Pandas 판다스

판다스(Pandas)는 파이썬 데이터 분석 라이브러리 중 하나로, 데이터 조작, 정제, 분석, 시각화 등을 위한 다양한 기능을 제공합니다.

- 데이터프레임(DataFrame): 2차원 데이터 구조로, 행과 열로 구성되어 있습니다. 엑셀 스프레드시트와 유사한 형태로 데이터를 저장하고 조작할 수 있습니다.
- 시리즈(Series): 1차원 데이터 구조로, 데이터의 배열 형태를 가지고 있습니다. 인덱스를 통해 각 데이터에 접근할 수 있습니다.
- 데이터 조작: 데이터의 필터링, 정렬, 집계, 변환 등의 다양한 조작 기능을 제공합니다.
- 데이터 입출력: CSV, Excel, SQL 등 다양한 포맷의 데이터를 읽고 쓸 수 있는 기능을 제공합니다.
- 시계열 데이터 처리: 시간 기반의 데이터를 쉽게 처리하고 분석할 수 있는 기능을 포함하고 있습니다.

판다스 설치

In [26]:
```
pip install pandas
```

```
Requirement already satisfied: pandas in /Users/youngjinseo/anaconda3/lib/python3.10/site-packages (2.1.4)
Requirement already satisfied: pytz>=2020.1 in /Users/youngjinseo/anaconda3/lib/python3.10/site-packages (from pandas) (2022.7)
Requirement already satisfied: numpy<2,>=1.22.4 in /Users/youngjinseo/anaconda3/lib/python3.10/site-packages (from pandas) (1.23.5)
Requirement already satisfied: tzdata>=2022.1 in /Users/youngjinseo/anaconda3/lib/python3.10/site-packages (from pandas) (2023.3)
Requirement already satisfied: python-dateutil>=2.8.2 in /Users/youngjinseo/anaconda3/lib/python3.10/site-packages (from pandas) (2.8.2)
Requirement already satisfied: six>=1.5 in /Users/youngjinseo/anaconda3/lib/python3.10/site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

DataFrame 객체

DataFrame 객체는 행과 열로 이루어진 2차원 데이터를 다루기 위한 객체입니다. 열은 각각의 변수를 나타내고, 행은 각각의 관측치를 나타냅니다. DataFrame 객체는 여러 가지 방법으로 생성할 수 있습니다.

In [1]:
```
# pandas 추출할때
import pandas as pd
```

In [27]:
```
data = [['A', 1], ['B', 2], ['C', 3]]
df = pd.DataFrame(data, columns = ['col1','col2'])
df
```

Out[27]:

	col1	col2
0	A	1
1	B	2
2	C	3

In [28]:
```
# 딕셔너리를 사용하여 DataFrame 객체 생성하기
data1 = {'col1': ['A','B','C'],'col2':[1,2,3]}
df1 = pd.DataFrame(data1)
df1
```

Out[28]:

	col1	col2
0	A	1
1	B	2
2	C	3

Series 객체

Series 객체는 인덱스와 값으로 이루어진 1차원 데이터를 다루기 위한 객체입니다. Series 객체는 DataFrame 객체에서 열을 선택하여 추출할 수 있습니다.

In [29]:
```
data = [1, 2, 3]
df3 = pd.Series(data, index =['a','b','c'])
df3
```

Out[29]:
```
a   1
b   2
c   3
dtype: int64
```

In [30]:
```
# 딕셔너리를 사용하여 Series 객체 생성하기
data1 = {'a':1,'b':2,'c':3}
df4 = pd.Series(data1)
df4
```

Out[30]:
```
a   1
b   2
c   3
dtype: int64
```

선택하기

In [31]:
```
# 데이터프레임 만들기
data3 = {'name':['Bob','Jessica','Mary','John','Mel'],
         'Birth':[968,155,77,578,973]}

