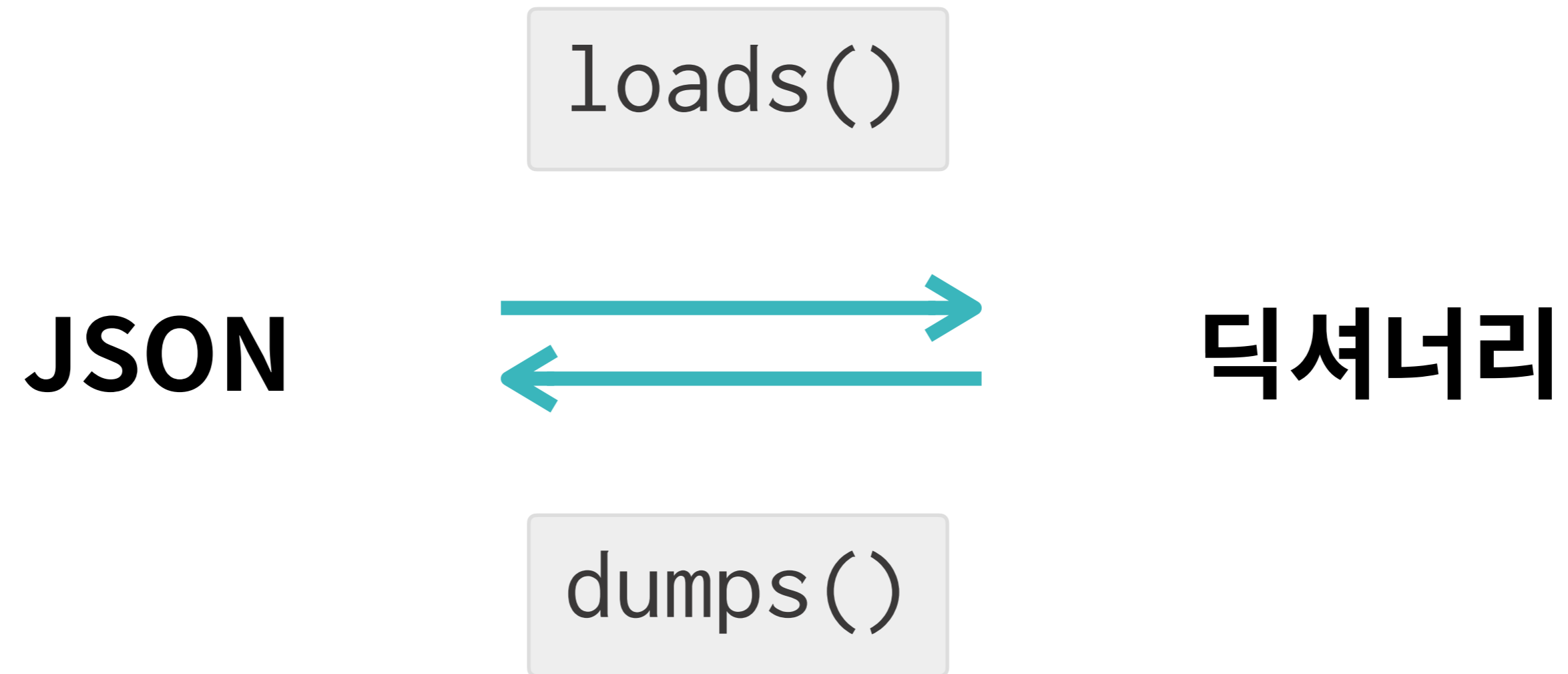
```
{ key: value }
```

"JavaScript Object Notation"

