

/* elice */

데이터 분석 기초

김건우 선생님 · 일 14:00



4주차 수업

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

01 CSV

`/* elice */`

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

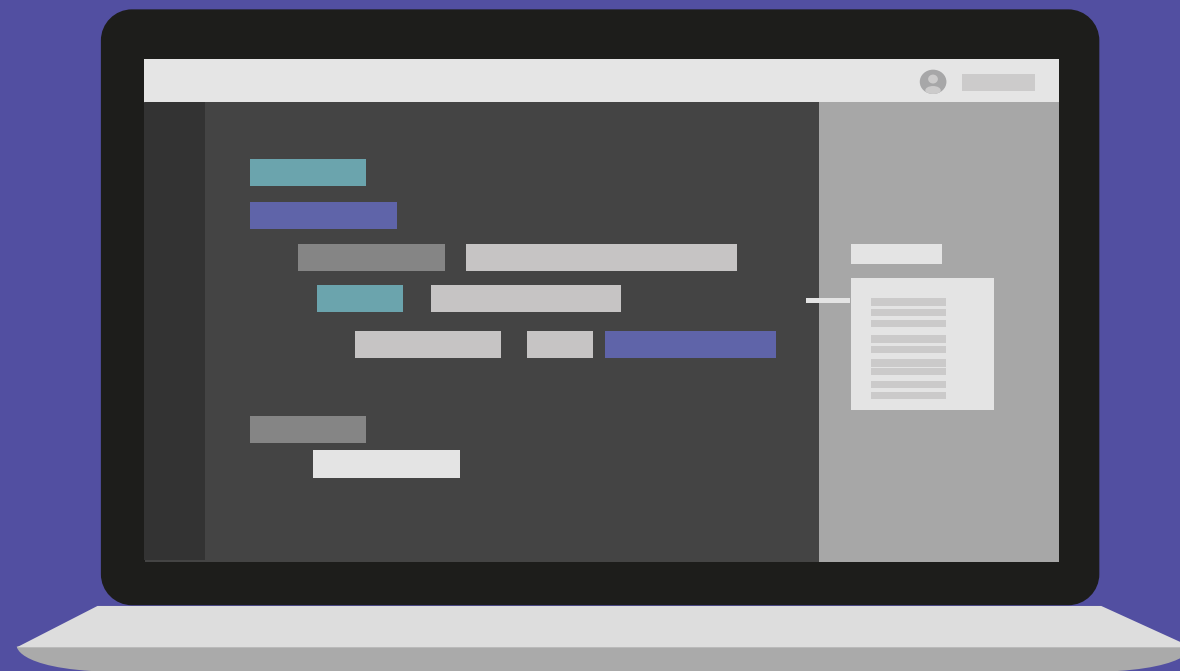
with open('movies.csv') as file:
    reader = csv.reader(file, delimiter=',')
    for row in reader:
        print(row[0])
```

실습 1: CSV 데이터 읽기



```
/* elice */
```

실습 2: CSV 데이터 변환하기



```
/* elice */
```

02 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)
```

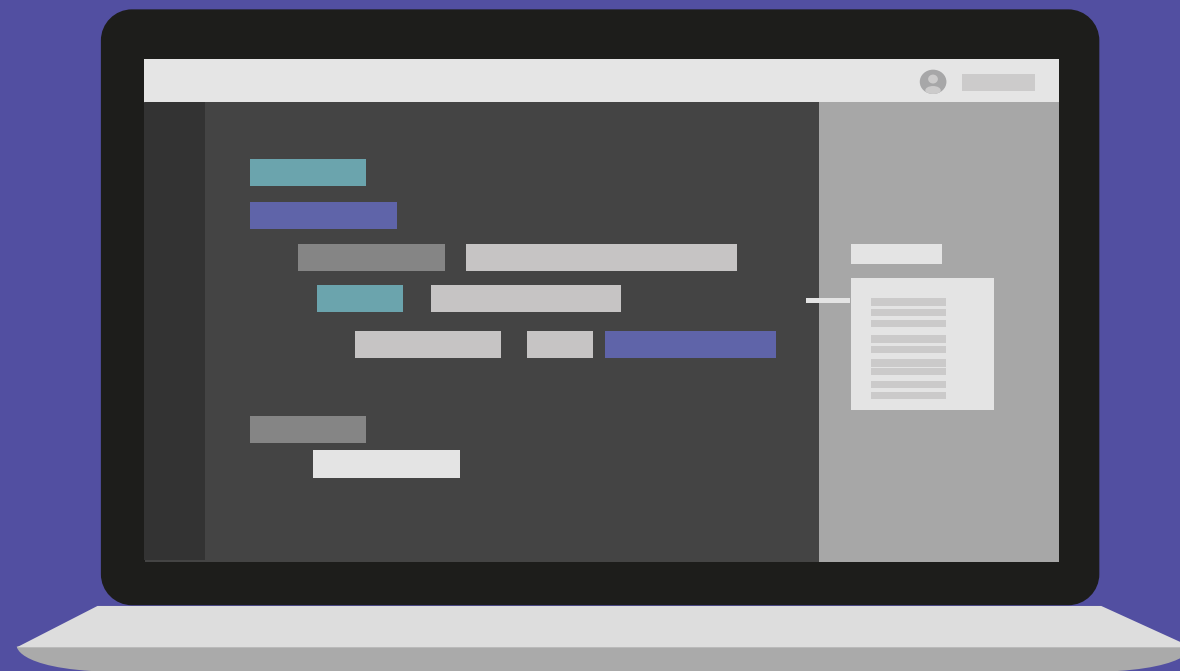
lambda

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

assert()

