

▼ 1. 데이터분석 및 시각화 개요

1.1. 판다스 라이브러리란?



판다스(pandas)

- ▶ 데이터 분석을 위한 **Python**의 라이브러리
- ▶ 특히 시계열 데이터를 다루는 것과 연산이 뛰어나다

1
판다스 라이브러리와 데이터프레임

- 데이터프레임과 시리즈를 사용하는 라이브러리
- Python에서 표데이터를 쉽게 다룰 수 있게 해준다

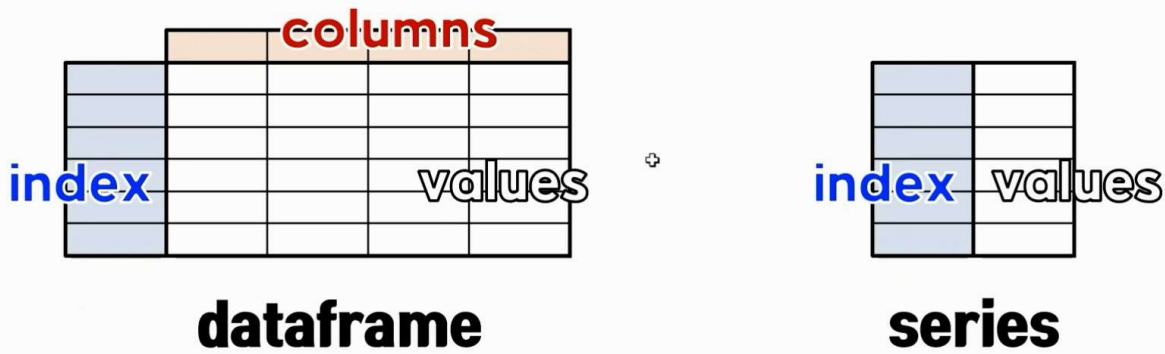
Python의 x

1. 데이터 분석을 위한 Python의 라이브러리
2. 특히 시계열 데이터를 다루는 것과 연산이 뛰어나다
3. 데이터프레임과 시리즈라는 자료형을 이용해 표데이터를 다룰 수 있게 해준다

판다스 = Python의 엑셀

▼ 1.2. 데이터프레임과 시리즈

1.2.1. 데이터프레임과 시리즈란 무엇인가?



데이터프레임과 시리즈는 판다스에서 다루는 핵심 자료형이다.

- 2차원 표 데이터가 데이터프레임이고 1차원 표데이터가 시리즈이다
- 표의 데이터부분을 values 라고 한다
- 표에서의 행이름을 index라고 한다

1.2.2. 데이터프레임과 시리즈의 특징

- 표 전체에 함수를 적용해 개별요소를 바꾼다.

| | 국어 | 영어 | 수학 | 과학 |
|-----|----|----|----|----|
| 송중기 | 67 | 93 | 91 | 88 |
| 김나현 | 75 | 69 | 96 | 69 |
| 권보아 | 75 | 81 | 74 | 82 |
| 박효신 | 96 | 65 | 84 | 66 |
| 김범수 | 79 | 70 | 76 | 75 |
| 이효리 | 62 | 99 | 87 | 76 |

df

| | 국어 | 영어 | 수학 | 과학 |
|-----|-------|-------|-------|-------|
| 송중기 | False | True | True | True |
| 김나현 | False | False | True | False |
| 권보아 | False | True | False | True |
| 박효신 | True | False | True | False |
| 김범수 | False | False | False | False |
| 이효리 | False | True | True | False |