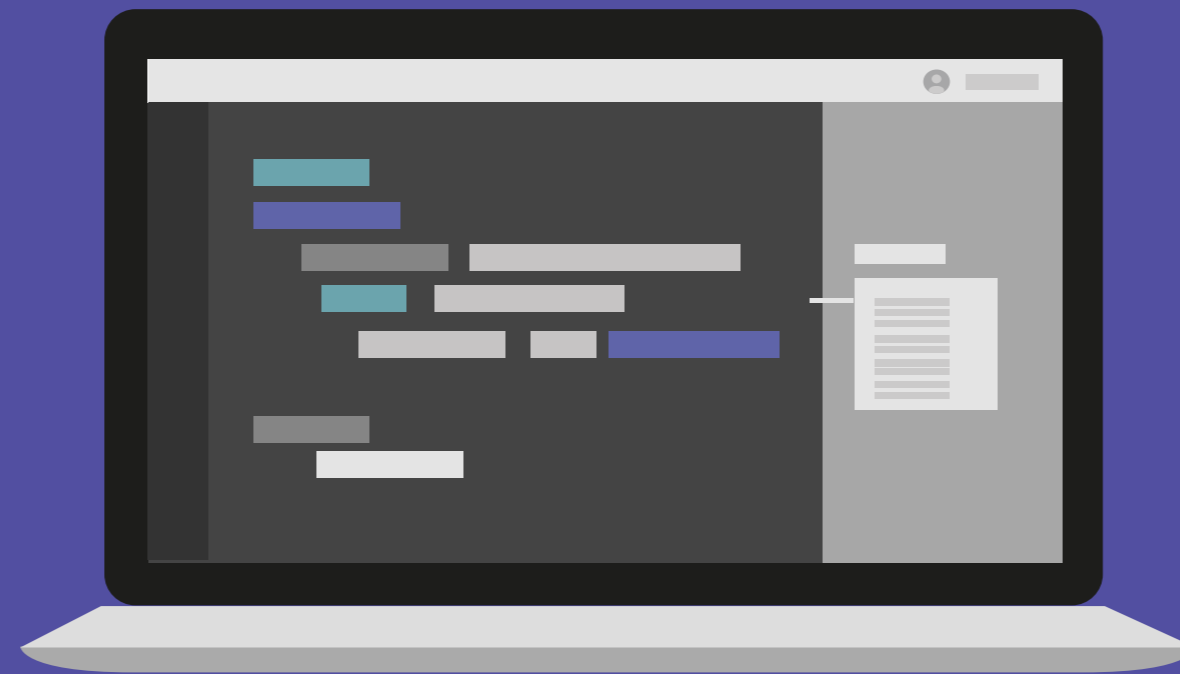
JSON

- **웹 환경**에서 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}
```

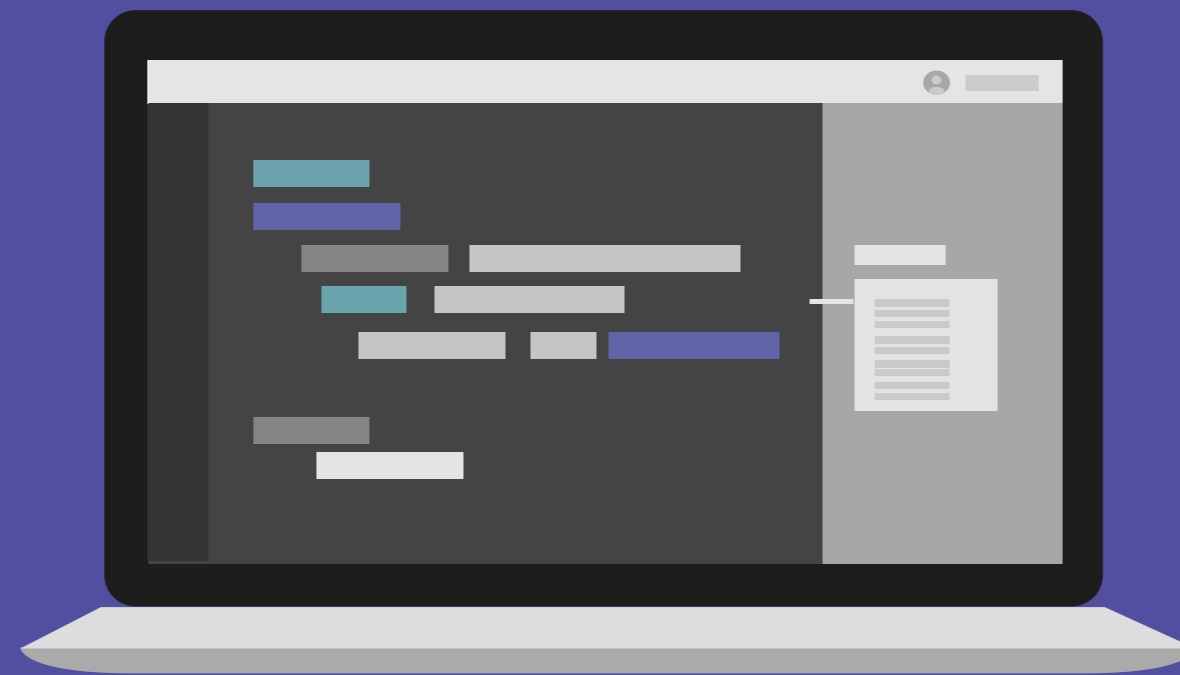

원소 추가/삭제

```
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

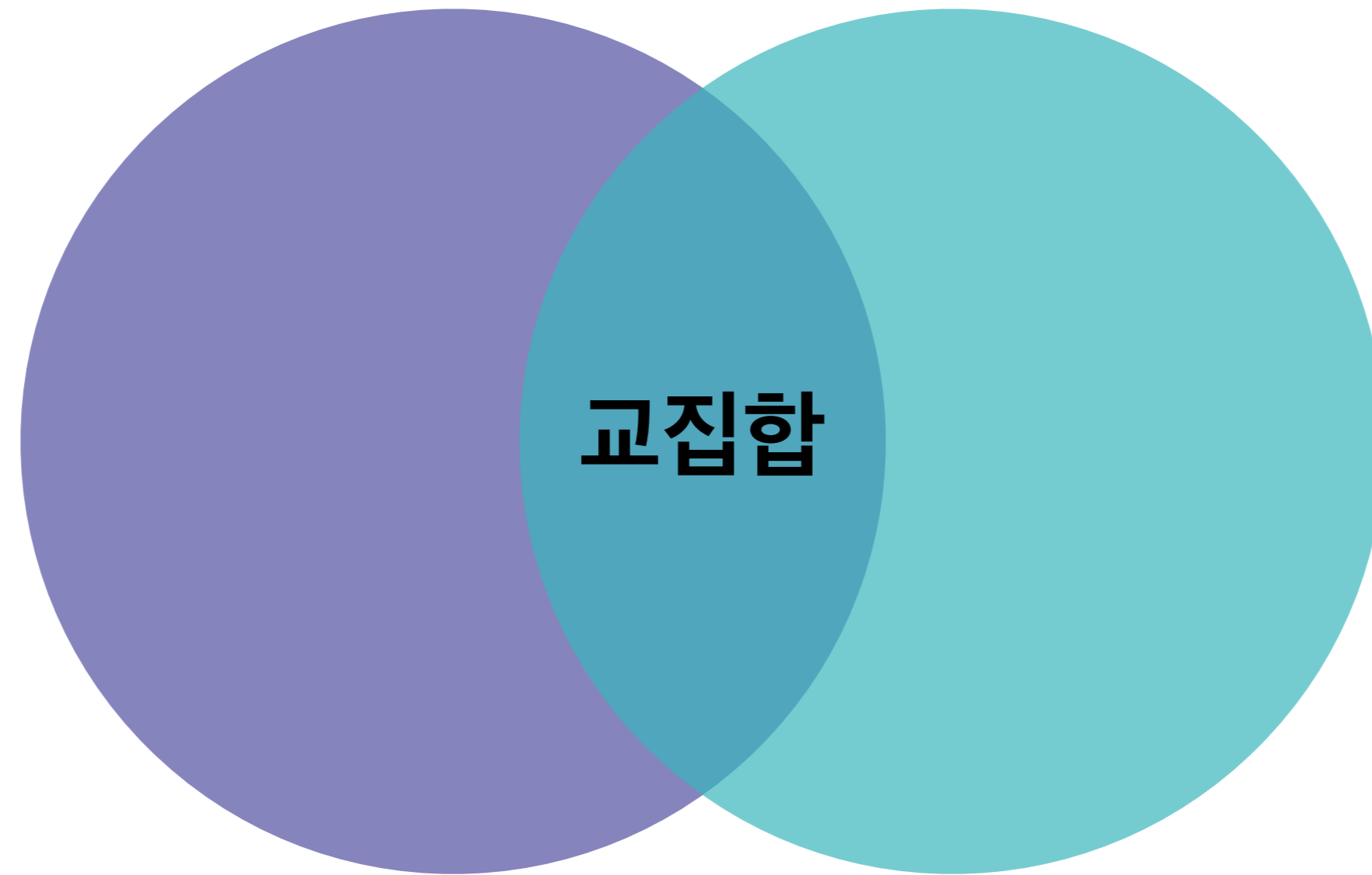
```
num_set = {1, 3, 5, 7}
print(6 in num_set)      # False
print(len(num_set))     # 4
```

[실습 4] set()