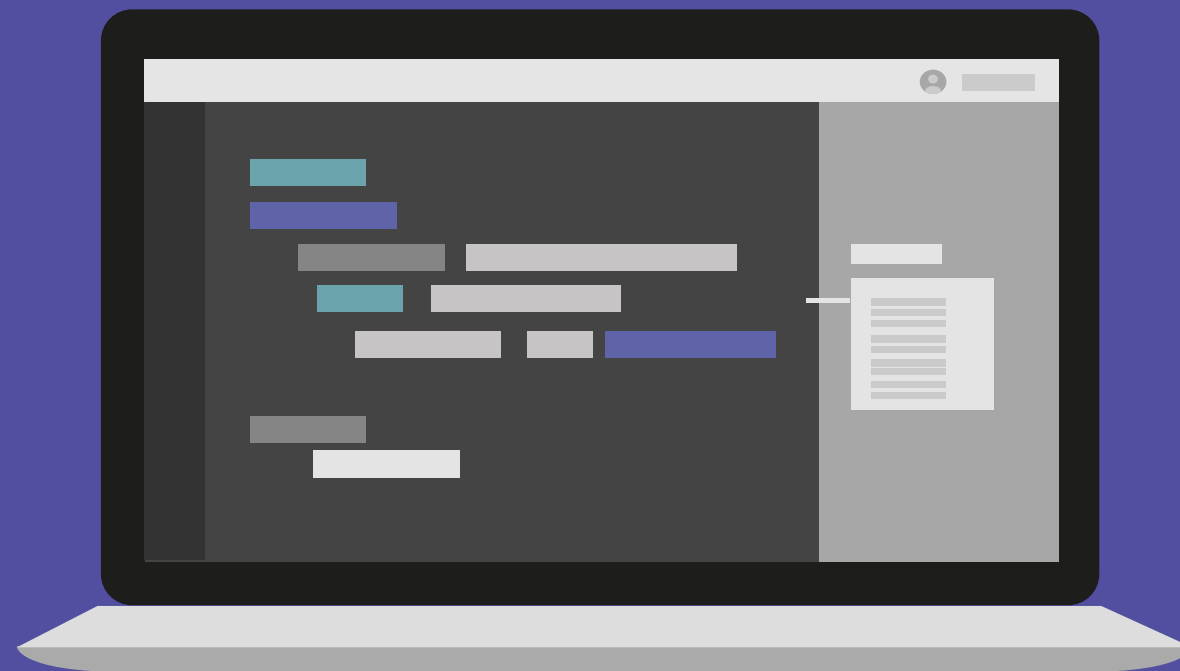
```
def square1(x):  
    return x * x  
  
square2 = lambda x: x * x  
  
# 두 값이 같으면 통과, 아니면 에러  
assert(square1(3) == square2(3))
```

실습 3: lambda 함수



```
/* elice */
```

실습 4: 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)
```

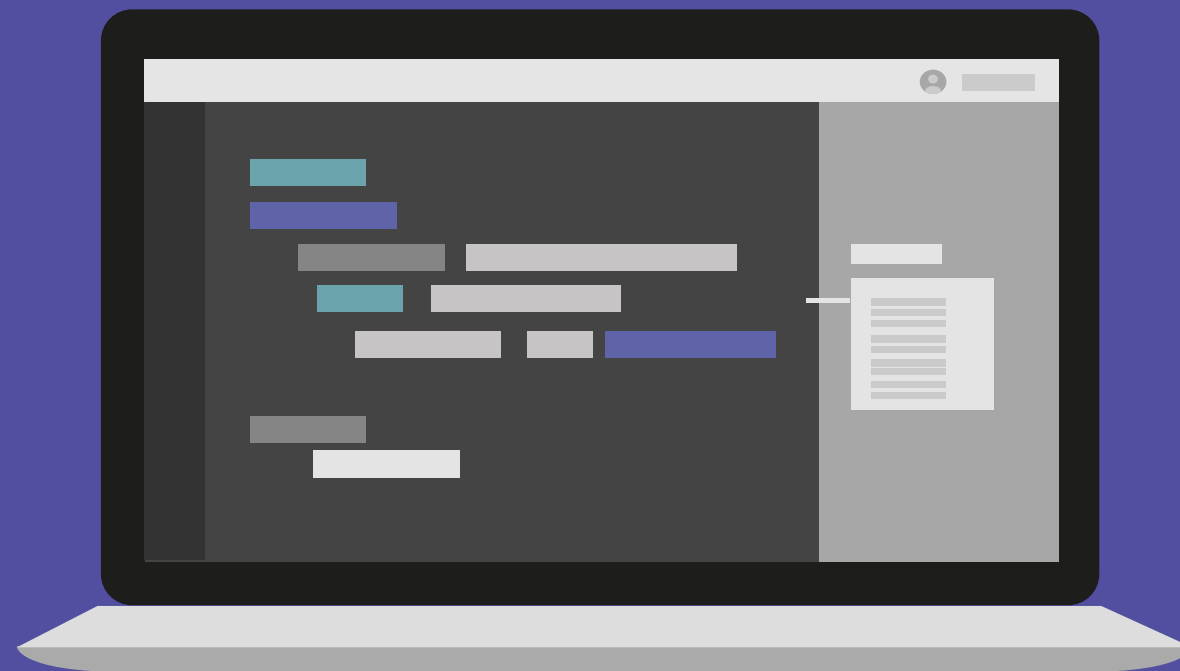
```
map(lambda row: row.split(',')[1],  
     movies)
```

map()

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

리스트가 아닌 map이라는 타입을 가짐

실습 5: 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

실습 6: filter



```
/* elice */
```

`/* elice */`

문의 및 연락처

academy.elice.io

contact@elice.io

facebook.com/elice.io

medium.com/elice