df3 = pd.DataFrame(data3)
df3
```

Out[31]:

	name	Birth
0	Bob	968
1	Jessica	155
2	Mary	77
3	John	578
4	Mel	973

In [32]:
```
#name만 열선택
df3['name']
```

Out[32]:
```
0       Bob
1    Jessica
2       Mary
3       John
4        Mel
Name: name, dtype: object
```

In [33]:
```
#birth만 열선택
df3[['name','Birth']]
```

Out[33]:
```
0    968
1    155
2     77
3    578
4    973
Name: Birth, dtype: int64
```

In [34]:
```
#행 선택
df3.loc[0]
```

Out[34]:
```
name     Bob
Birth    968
Name: 0, dtype: object
```

In [35]:
```
#행 선택
df3.loc[[0,1,2]]
```

Out[35]:

	name	Birth
0	Bob	968
1	Jessica	155
2	Mary	77

조작하기

In [36]:
```
# 열 추가하기
df3['age'] = [27,19,30,24,30]
df3
```

Out[36]:

	name	Birth	age
0	Bob	968	27
1	Jessica	155	19
2	Mary	77	30
3	John	578	24
4	Mel	973	30

In [37]:
```
# 열 삭제하기
df3.drop('Birth',axis = 1,inplace = True)
df3
```

Out[37]:

	name	age
0	Bob	27
1	Jessica	19
2	Mary	30
3	John	24
4	Mel	30

In [38]:
```
#행 추가
df3.loc[5]=['Kelly',29]
df3
```

Out[38]:

	name	age
0	Bob	27
1	Jessica	19
2	Mary	30
3	John	24
4	Mel	30
5	Kelly	29

In [40]:
```
# 행 삭제
df3.drop(5,inplace = True)
df3
```

Out[40]:

	name	age
0	Bob	27
1	Jessica	19
2	Mary	30
4	Mel	30
5	Kelly	29

In [41]:
```
# 열 이름 바꾸기
df3.rename(columns = {'name':'person'},inplace = True)
df3
```

Out[41]:

	person	age
0	Bob	27
1	Jessica	19
2	Mary	30
4	Mel	30
5	Kelly	29

문제 풀어보기

1. 다음 조건을 만족하는 DataFrame을 만들어보세요

- 이름: 'Alice', 'Bob', 'Charlie', 'David', 'Eva'
- 나이: 23, 25, 22, 24, 23
- 성별: 'F', 'M', 'M', 'M', 'F'
- 수학 점수: 88, 92, 85, 90, 95

In [43]:
```
info = {'이름':['Alice','Bob','Charlie','David','Eva'],
        '나이':[23,25,22,24,23],
        '성별':['F','M','M','M','F'],
        '수학점수':[88,92,85,90,95]}

df6 = pd.DataFrame(info)

df6
```

Out[43]:

	이름	나이	성별	수학점수
0	Alice	23	F	88
1	Bob	25	M	92
2	Charlie	22	M	85
3	David	24	M	90
4	Eva	23	F	95

1. 이름,나이, 수학점수 (일)만 가져오시오.

In [44]:
```
df6[['이름','나이','수학점수']]
```

Out[44]:

	이름	나이	수학점수
0	Alice	23	88
1	Bob	25	92
2	Charlie	22	85
3	David	24	90
4	Eva	23	95

1. Bob부터 David 행만 가져오시요.

In [45]:
```
df6
```

Out[45]:

	이름	나이	성별	수학점수
0	Alice	23	F	88
1	Bob	25	M	92
2	Charlie	22	M	85
3	David	24	M	90
4	Eva	23	F	95

In [46]:
```
df6.loc[[1,2,3]]
```

Out[46]:

	이름	나이	성별	수학점수
1	Bob	25	M	92
2	Charlie	22	M	85
3	David	24	M	90

1. 열 '구' 추가하시요.

- 구 '강남','종로','마포','용산','강남'

In [47]:
```
df6['구'] =['강남','종로','마포','용산','강남']
df6
```

Out[47]:

	이름	나이	성별	수학점수	구
0	Alice	23	F	88	강남
1	Bob	25	M	92	종로
2	Charlie	22	M	85	마포
3	David	24	M	90	용산
4	Eva	23	F	95	강남

1. 행 'Charlie'만 삭제하세요.

In [48]:
```
df6.drop(2, inplace = True)
df6
```

Out[48]:

	이름	나이	성별	수학점수	구
0	Alice	23	F	88	강남
1	Bob	25	M	92	종로
3	David	24	M	90	용산
4	Eva	23	F	95	강남

1. 열'나이' 삭제하세요.

In [49]:
```
df6.drop('나이',axis = 1, inplace = True)
df6
```

Out[49]:

	이름	성별	수학점수	구
0	Alice	F	88	강남
1	Bob	M	92	종로
3	David	M	90	용산
4	Eva	F	95	강남