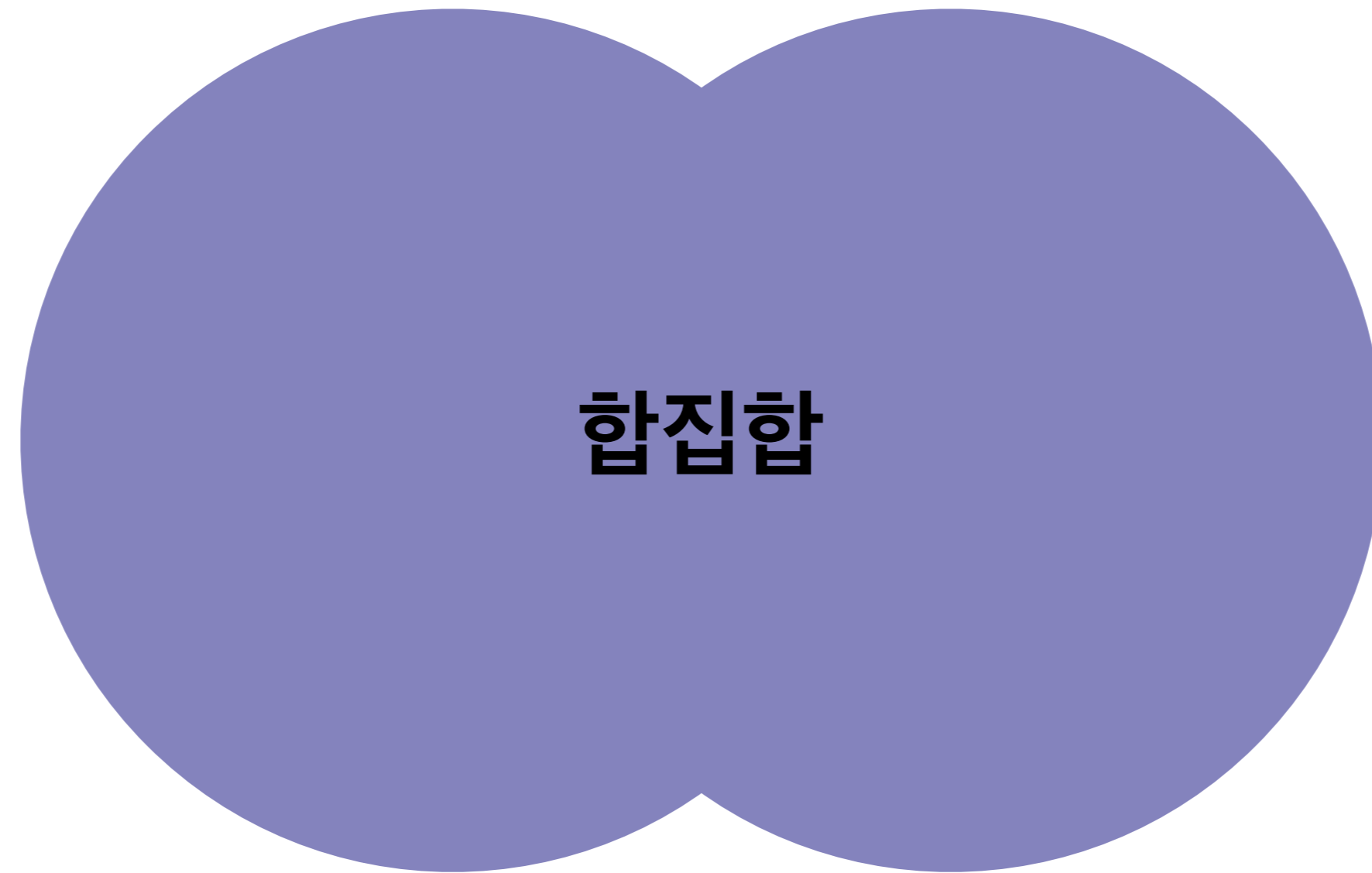


```
/* elice */
```

집합 연산

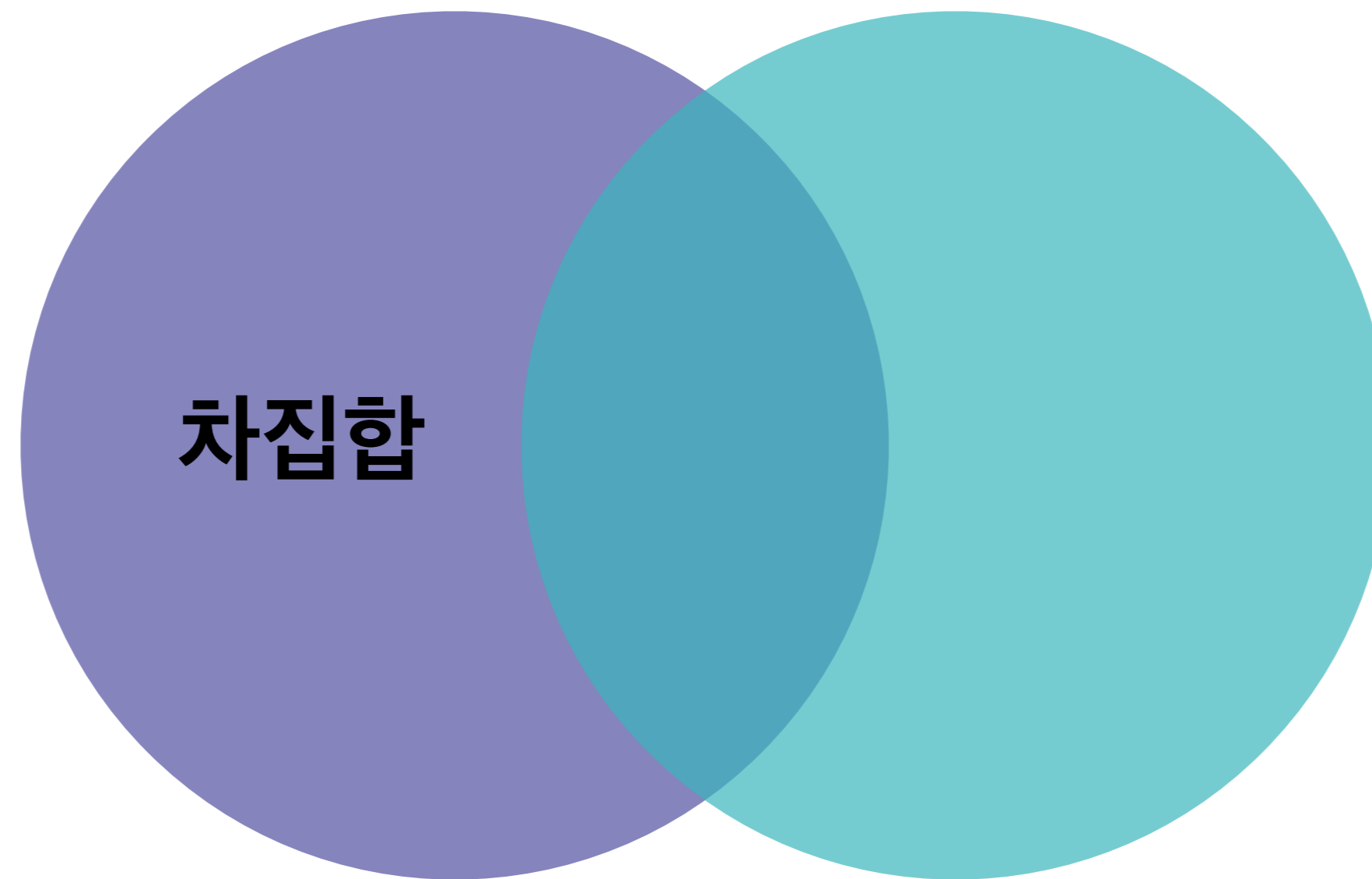


집합 연산

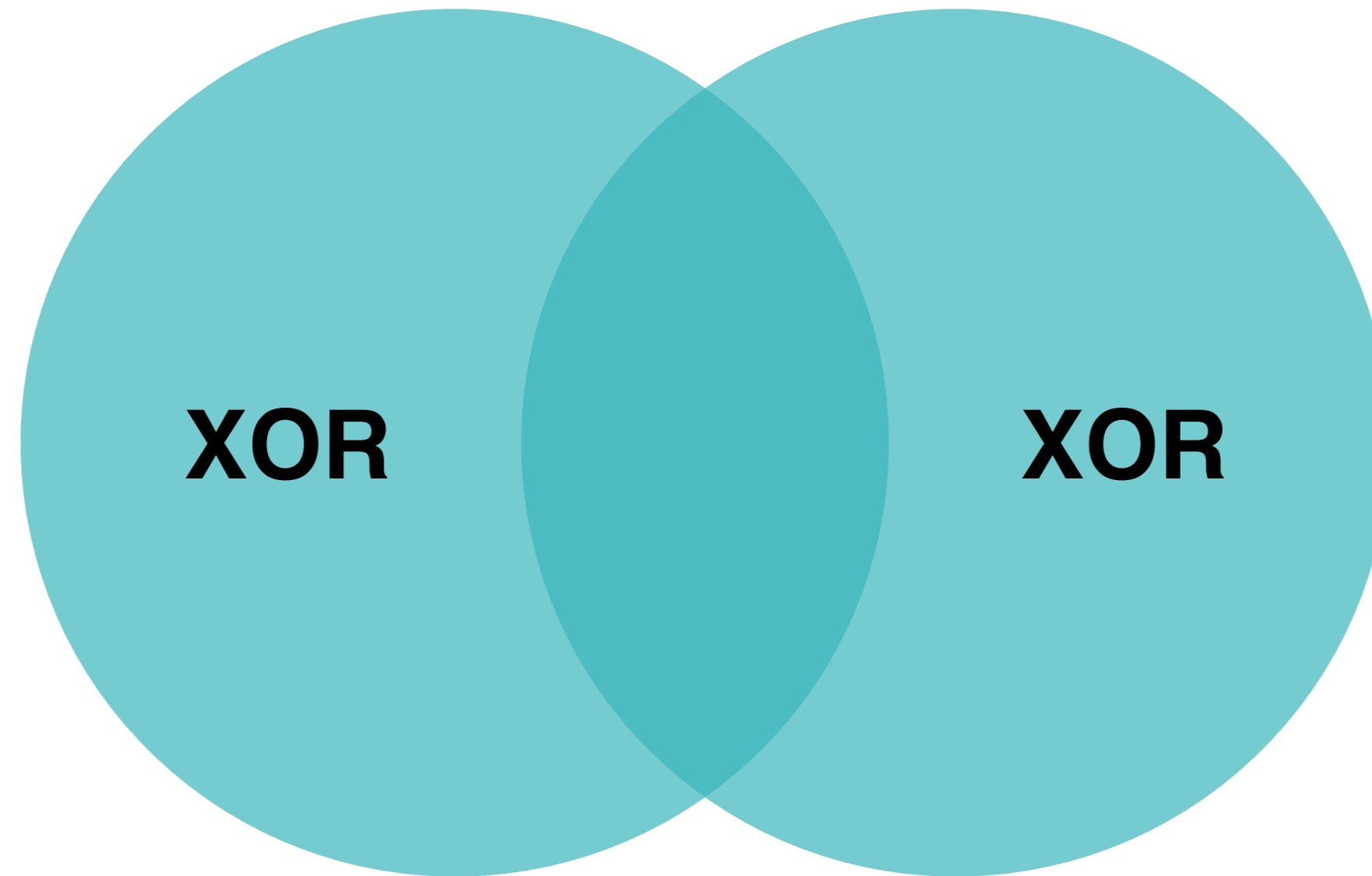


합집합

집합 연산



집합 연산



집합 연산

set1 = {1, 3, 5, 7}

set2 = {1, 3, 9, 27}

집합 연산

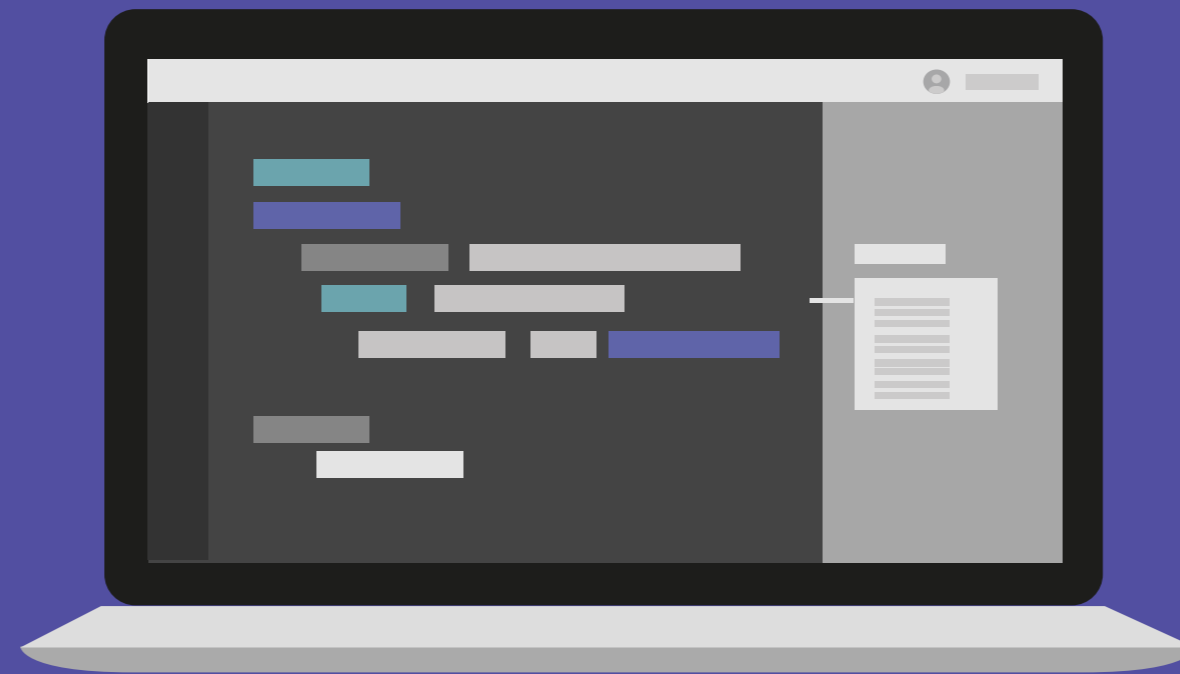
```
union = set1 | set2      # 합집합
```

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

```
diff = set1 - set2      # 차집합
```

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

[실습5] 집합 연산자



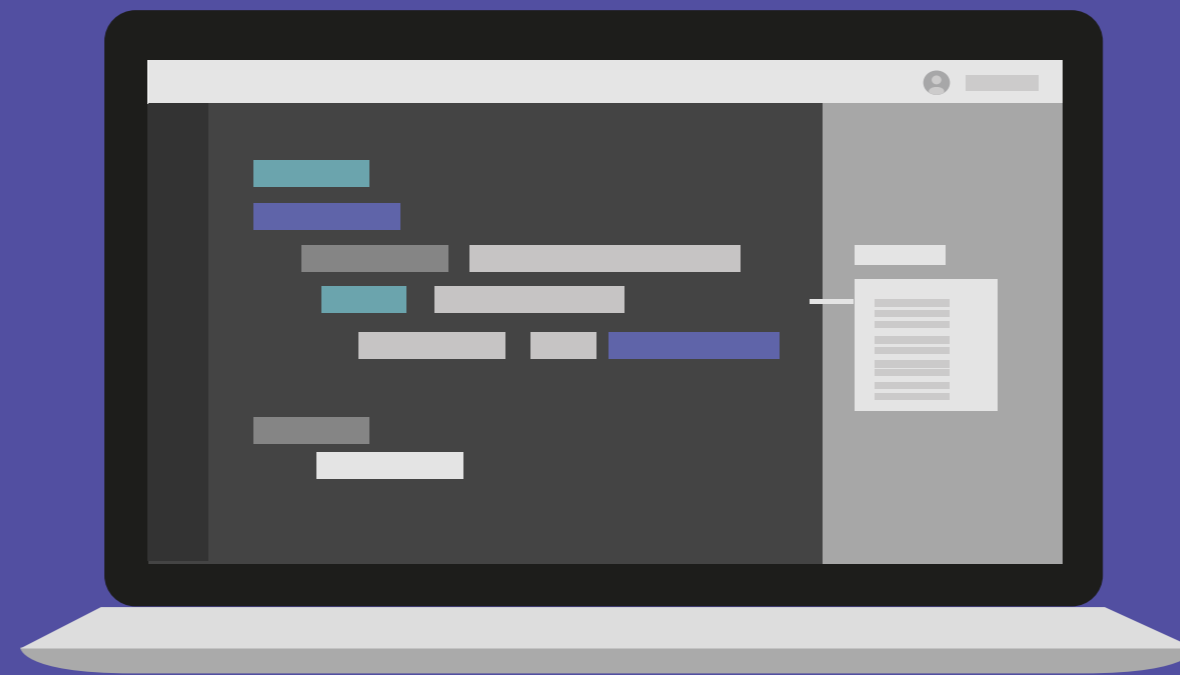
`/* elice */`

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

/* elice */

파이썬 실전 데이터 분석

복잡한 형태의 데이터 다루기

The image displays a complex graphic with three main components:

