Pandas DataFrame

Pandas의 Series가 1차원 형태의 자료구조라면
DataFrame은 여러 개의 칼럼(column)으로 구성된 2차원 형태의 자료구조.

Pandas의 DataFrame을 사용하면
로우와 칼럼으로 구성된 2차원 구조의 데이터를
쉽게 저장하고 조작할 수 있다.

DataFrame 객체를 생성하는 가장 쉬운 방법은
파이썬의 딕셔너리를 사용하는 것.

딕셔너리를 통해 각 칼럼에 대한 데이터를 저장한 후,
딕셔너리를 DataFrame 클래스의 생성자 인자로 넘겨주면
DataFrame 객체가 생성된다.

- **딕셔너리를 사용한 DataFrame 객체 생성**

```
from pandas import Series, DataFrame
raw_data = {'col0':[1, 2, 3, 4],
            'col1':[10, 20, 30, 40],
            'col2':[100, 200, 300, 400]}

data = DataFrame(raw_data)
print(data)


#       Series('col0') Series('col2')
# index     value         value
# 0          1             10
# 1          2             20
# 2          3             30

# 위와 같이 구성되어있다.
# 딕셔너리 key값들은 col0,1,2로 총 3개로 되어있고, 시리즈가 총 3개로 각각의 인덱스가
```

> col0, col1, col2 라는 세 개의 칼럼이 존재
> 'col0', 'col1', 'col2' 라는 문자열은
> DataFrame의 각 칼럼을 인덱싱하는 데 사용된다.

> 로우 방향으로는
> Series와 유사하게 정수값으로 자동으로 인덱싱 된 것을
> 확인할 수 있다.

```
# 'col0', 'col1', 'col2' 를 사용하여 각 컬럼을 선택
print(data['col0'])
print(data['col0'])
print(data['col1'])
```

```
print(type(data['col0']))
# <class 'pandas.core.series.Series'>
```

> DataFrame에 있는 각 칼럼은 Series 객체임을 알 수 있다.
> 즉, DataFrame은
> 인덱스가 같은 여러개의 Series 객체로 구성된 자료구조.

> data라는 변수가 바인딩 하는 DataFrame에는 3개의 Series 객체가 있다.
> 이는 'col0', 'col1', 'col2'라는 키(key)에 각각 대응되는 값(value)이고
> 이것들을 하나의 파이썬 딕셔너리 객체로 생각하는 것.

> 따라서 'col0', 'col1', 'col2'라는 키(key)를 통해
> value에 해당하는 Series 객체에 접근할 수 있다.

```
daeshin = {'open': [11650, 11100, 11200, 11100, 11000],
           'high': [12100, 11800, 11200, 11100, 11150],
           'low' : [11600, 11050, 10900, 10950, 10900],
           'close': [11900, 11600, 11000, 11100, 11050]}

daeshin_day = DataFrame(daeshin)
print(daeshin_day)

#     open   high    low  close
# 0  11650  12100  11600  11900
# 1  11100  11800  11050  11600
# 2  11200  11200  10900  11000
# 3  11100  11100  10950  11100
# 4  11000  11150  10900  11050
```

> DataFrame 객체에서 칼럼의 순서는
> DataFrame 객체를 생성할 때 columns라는 키워드를 지정할 수 있다.

```
daeshin_day2 = DataFrame(daeshin,
                         columns = ['CLOSE', 'OPEN', 'HIGH', 'LOW'])
print(daeshin_day2)

# Empty DataFrame
# Columns: [CLOSE, OPEN, HIGH, LOW]
# Index: []

daeshin_day2 = DataFrame(daeshin,
                         columns = ['close', 'open', 'high', 'low'])
print(daeshin_day2)

#    close   open   high    low
# 0  11900  11650  12100  11600
# 1  11600  11100  11800  11050
# 2  11000  11200  11200  10900
# 3  11100  11100  11100  10950
# 4  11050  11000  11150  10900
```

DataFrame에서 인덱스 역시 DataFrame 객체를 생성하는 시점에 index를 통해 지정할 수 있다.

먼저 인덱싱에 사용할 값을 만든 후,
이를 DataFrame 객체 생성 시점에 지정하면 된다.

```
date = ['20.02.29', '20.02.26', '20.02.25', '20.02.24', '20.02.23']
daeshin_day3 = DataFrame(daeshin,
                         columns = ['open', 'high', 'low', 'close'],
                         index = date)
print(daeshin_day3)

#            open   high    low  close
# 20.02.29  11650  12100  11600  11900
# 20.02.26  11100  11800  11050  11600
# 20.02.25  11200  11200  10900  11000
# 20.02.24  11100  11100  10950  11100
# 20.02.23  11000  11150  10900  11050
```

- **DataFrame 칼럼, 로우 선택**

종가를 기준으로만 데이터를 분석한다면
'close' 칼럼에 대한 데이터만을 DataFrame 객체로부터 얻어낸다.

```
close = daeshin_day3['close']
print(close)

# 20.02.29    11900
# 20.02.26    11600
# 20.02.25    11000
# 20.02.24    11100
# 20.02.23    11050
# Name: close, dtype: int64
```

DataFrame 객체의 칼럼 이름과 인덱스 값을 확인하려면
각각 columns와 index 속성을 사용

```
print(daeshin_day.columns)
print(daeshin_day.index)

# Index(['open', 'high', 'low', 'close'], dtype='object')
# RangeIndex(start=0, stop=5, step=1)
```