[2-3. 데이터 합치고, 변형하기(merge, concat)]

1. pandas.concat 함수

- 데이터프레임을 축에 따라서 이어붙이는 pandas 함수
- 연결하는 기준이 되는 칼럼 또는 로우에 대하여 교집합 또는 합집합 방식으로 연결할 수 있음.
- 주요 인자
  - axis : 0(default) 로우 방향, 1 칼럼 방향
  - join : 'outer'(default) 외부조인(합집합), 'inner' 내부조인(교집합)
  - ignore_index : 연결된 인덱스를 유지하지 않고, 새로운 인덱스를 생성

In [1]:
```
from pandas import Series, DataFrame
import pandas as pd
import numpy as np
```

In [2]:
```
data = [[10, 100, 1000],
        [20, 200, 2000],
        [30, 300, 3000],
        [40, 400, 4000]]
df1 = DataFrame(data, columns=['data0', 'data1', 'data2'],
                index=['one', 'two', 'three', 'four'])
```

In [3]:
```
data = [[15, 150, 1500],
        [25, 250, 2500],
        [35, 350, 3500],
        [45, 450, 4500]]
df2 = DataFrame(data, columns=['data1', 'data2', 'data3'],
                index=['one', 'two', 'five', 'six'])
```

In [4]:
```
pd.concat([df1, df2]) #axis=0, join='outer' default
```

Out[4]:

	data0	data1	data2	data3
one	10.0	100	1000	NaN
two	20.0	200	2000	NaN
three	30.0	300	3000	NaN
four	40.0	400	4000	NaN
one	NaN	15	150	1500.0
two	NaN	25	250	2500.0
five	NaN	35	350	3500.0
six	NaN	45	450	4500.0

In [5]:
```
pd.concat([df1, df2], join='inner') #axis=0, join='inner'
```

Out[5]:

	data1	data2
one	100	1000
two	200	2000
three	300	3000
four	400	4000
one	15	150
two	25	250
five	35	350
six	45	450

In [6]:
```
pd.concat([df1, df2], join='inner', ignore_index=True)
#[0 ~ 인덱스 길이] 만큼의 새로운 인덱스를 생성
```

Out[6]:

	data1	data2
0	100	1000
1	200	2000
2	300	3000
3	400	4000
4	15	150
5	25	250
6	35	350
7	45	450

In [7]:
```
pd.concat([df1, df2], axis=1)  #join='outer' default
```

Out[7]:

	data0	data1	data2	data1	data2	data3
one	10.0	100.0	1000.0	15.0	150.0	1500.0
two	20.0	200.0	2000.0	25.0	250.0	2500.0
three	30.0	300.0	3000.0	NaN	NaN	NaN
four	40.0	400.0	4000.0	NaN	NaN	NaN
five	NaN	NaN	NaN	35.0	350.0	3500.0
six	NaN	NaN	NaN	45.0	450.0	4500.0

In [8]:
```
pd.concat([df1, df2], axis=1, join='inner')
```

Out[8]:

	data0	data1	data2	data1	data2	data3
one	10	100	1000	15	150	1500
two	20	200	2000	25	250	2500

In [9]:
```
pd.concat([df1, df2], ignore_index=True)
```

Out[9]:

	data0	data1	data2	data3
0	10.0	100	1000	NaN
1	20.0	200	2000	NaN
2	30.0	300	3000	NaN
3	40.0	400	4000	NaN
4	NaN	15	150	1500.0
5	NaN	25	250	2500.0
6	NaN	35	350	3500.0
7	NaN	45	450	4500.0

In [ ]:
```

```

2. pandas.merge 함수

- 하나 이상의 key(칼럼값)을 기준으로 데이터프레임을 결합함.
- 엑셀의 vlookup 함수와 유사한 결과물을 도출하는데 사용함.

In [10]:
```
data = [['a', 100, 1000],
        ['b', 200, 2000],
        ['c', 300, 3000],
        ['a', 400, 4000]]
df1 = DataFrame(data, columns=['key', 'data1', 'data2'],
                index=['one', 'two', 'three', 'four'])
```

In [11]:
```
data = [['a', 150, 1500],
        ['c', 250, 2500],
        ['b', 350, 3500],
        ['d', 450, 4500]]
df2 = DataFrame(data, columns=['key', 'data10', 'data20'],
                index=['one', 'two', 'five', 'six'])
```

In [12]:
```
pd.merge(df1, df2, on='key')
```

Out[12]:

	key	data1	data2	data10	data20
0	a	100	1000	150	1500
1	a	400	4000	150	1500
2	b	200	2000	350	3500
3	c	300	3000	250	2500

- 인덱스는 무시되고 새로운 인덱스가 생성됨.
- default로 join 기준이 교집합(내부조인)으로 설정되어 결과물이 도출됨.

In [ ]:
```
```

join 기준을 합집합(외부조인) 또는 왼쪽, 오른쪽 우선 방식으로 merge

=> how 인자를 사용함(default는 'inner')

In [13]:
```
pd.merge(df1, df2, on='key', how='outer')  #합집합(외부조인)
```

Out[13]:

	key	data1	data2	data10	data20
0	a	100.0	1000.0	150	1500
1	a	400.0	4000.0	150	1500
2	b	200.0	2000.0	350	3500
3	c	300.0	3000.0	250	2500
4	d	NaN	NaN	450	4500

In [14]:
```
pd.merge(df1, df2, on='key', how='left')  #왼쪽 데이터 우선 join
```

Out[14]:

	key	data1	data2	data10	data20
0	a	100	1000	150	1500
1	b	200	2000	350	3500
2	c	300	3000	250	2500
3	a	400	4000	150	1500

In [15]:
```
pd.merge(df1, df2, on='key', how='right')  #오른쪽 데이터 우선 join
```

Out[15]:

	key	data1	data2	data10	data20
0	a	100.0	1000.0	150	1500
1	a	400.0	4000.0	150	1500
2	c	300.0	3000.0	250	2500
3	b	200.0	2000.0	350	3500
4	d	NaN	NaN	450	4500

In [ ]:
```
```

서로 다른 칼럼명을 기준으로 merge하고자 하는 경우

=> left_on, right_on 인자를 사용함

In [16]:
```
data = [['a', 100, 1000],
        ['b', 200, 2000],
        ['c', 300, 3000],
        ['a', 400, 4000]]
df1 = DataFrame(data, columns=['keyL', 'data1', 'data2'],
                index=['one', 'two', 'three', 'four'])
```

In [17]:
```
data = [['a', 150, 1500],
        ['c', 250, 2500],
        ['b', 350, 3500],
        ['d', 450, 4500]]
df2 = DataFrame(data, columns=['keyR', 'data10', 'data20'],
                index=['one', 'two', 'five', 'six'])
```

In [18]:
```
pd.merge(df1, df2, left_on = 'keyL', right_on = 'keyR')
```

Out[18]:

	keyL	data1	data2	keyR	data10	data20
0	a	100	1000	a	150	1500
1	a	400	4000	a	150	1500
2	b	200	2000	b	350	3500
3	c	300	3000	c	250	2500

In [ ]:
```
```