

[5-1. CSV 파일 읽고 쓰기]

1. DataFrame을 CSV 파일로 쓰기

Pandas의 DataFrame의 'to_csv' 메서드를 사용하면 DataFrame을 바로 csv파일로 변환할 수 있습니다.

```
In [ ]: # 예시 데이터프레임 작성
import pandas as pd
from pandas import DataFrame

data = {
    'Name': ['John', 'Anna', 'Peter', 'Linda'],
    'Age': [28, 34, 29, 32],
    'City': ['New York', 'Paris', 'Berlin', 'London']
}

df = DataFrame(data)
print(df)
```

	Name	Age	City
0	John	28	New York
1	Anna	34	Paris
2	Peter	29	Berlin
3	Linda	32	London

```
In [ ]: # csv 파일로 저장하기
df.to_csv('csv_data3.csv', index=False)
```

- index=False는 DataFrame의 인덱스를 파일에 포함하지 않도록 설정합니다. 인덱스가 필요한 경우 이 옵션을 생략하거나 index=True로 설정할 수 있습니다.

2. CSV 파일 읽기

pd.read_csv 메서드를 사용하여 저장된 csv 파일을 다시 불러와 DataFrame으로 변환할 수 있습니다.

```
In [ ]: # CSV 파일을 DataFrame으로 불러오기
loaded_df = pd.read_csv('csv_data3.csv')

print(loaded_df)
```

	Name	Age	City
0	John	28	New York
1	Anna	34	Paris
2	Peter	29	Berlin
3	Linda	32	London

3. csv 라이브러리를 이용하여 csv 파일 읽고, DataFrame으로 설정하기

csv 라이브러리의 DictReader 메서드를 사용하면 데이터의 첫번째 행을 딕셔너리의 키값으로 설정한 후 이후 전체 데이터를 딕셔너리 형태로 읽어들이게 됩니다. 딕셔너리 형태로 읽어들이는 결과 값은 그대로 DataFrame으로 변환할 수 있습니다.

```
In [ ]: import csv

with open('csv_data3.csv', 'r', newline='') as file:
    # CSV 딕셔너리리더 객체 생성
    reader = csv.DictReader(file)
    data = [row for row in reader]

df = DataFrame(data)
print(df)
```

	Name	Age	City
0	John	28	New York
1	Anna	34	Paris
2	Peter	29	Berlin
3	Linda	32	London

4. csv 라이브러리를 이용하여 DataFrame을 csv 파일로 쓰기

csv 라이브러리의 writer 메서드를 이용하여 writer 객체를 생성한 후 writer 객체를 이용하여 파일에 데이터를 작성합니다.

DataFrame의 iterrows 메서드를 사용하면 데이터를 한 행씩 꺼내서 사용할 수 있습니다.

```
In [ ]: with open('csv_data4.csv', 'w', newline='') as file:
    # CSV writer 객체 생성
    writer = csv.writer(file)

    # 컬럼명 쓰기
    writer.writerow(df.columns)

    # 데이터 쓰기
    for index, row in df.iterrows():
        writer.writerow(row)
```

```
In [ ]: # 파일 출력이 제대로 되었는지 확인  
pd.read_csv('csv_data4.csv')
```

```
Out[ ]:
```

	Name	Age	City
0	John	28	New York
1	Anna	34	Paris
2	Peter	29	Berlin
3	Linda	32	London