

## 4-8. 여러가지 파일 포맷 다루기 (1) CSV 파일

### CSV

**CSV**는 Comma Separated Value의 약자로, 쉼표로 구분된 파일을 말합니다.

각각의 칼럼(column)을 쉼표( , )로 구분하는데요. 간단한 csv 파일을 만들어 보겠습니다.

아래 데이터는 빌보드 차트 1위 ~ 5위까지의 곡들입니다. 각 키(key)는 순위를, 값(value)은 곡명, 가수, 그리고 발매일에 대한 정보를 가지고 있네요. 코드 실행을 통해 **billboardchart.csv** 파일을 생성해 봅시다.

#### [Input]

```
billboardchart = {  
    1 : ["Tho Box", "Roddy Ricch", "2019-12-19"],  
    2 : ["Don't Start Now", "Dua Lipa", "2019-11-01"],  
    3 : ["Life Is Good", "Future Featuring Drake", "2020-02-10"],  
    4 : ["Blinding", "The Weeknd", "2019-11-29"],  
    5 : ["Circles", "Post Malone", "2019-08-30"]}  
  
with open("billboardchart.csv", "w") as f:  
    for i in billboardchart.values():  
        data = ",".join(i)  
        print(i)  
        print(data)  
        f.write(data + "\n")
```



실행 ▶

#### [Output]

csv 파일은 주피터 실행 경로에 저장되어 있을 것입니다. 파일을 검색 해 열어 보시면 각각의 데이터가 쉼표 (,) 로 구분돼 있는 것을 확인하실 수 있을 거예요. 각 칼럼이 무엇을 의미하는지 추가하면 좋을 것 같은데요. 첫 번째 줄에 **title, singer, released date** 를 헤더로 추가해 본 후 다시 확인해 봅시다.

#### [Input]



```
import csv

header = ["title", "singer", "released date"]

with open("billboardchart.csv", "r") as inputfile:
    with open("billboardchart_out.csv", "w", newline='\n') as outputfile:
        fi = csv.reader(inputfile, delimiter=',')
        fo = csv.writer(outputfile, delimiter=',')
        fo.writerow(header)
        for row in fi:
            fo.writerow(row)
```

실행 ▶

#### [Output]

## CSV 파일과 Pandas

판다스(pandas)의 DataFrame은 **to\_csv** 메소드를 지원합니다. 이 메소드를 이용하면 csv 파일로 쉽게 저장할 수 있어요. 데이터를 준비한 뒤 판다스를 활용 해 csv 파일로 저장해 보겠습니다.

#### [Input]



```
#- 1. 데이터를 준비합니다.
```

```
fields = ["title", "singer", "released date"]
rows = [ ["Tho Box", "Roddy Ricch", "2019-12-19"],
          ["Don't Start Now", "Dua Lipa", "2019-11-01"],
          ["Life Is Good", "Future Featuring Drake", "2020-02-10"],
          ["Blinding", "The Weeknd", "2019-11-29"],
          ["Circles", "Post Malone", "2019-08-30"]]
```

실행 ▶

[Output]

[Input]

#- 2. 판다스를 이용해 데이터를 csv 파일로 저장합니다.

```
import pandas as pd
```

```
df=pd.DataFrame(rows, columns=fields)
df.to_csv('pandas.csv',index=False)
```

실행 ▶

[Output]

[Input]

#- 3. 동일한 내용을 csv.writer를 이용해 수행해 봅니다.

```
import csv
```

```
filename = "test.csv"
with open(filename, 'w+', newline='\n') as csv_file:
    csv_writer = csv.writer(csv_file)
```

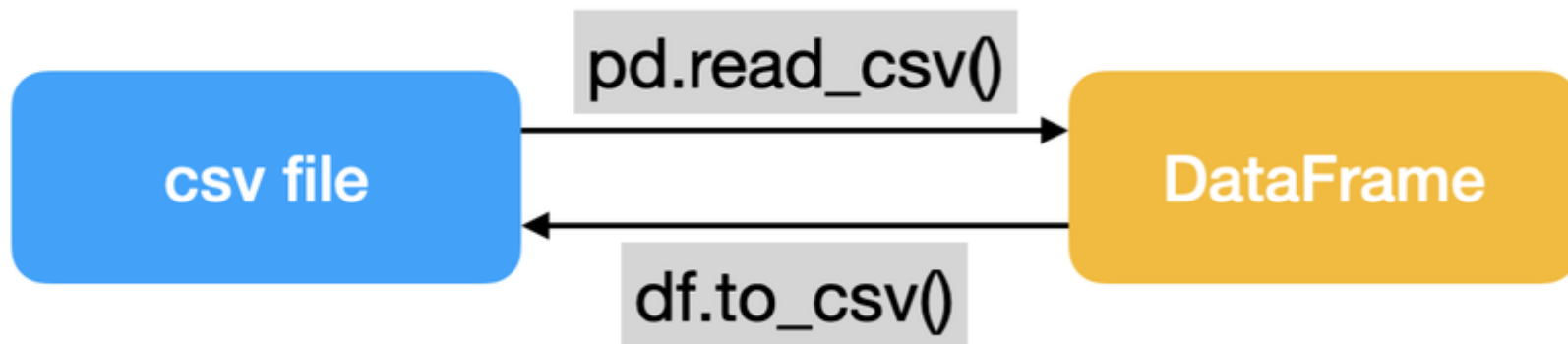
```
csv_writer.writerow(fields)
csv_writer.writerows(rows)
```

#- test.csv 파일을 직접 열어서 눈으로 살펴 보세요. -#

실행 ▶

[Output]

반대로, csv 파일을 DataFrame으로 변환시키면 데이터 분석 등 사용자가 편집하기에 용이하겠죠? 마찬가지로 매우 간단합니다.



[Input]

```
df = pd.read_csv('pandas.csv')
df.head()
```



실행 ▶

[Output]