- Network Diagram:** A diagram at the top showing a network of interconnected nodes, representing data relationships.
- Bar Chart:** A bar chart in the middle with five bars of varying heights, representing data distribution.
- Python Code:** A large block of Python code at the bottom, featuring various symbols, characters, and syntax, likely demonstrating data manipulation techniques.

김건우 선생님

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. CSV
2. 고급 파이썬

CSV

CSV

```
name , age , address , gender
```

- Comma Separated Value
- 각 열이 특정한 의미를 가짐

CSV

```
# movies.csv
```

```
# 국문 제목,영문 제목,개봉 연도
```

```
다크나이트,The Dark Knight,2008
```

```
겨울왕국,Frozen,2013
```

```
슈렉,Shrek,2001
```

```
슈퍼맨,Superman,1978
```

CSV

```
# movies.csv
```

```
# 다른 구분 문자(delimiter)도 사용 가능
```

```
다크나이트|The Dark Knight|2008
```

```
겨울왕국|Frozen|2013
```

```
슈렉|Shrek|2001
```

```
슈퍼맨|Superman|1978
```

CSV

국문 제목	영문 제목	개봉 연도
다크나이트	The Dark Knight	2008
겨울왕국	Frozen	2013
슈렉	Shrek	2001
슈퍼맨	Superman	1978

데이터에 가 포함된 경우

```
# movies.csv
```

```
먹고 기도하고 사랑하라, "Eat, Pray, Love", 2010
```

```
"헬로우, 뉴욕", "Hello, New York", 2013
```

큰따옴표 (" ") 를 이용하여 데이터를 감싼다

CSV의 장점

```
# movies.csv
```

```
아이언맨, Iron Man, 2008
```

```
겨울왕국, Frozen, 2013
```

```
# movies.json
```

```
[{"ko": "아이언맨", "en": "Iron Man", "year": 2008},
```

```
{"ko": "겨울왕국", "en": "Frozen", "year": 2013}]
```

같은 데이터를 저장하는 데 용량을 적게 소모함

CSV의 단점

```
# movies.csv
```

```
아이언맨, Iron, Man, 2008
```

```
겨울왕국, Frozen, 2013
```

