

[TM_Python_Pandas] 02. Pandas 데이터 분석 기초 => DataFrame

- 학습 목표
 - Pandas의 자료형 DataFrame 대해 이해할 수 있다.
 - 간단한 DataFrame 자료형을 만들어 보고 실습해 본다.
- 학습 내용
 - 2-1 Pandas의 DataFrame의 이해
 - (가) DataFrame의 객체 생성하기
 - (나) DataFrame의 객체 생성하기(2)
 - [실습해 보기2]

2-1 Pandas의 DataFrame의 이해

pandas의 Series가 1차원 형태의 자료구조라면,
DataFrame은 여러개의 컬럼(Column)으로 구성된 2차원 형태의 자료구조이다.

일자별								
일자	증가	전일비	등락률	거래량	거래대금	개인(억)	외국인(억)	기관(억)
18.03.19	2,475.03	▼18.94	-0.76%	477,219	5,994,568	+1,483	-833	-531
18.03.16	2,493.97	▲1.59	+0.06%	377,766	6,566,428	+12	+1,637	-196
18.03.15	2,492.38	▲6.30	+0.25%	344,009	6,065,884	+194	-420	-121
18.03.14	2,486.08	▼8.41	-0.34%	360,865	6,379,749	+2,840	-1,294	-2,300
18.03.13	2,494.49	▲10.37	+0.42%	350,203	7,828,709	-3,103	+5,817	-2,278
18.03.12	2,484.12	▲24.67	+1.00%	391,303	6,570,618	-2,917	+1,938	+618
18.03.09	2,459.45	▲26.37	+1.08%	437,265	8,573,691	-4,900	+1,989	+2,015
18.03.08	2,433.08	▲31.26	+1.30%	394,848	9,053,040	-4,264	-1,742	+5,738
18.03.07	2,401.82	▼9.59	-0.40%	412,971	8,956,678	+3,584	-2,595	-523
18.03.06	2,411.41	▲36.35	+1.53%	400,110	6,598,571	-1,713	+130	+1,033

1 2 3 4 5 6 7 8 9 10 11~20 >

(가) DataFrame의 객체 생성하기

- (A) 가장 간단한 방법은 파이썬의 딕셔너리를 이용하는 방법이다.
 (B) 딕셔너리를 통해 칼럼에 대한 데이터를 저장.
 (C) DataFrame의 객체를 만든다.

```
In [1]: 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)
data
```

```
Out[1]:
```

	col0	col1	col2
0	1	10	100
1	2	20	200
2	3	30	300
3	4	40	400

```
In [2]: print(data['col0'])
        print(data['col1'])
        print(data['col2'])

0    1
1    2
2    3
3    4
Name: col0, dtype: int64
0    10
1    20
2    30
3    40
Name: col1, dtype: int64
0    100
1    200
2    300
3    400
Name: col2, dtype: int64
```