df > 80

데이터 프레임 전체에 연산을 적용하는 것으로 개별 요소별로 80보다 큰지 여부를 True 또는 False로 반환한다

- 함수를 적용할 때는 전체가 아닌 부분이 필요하다면 키(key)값인 index와 columns를 이용한다

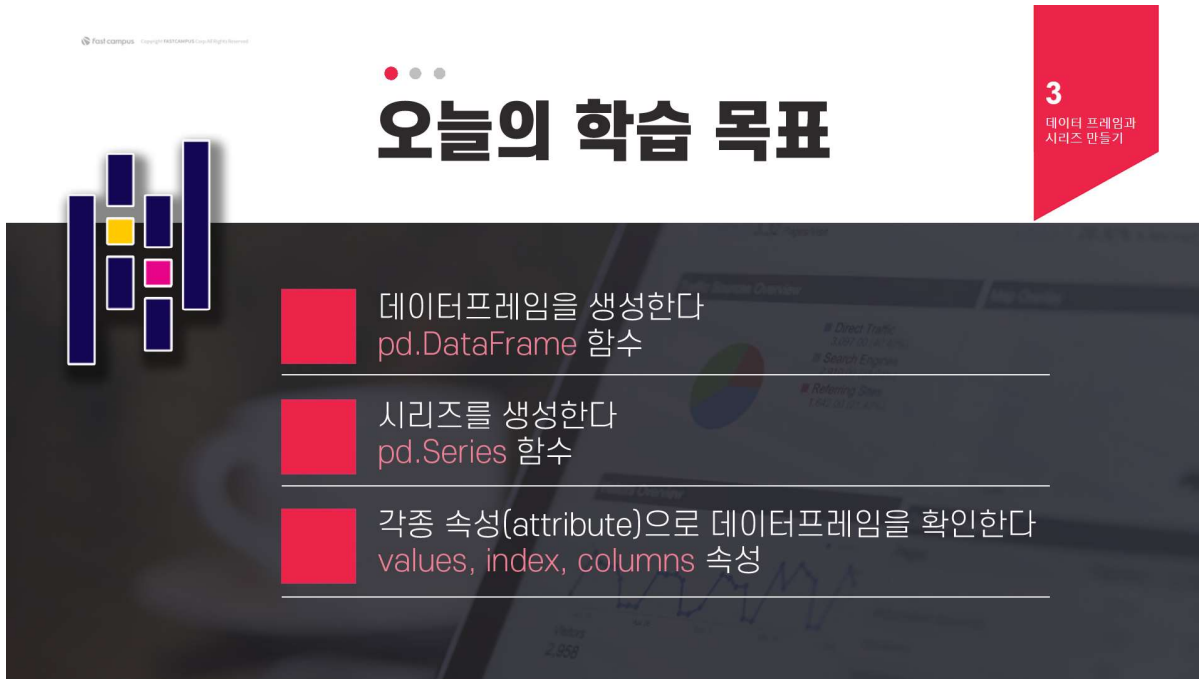
| | 국어 | 영어 | 수학 | 과학 |
|-----|----|----|----|----|
| 송중기 | 67 | 93 | 91 | 88 |
| 김나현 | 75 | 69 | 96 | 69 |
| 권보아 | 75 | 81 | 74 | 82 |
| 박효신 | 96 | 65 | 84 | 66 |
| 김범수 | 79 | 70 | 76 | 75 |
| 이효리 | 62 | 99 | 87 | 76 |

df

| | 국어 | 영어 | 수학 | 과학 |
|-----|----|----|----|----|
| 이효리 | 62 | 99 | 87 | 76 |
| 송중기 | 67 | 93 | 91 | 88 |
| 김나현 | 75 | 69 | 96 | 69 |
| 권보아 | 75 | 81 | 74 | 82 |
| 김범수 | 79 | 70 | 76 | 75 |
| 박효신 | 96 | 65 | 84 | 66 |

df.sort_values('국어')

▼ 1.3. 실습



오늘의 학습 목표

- 데이터프레임을 생성한다
`pd.DataFrame` 함수
- 시리즈를 생성한다
`pd.Series` 함수
- 각종 속성(attribute)으로 데이터프레임을 확인한다
`values`, `index`, `columns` 속성

위의 학습 목표 세가지를 중점적으로 실습을 해보자.

▼ 1.3.1 데이터프레임 생성하기

```
import pandas as pd
data1 = [[67, 93, 91, 88],
          [75, 69, 96, 69],
          [75, 81, 74, 82],
          [96, 65, 84, 66],
          [79, 70, 76, 75],
          [62, 99, 87, 76]]
idx1 = ['송중기', '김나현', '권보아', '박효신', '김범수', '이효리']
col1 = ['국어', '영어', '수학', '과학']
df1 = pd.DataFrame(data1, index=idx1, columns=col1)
df1
```

| | 국어 | 영어 | 수학 | 과학 |
|-----|----|----|----|----|
| 송중기 | 67 | 93 | 91 | 88 |
| 김나현 | 75 | 69 | 96 | 69 |

▼ 1.3.2 각종 속성(attribute)으로 데이터프레임 확인하기

- 속성은 함수와 비슷하게 약속된 데이터를 반환하지만 인자와 인수를 입력받지 않는다.
- 인자와 인수를 입력받지 않기 때문에 소괄호를 쓰지 않는다
- 대표적인 속성(attribute)은 values, columns, index 이다
- 데이터프레임 뿐만 아니라 시리즈도 values, index 속성으로 확인이 가능하다. (시리즈의 경우 columns는 불가능)

df1.values

```
array([[67, 93, 91, 88],
       [75, 69, 96, 69],
       [75, 81, 74, 82],
       [96, 65, 84, 66],
       [79, 70, 76, 75],
       [62, 99, 87, 76]])
```

df1.index

```
Index(['송중기', '김나현', '권보아', '박효신', '김범수', '이효리'], dtype='object')
```

df1.columns

```
Index(['국어', '영어', '수학', '과학'], dtype='object')
```

속성을 실행하면 두가지 자료형을 알려준다

df1.index

Index(['송중기', '김나현', '권보아', '박효신', '김범수', '이효리'], dtype='object')