데이터 오염에 취약함

CSV

국문 제목	영문 제목	개봉 연도	???
아이언맨	Iron	Man	2008
겨울왕국	Frozen	2013	

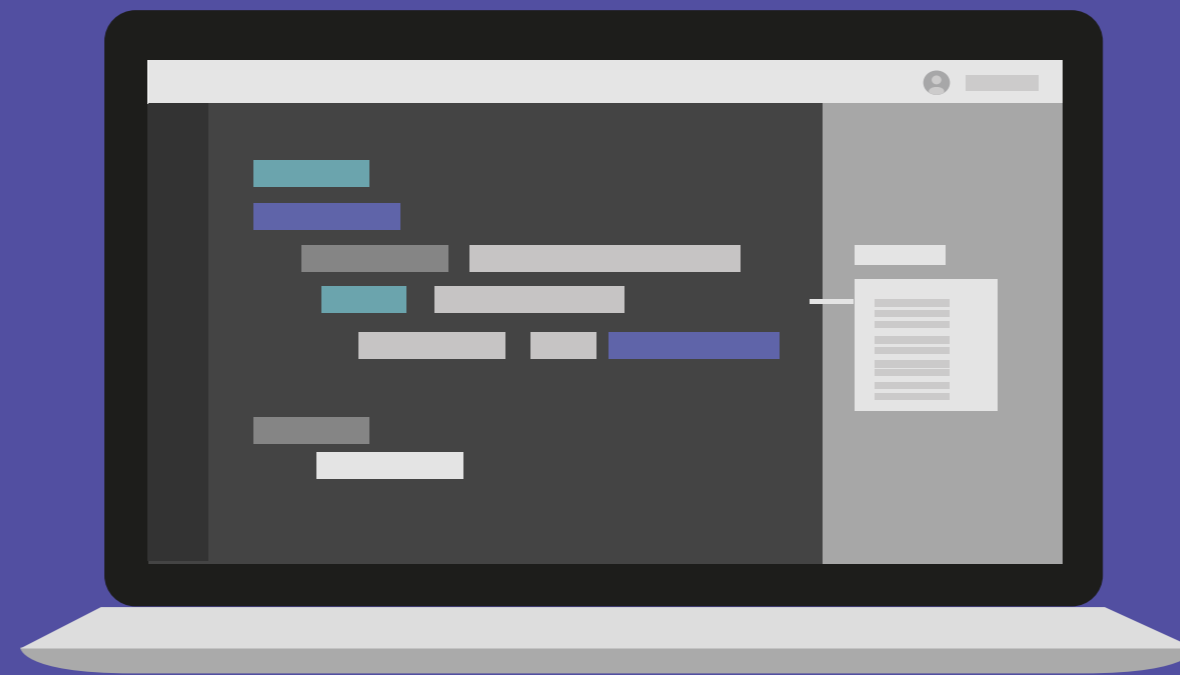
데이터 오염에 취약함

CSV

```
import csv

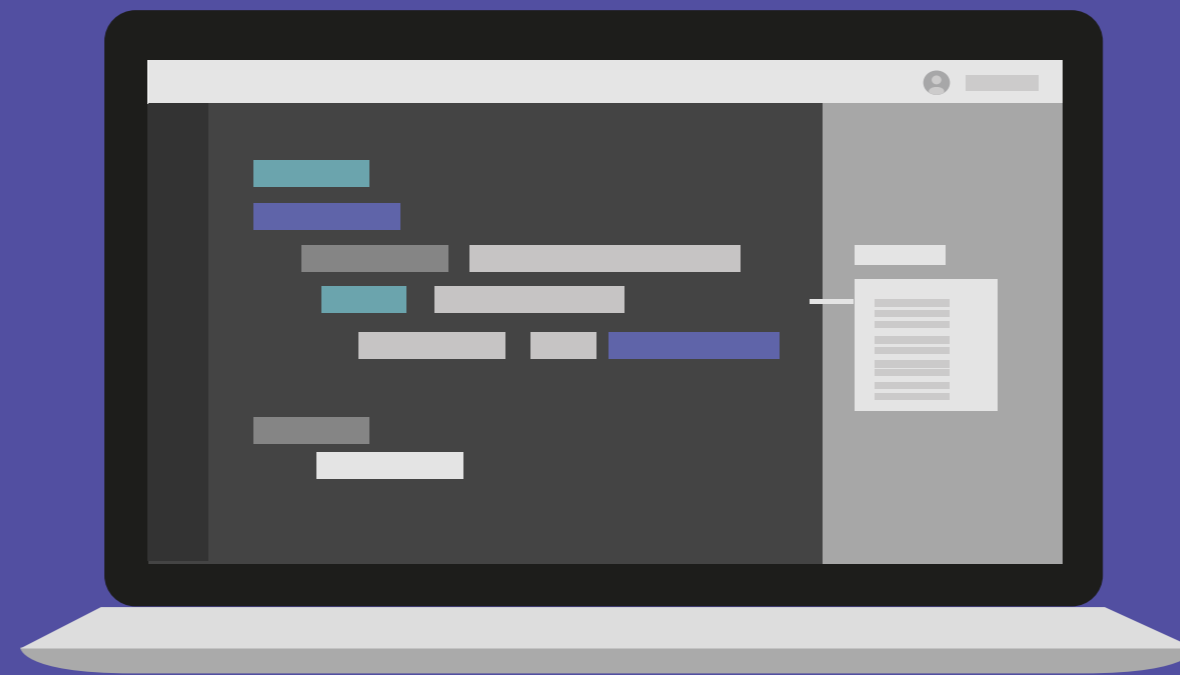
with open('movies.csv') as file:
    reader = csv.reader(file, delimiter=',')
    for row in reader:
        print(row[0])
```

[실습1] CSV 데이터 읽기



```
/* elice */
```

[실습2] CSV 데이터 변환하기



`/* elice */`

고급 파이썬

lambda

```
def square(x):  
    return x * x
```

```
square = lambda x: x * x
```

lambda

```
movies = [  
    "다크나이트, The Dark Knight, 2008",  
    "겨울왕국, Frozen, 2013",  
    "슈렉, Shrek, 2001",  
    "슈퍼맨, Superman, 1978"  
]
```


lambda

```
def get_eng_title(row):  
    split = row.split(',')  
    return split[1]  
  
sorted(movies, key=get_eng_title)
```

lambda

```
get_eng_title = lambda row: row.split(',')[1]  
sorted(movies, key=get_eng_title)
```