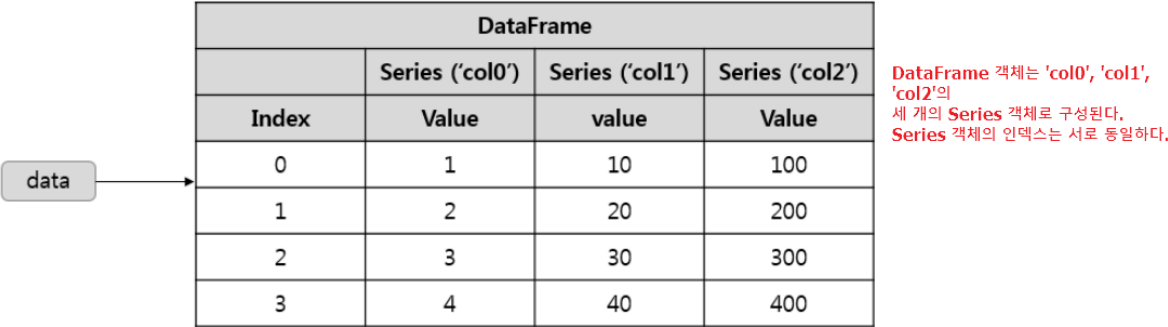
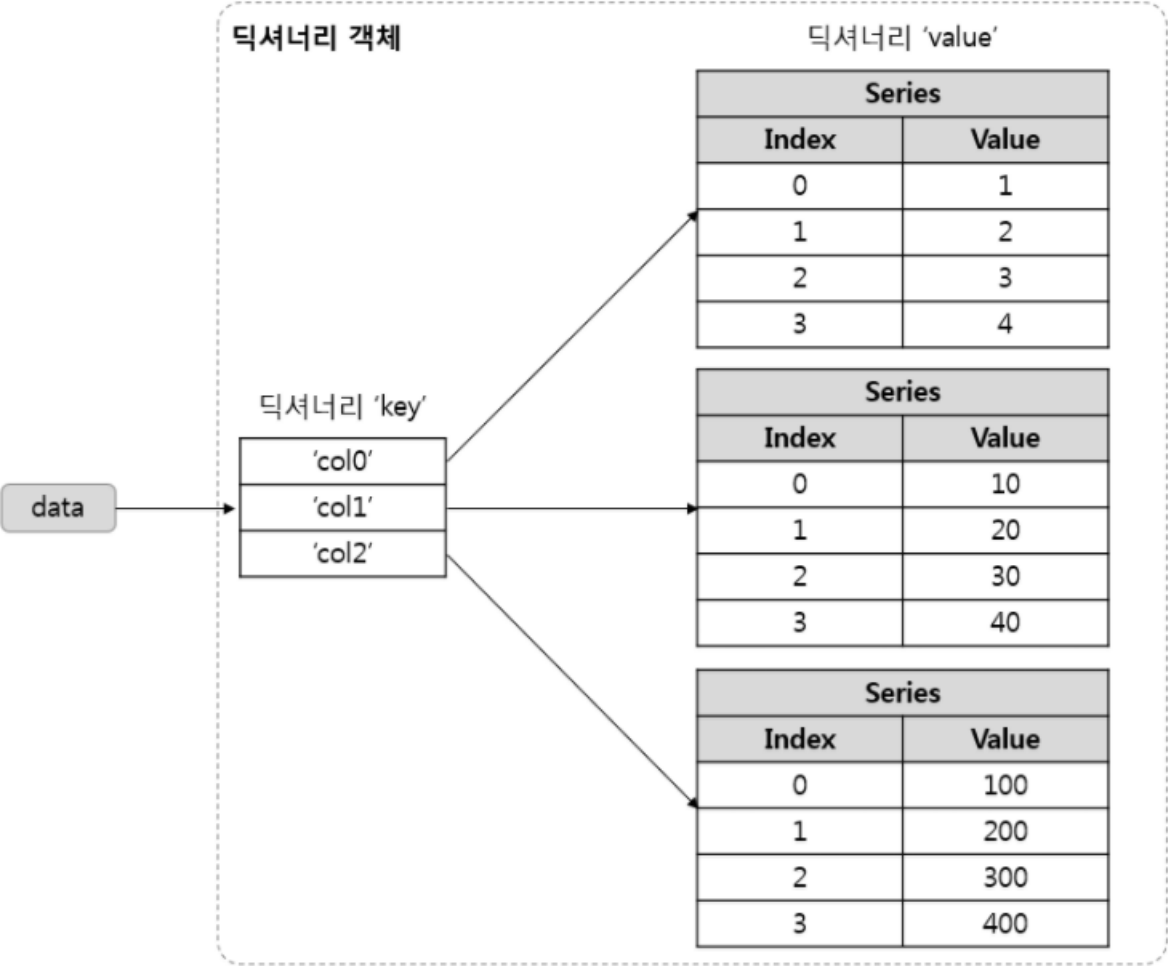


그림 13.10 DataFrame 객체의 내부 구조



(나) DataFrame의 객체 생성하기(2)

(A) 파이썬 딕셔너리 형태로 각 컬럼 대응 데이터 표현.

(B) DataFrame 클래스의 생성자로 넘겨주기

* 컬럼의 순서가 다르게 나타날 경우, columns를 이용하여 컬럼의 순서를 지정 할 수 있다.

In [3]: `from pandas import Series, DataFrame`

```
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)
```

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

In [5]: `date = ['16.02.29', '16.02.26', '16.02.25', '16.02.24', '16.02.23']`
`daeshin_day = DataFrame(daeshin, columns=['open', 'high', 'low', 'close'], index=date)`
`daeshin_day`

Out[5]:

	open	high	low	close
16.02.29	11650	12100	11600	11900
16.02.26	11100	11800	11050	11600
16.02.25	11200	11200	10900	11000
16.02.24	11100	11100	10950	11100
16.02.23	11000	11150	10900	11050

[실습해보기 2]

종목명	금액▼	수량▼	등락률▼
SK하이닉스	30,406	335	0.11%
삼성전기	14,179	142	1.74%
셀트리온	13,275	41	-1.54%
삼성SDI	10,121	48	-4.41%
POSCO	9,558	27	0.29%

위의 그림과 동일한 DataFrame 객체를 생성해 보자.

In []: