DataFrame

#01. DataFrame의 이해

행과 열로 구성된 데이터 구조

데이터 분석에서 가장 일반적으로 사용되는 형태로 엑셀의 시트 구조를 생각하면 이해하기 쉽다.

DataFrame으로 변환 가능한 데이터 형태

1. 2차원 리스트
2. 리스트를 원소로 갖는 딕셔너리
3. 딕셔너리를 원소로 갖는 리스트
4. csv 파일로부터 가져오기 (쉼표로 구분)
5. 엑셀 파일로부터 가져오기
6. 데이터베이스로부터 가져오기 (SQL)
7. 웹 데이터 수집 (OpenAPI 연동, 웹 크롤링)

> 데이터베이스로부터 가져오기는 다음 예제에서 진행, 크롤링은 이전 과목에서 이미 진행함

#02. 패키지 참조하기

`Pandas` 패키지가 설치되어 있어야 한다.

```
from pandas import DataFrame
```

#03. DataFrame 생성하기

1) 2차원 리스트

DataFrame 생성하기

```
# 어느 학급의 성적표를 표현한 2차원 리스트
grade_data = [
                [1, "남자", 98,  77,  88,  64],
                [2, "여자", 88,  90,  62,  72],
                [1, "남자", 92,  70,  83,  79],
                [3, "여자", 63,  60,  31,  70],
                [4, "남자", 75,  50,  90,  88]
            ]

# 2차원 리스트를 데이터프레임으로 변환
# -> 학생별 점수 표현
df = DataFrame(grade_data)
df
```

인덱스와 컬럼 이름 지정하면서 데이터 프레임 생성

```
# 인덱스 이름으로 사용할 리스트
i_names = ['철수', '영희', '민철', '수현', '호영']

# 컬럼 이름으로 사용할 리스트
c_names = ['학년', '성별', '국어', '영어', '수학', '과학']

# 인덱스와 컬럼이름 지정하기
df = DataFrame(grade_data, index=i_names, columns=c_names)
df
```

2) 리스트를 원소로 갖는 딕셔너리

```
# 딕셔너리를 통한 데이터 프레임 만들기
# 어느 학급의 성적표를 표현한 딕셔너리
grade_dic = {
    '학년': [1, 2, 1, 3, 4],
    '성별': ['남자', '여자', '남자', '여자', '남자'],
    '국어': [98, 88, 92, 63, 75],
    '영어': [77, 90, 70, 60, 50],
    '수학': [88, 62, 83, 31, 90],
    '과학': [64, 72, 79, 70, 88]
}

# → 딕셔너리의 key는 DataFrame의 컬럼(열)이름이 된다.
df = DataFrame(grade_dic)
df
```

인덱스 이름을 지정하면서 데이터 프레임 생성

```
# 인덱스 지정하기
df = DataFrame(grade_dic, index=['철수', '영희', '민철', '수현', '호영'])
# 인덱스 제목 지정하기
df.index.name = '이름'
df
```

3) 딕셔너리를 원소로 갖는 리스트로부터 생성

```
grade_list = [
    {"학년": 1, "성별": "남자", "국어": 98, "영어": 77, "수학": 88, "과학": 64},
    {"학년": 2, "성별": "여자", "국어": 88, "영어": 90, "수학": 62, "과학": 72},
    {"학년": 1, "성별": "남자", "국어": 92, "영어": 70, "수학": 83, "과학": 79},
    {"학년": 3, "성별": "여자", "국어": 63, "영어": 60, "수학": 31, "과학": 70},
    {"학년": 4, "성별": "남자", "국어": 120, "영어": 50, "수학": 90, "과학": 88}
]

df = DataFrame(grade_list)
df
```

인덱스 이름을 지정하면서 데이터 프레임 생성

```
# 인덱스 이름 지정하기
df = DataFrame(grade_list, index=['철수', '영희', '민철', '수현', '호영'])
# 인덱스 제목 지정하기
df.index.name = '이름'
df
```

4) csv 파일로부터 가져오기 (쉼표로 구분)

Pandas로부터 `read_csv` 함수 참조

```
from pandas import read_csv
```

csv 파일을 데이터 프레임으로 가져오기

로컬에 위치한 파일을 상대,절대 경로 방식으로 가져올 수 있으며 온라인상의 파일을 URL을 기반으로 가져올 수 있다.

```
df = read_csv("https://data.hossam.kr/grade.csv", encoding="euc-kr")
df
```

로딩이 완료된 데이터프레임의 특정 열을 인덱스로 변경

원본에는 변화가 없으며 결과가 적용된 결과가 새로운 데이터프레임으로 리턴된다.

```
df2 = df.set_index('이름')
df2
```

인덱스 설정을 원본에 반영하기

데이터프레임 메서드중에서 `inplace=True`를 지원하는 경우 리턴값 없이 원본에 즉시 반영된다.

```
df.set_index('이름', inplace=True)
df
```

데이터 로딩시 인덱스 열을 지정하기

데이터파일을 가져오는 과정에서 인덱스로 사용할 열 이름을 미리 지정할 수 있다.

```
df = read_csv("https://data.hossam.kr/grade.csv", encoding="euc-kr", index_col='이름')
df
```

5) 엑셀 파일로부터 가져오기

인코딩을 지정하는 것을 제외하고는 `read_csv()` 함수와 동일하다.

```
from pandas import read_excel
```

```
df = read_excel("https://data.hossam.kr/grade.xlsx", index_col='이름')
df
```