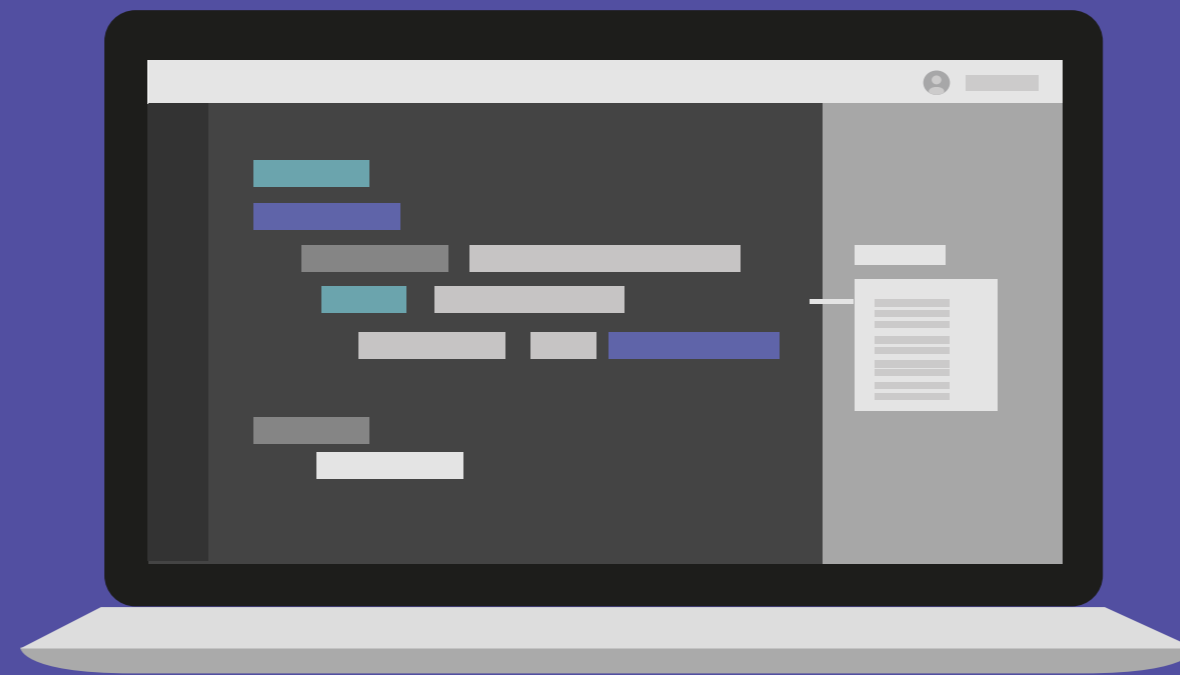
lambda

```
sorted(movies,  
       key=lambda row: row.split(',')[1])
```

assert()

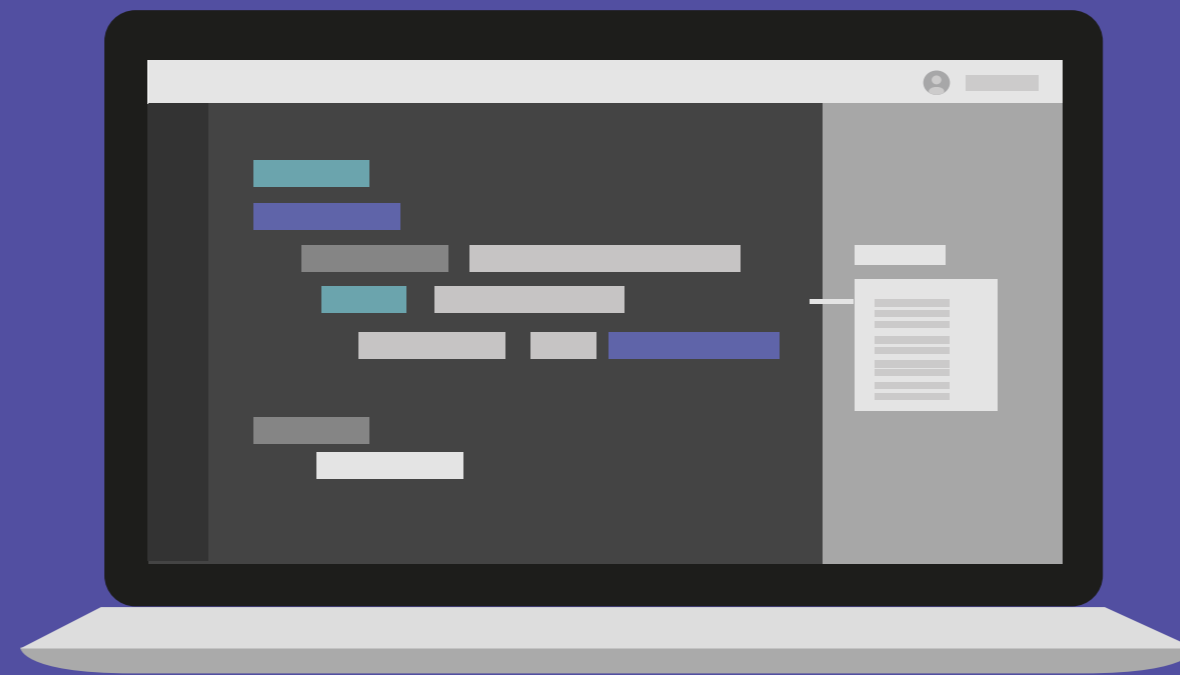
```
def square1(x):  
    return x * x  
  
square2 = lambda x: x * x  
  
# 두 값이 같으면 통과, 아니면 에러  
assert(square1(3) == square2(3))
```

[실습3] 한 줄 함수 작성하기



```
/* elice */
```

[실습4] 함수를 리턴하는 함수



```
/* elice */
```

```
map()
```

```
movies = [  
    "다크나이트, The Dark Knight, 2008",  
    "겨울왕국, Frozen, 2013",  
    "슈렉, Shrek, 2001",  
    "슈퍼맨, Superman, 1978"  
]
```

map()

```
eng_titles = [  
    "The Dark Knight",  
    "Frozen",  
    "Shrek",  
    "Superman"  
]
```


map()

```
def get_eng_title(row):  
    split = row.split(',')  
    return split[1]
```

```
eng_titles = \  
    [get_eng_title(row) for row in movies]
```

map()

```
def get_eng_title(row):  
    split = row.split(',')  
    return split[1]  
  
eng_titles = map(get_eng_title, movies)
```

map()

```
eng_titles = map(  
    lambda row: row.split(',')[1],  
    movies  
)
```

map()

```
[get_eng_title(row) for row in movies]
```

```
[row.split(',')[1] for row in movies]
```

```
map(get_eng_title, movies)
```