각 셀의 자료형은 object

전체의 자료형은 Index

| | 국어 | 영어 | 수학 | 과학 |
|-----|----|----|----|----|
| 송중기 | 67 | 93 | 91 | 88 |
| 김나현 | 75 | 69 | 96 | 69 |
| 권보아 | 75 | 81 | 74 | 82 |
| 박효신 | 96 | 65 | 84 | 66 |
| 김범수 | 79 | 70 | 76 | 75 |
| 이효리 | 62 | 99 | 87 | 76 |

1. 데이터프레임을 생성한다
pd.DataFrame 함수
2. 시리지를 생성한다
pd.Series 함수
3. 각종 속성으로 확인한다
values, index, columns 속성

1.3.3. 시리즈 생성하기

```
data_k = [67, 75, 75, 96, 79, 62]
s1 = pd.Series(data_k, index=idx1)
s1
```

```
송중기    67
김나현    75
권보아    75
박효신    96
김범수    79
이효리    62
dtype: int64
```

```
# 시리즈도 벡터화 연산을 한다
s1 > 80
```

```
송중기    False
김나현    False
권보아    False
박효신     True
김범수    False
이효리    False
dtype: bool
```

- 시리즈도 index와 values 속성을 사용해 각각의 데이터를 리턴받을 수 있다
(columns 속성은 사용할 수 없다)

```
s1.index
```

```
Index(['송중기', '김나현', '권보아', '박효신', '김범수', '이효리'], dtype='object')
```

s1.values

```
array([67, 75, 75, 96, 79, 62])
```

▼ 1.4 연습문제

▼ 1.4.1 데이터프레임 만들기 연습문제(1)

- 아래와 같은 데이터프레임을 만들어라

| | 국어 | 영어 | 수학 |
|---|----|----|----|
| A | 67 | 93 | 91 |
| B | 75 | 69 | 96 |
| C | 75 | 81 | 74 |
| D | 96 | 65 | 84 |

```
data2 = [[67, 93, 91],[75, 69, 96],
          [75, 81, 74],[96, 65, 84]]
col2 = ['국어', '영어', '수학']
df2 = pd.DataFrame(data2, index=list('ABCD'), columns=col2)
df2
```

| | 국어 | 영어 | 수학 |
|---|----|----|----|
| A | 67 | 93 | 91 |
| B | 75 | 69 | 96 |
| C | 75 | 81 | 74 |
| D | 96 | 65 | 84 |

▼ 1.4.2 데이터프레임 만들기 연습문제(2)

- 아래와 같은 데이터프레임을 만들어라

| | col1 | col2 | col3 |
|---|------|------|------|
| 0 | 0 | 1 | 2 |
| 1 | 3 | 4 | 5 |
| 2 | 6 | 7 | 8 |

```
data3 = [[ 0, 1, 2], [ 3, 4, 5], [ 6, 7, 8], [ 9, 10, 11]]
pd.DataFrame(data3, columns=['col1', 'col2', 'col3'])
```

| | col1 | col2 | col3 |
|---|------|------|------|
| 0 | 0 | 1 | 2 |
| 1 | 3 | 4 | 5 |
| 2 | 6 | 7 | 8 |
| 3 | 9 | 10 | 11 |

▼ 1.4.3 데이터프레임 만들기 연습문제(3)

- 아래와 같은 데이터프레임을 만들어라

| | 국어학점 | 영어학점 | 수학학점 |
|-----|------|------|------|
| 송중기 | A | B | C |
| 김나현 | A | A | B |

```
data4 = [['A', 'B', 'C'], ['A', 'A', 'B']]
pd.DataFrame(data4, index=['송중기', '김나현'],
              columns=['국어학점', '영어학점', '수학학점'])
```

| | 국어학점 | 영어학점 | 수학학점 |
|-----|------|------|------|
| 송중기 | A | B | C |
| 김나현 | A | A | B |

▼ 1.5 마무리

- 다음 질문에 답해보자.

1. 판다스는 무엇인가
2. 데이터프레임이란?
3. 시리즈란?
4. 데이터프레임과 시리즈를 만들수 있는가?
5. 데이터프레임과 시리즈의 각 요소를 확인할수 있는가?

1.6. 참고문헌

1. 판다스 공식문서 `pd.DataFrame`

<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html>

2. 판다스 공식문서 `pd.Series`

<https://pandas.pydata.org/docs/reference/api/pandas.Series.html>

3. 엑셀투파이썬 유튜브 : 데이터프레임이란?

<https://youtu.be/SVjKsvvhWlQ>