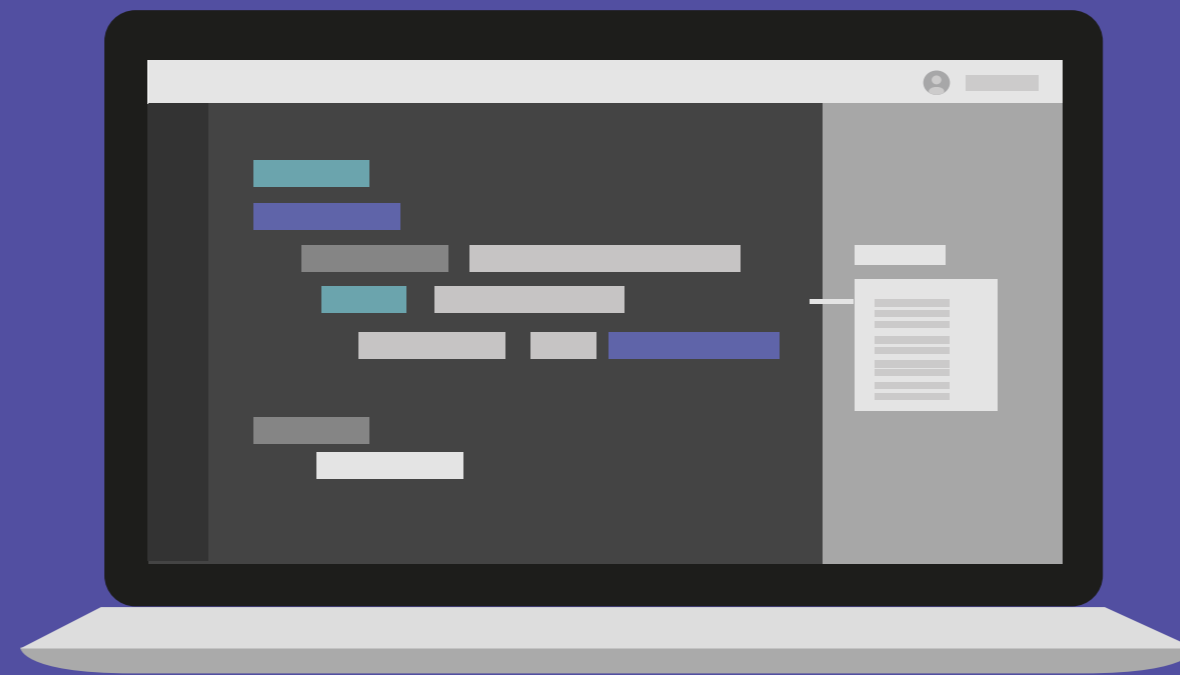
```
map(lambda row: row.split(',')[1],  
     movies)
```

map()

```
eng_titles = map(get_eng_title, movies)
print(eng_titles)
# <map object at 0x104154f98>
```

리스트가 아닌 map이라는 타입을 가짐

[실습5] map()



```
/* elice */
```

filter()

```
words = ['real', 'man', 'rhythm', ...]
```

```
r_words = ['real', 'rhythm', 'right', ...]
```

filter()

```
words = ['real', 'man', 'rhythm', ...]  
r_words = [word for word in words if  
word.startswith('r')]
```


filter()

```
def starts_with_r(word):  
    return word.startswith('r')  
  
words = ['real', 'man', 'rhythm', ...]  
r_words = filter(starts_with_r, words)
```

filter()

```
starts_with_r = lambda w: w.startswith('r')
```

```
words = ['real', 'man', 'rhythm', ...]
```

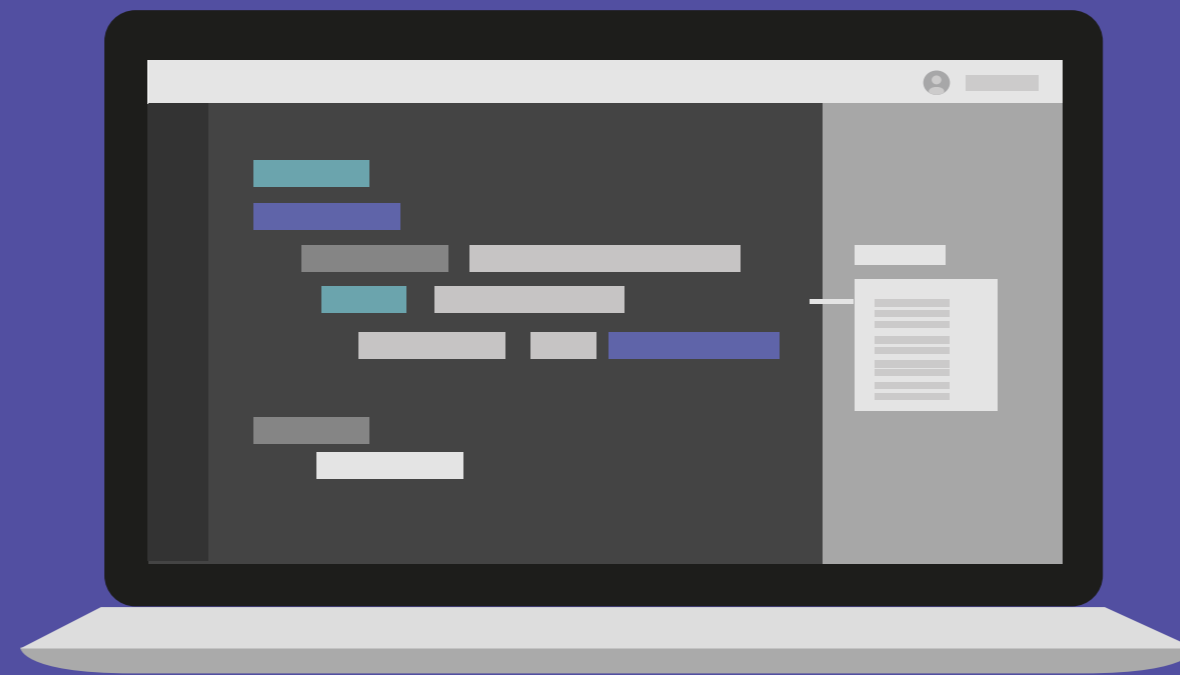
```
r_words = filter(starts_with_r, words)
```

filter()

```
r_words = filter(starts_with_r, words)
print(r_words)
# <filter object at 0x104154f98>
```

리스트가 아닌 filter 타입을 가짐

[실습6] filter()



```
/* elice */
```

`/* elice */`

문의 및 연락처

academy.elice.io

contact@elice.io

facebook.com/elice.io

medium.com/elice