In [3]:

```
import pandas as pd
```

In [13]:

```
df1 = pd.DataFrame([['a', 1], ['b', 2]], columns = ['letter', 'number'])
df2 = pd.DataFrame([['c', 3], ['d', 4]], columns = ['letter', 'number'])
df3 = pd.DataFrame([['e', 5, '!'], ['f', 6, '@']], columns = ['letter', 'number', 'etc']
)
```

In [14]:

```
df1
```

Out[14]:

	letter	number
0	a	1
1	b	2

In [15]:

```
df2
```

Out[15]:

	letter	number
0	c	3
1	d	4

In [16]:

```
df3
```

Out[16]:

	letter	number	etc
0	e	5	!
1	f	6	@

1.1. 컬럼명 기준으로 연결

`pd.concat(데이터프레임리스트)`

같은 컬럼 리스트를 가진 두 개의 다른 데이터프레임을 연결하면
두 데이터프레임을 합한 형태가 된다.
`pd.concat([df1, df2])`

In [20]:

```
df_rowconcat = pd.concat([df1, df2, df3])
df_rowconcat
```

Out[20]:

	letter	number	etc
0	a	1	NaN

	letter	number	etc
1	b	2	NaN
0	c	3	NaN
1	d	4	NaN
0	e	5	!
1	f	6	@

1.1.1 공통된 컬럼만 남기기

```
join = 'inner'
```

- **inner join**에 해당

In [23]:

```
df_rowconcat = pd.concat([df1, df2, df3], join = 'inner')
df_rowconcat
```

Out[23]:

	letter	number
0	a	1
1	b	2
0	c	3
1	d	4
0	e	5
1	f	6

인덱스가 그대로 들어가기 때문에 중복된 인덱스 발생

In [24]:

```
df_rowconcat.loc[0]
```

Out[24]:

	letter	number
0	a	1
0	c	3
0	e	5

인덱스 재지정 필요

In [25]:

```
df_rowconcat = pd.concat([df1, df2, df3], join = 'inner', ignore_index = 'True')
df_rowconcat
```

Out[25]:

	letter	number
0	a	1
1	b	2
2	c	3
3	d	4
4	e	5

	letter	number
5	f	6

1.2. 인덱스 기준으로 연결

`pd.concat(데이터프레임리스트, axis = 1)`

두 개의 데이터프레임의 인덱스가 같고, 컬럼명이 다르다면,
해당 컬럼의 데이터가 같은 각 인덱스의 뒤에 붙는 형태가 된다.
`pd.concat([df1, df2], axis = 1)`

In [34]:

```
# 샘플 데이터
df4 = pd.DataFrame({'age' : [20, 21, 22]}, index = ['amy', 'james', 'david'])
df5 = pd.DataFrame({'phone' : ['010-111-1111', '010-222-2222', '010-333-3333']}, index =
['amy', 'james', 'david'])
df6 = pd.DataFrame({'job' : ['student', 'programmer', 'ceo', 'designer']}, index = ['amy
', 'james', 'david', 'J'])
```

In [35]:

```
df4
```

Out[35]:

	age
amy	20
james	21
david	22

In [36]:

```
df5
```

Out[36]:

	phone
amy	010-111-1111
james	010-222-2222
david	010-333-3333

In [37]:

```
df6
```

Out[37]:

	job
amy	student
james	programmer
david	ceo
J	designer

In [38]:

```
pd.concat([df4, df5, df6], axis = 1)
```

Out[38]:

	age	phone	job
amy	20.0	010-111-1111	student
james	21.0	010-222-2222	programmer
david	22.0	010-333-3333	ceo
J	NaN	NaN	designer

1.2.1 공통된 인덱스만 남기기

In [40]:

```
df_column_concat = pd.concat([df4, df5, df6], axis = 1, join='inner')
df_column_concat
```

Out[40]:

	age	phone	job
amy	20	010-111-1111	student
james	21	010-222-2222	programmer
david	22	010-333-3333	ceo

2. 공통된 열을 기준으로 연결하기(merge)

`pd.merge(left = 왼쪽 데이터프레임, right = 오른쪽 데이터프레임, on = 기준 컬럼, how = 연결 방법)`

- 2개의 데이터프레임을 연결한다.

In [41]:

```
df = pd.read_csv('./data/scores.csv')
df
```

Out[41]:

	name	kor	eng	math
0	Aiden	100.0	90.0	95.0
1	Charles	90.0	80.0	75.0
2	Danial	95.0	100.0	100.0
3	Evan	100.0	100.0	100.0
4	Henry	NaN	35.0	60.0
5	Ian	90.0	100.0	90.0
6	James	70.0	75.0	65.0
7	Julian	80.0	90.0	55.0
8	Justin	50.0	60.0	100.0
9	Kevin	100.0	100.0	90.0
10	Leo	90.0	95.0	70.0
11	Oliver	70.0	75.0	65.0
12	Peter	100.0	95.0	100.0
13	Amy	90.0	75.0	90.0
14	Chloe	95.0	100.0	95.0
15	Danna	100.0	100.0	100.0
16	Ellen	NaN	60.0	NaN
17	Emma	70.0	65.0	70.0

	name	kor	eng	math
18	Jennifer	80.0	55.0	80.0
19	Kate	50.0	NaN	50.0
20	Linda	100.0	90.0	100.0
21	Olivia	90.0	70.0	90.0
22	Rose	70.0	65.0	70.0
23	Sofia	100.0	100.0	100.0
24	Tiffany	90.0	NaN	90.0
25	Vanessa	95.0	70.0	95.0
26	Viviana	100.0	80.0	100.0
27	Vikkie	NaN	50.0	100.0
28	Winnie	70.0	100.0	70.0
29	Zuly	80.0	90.0	95.0

In [42]:

```
df7 = df.loc[[1, 2, 3]][['name', 'eng']]
df7
```

Out[42]:

	name	eng
1	Charles	80.0
2	Danial	100.0
3	Evan	100.0

In [43]:

```
df8 = df.loc[[1, 2, 4]][['name', 'math']]
df8
```

Out[43]:

	name	math
1	Charles	75.0
2	Danial	100.0
4	Henry	60.0

2.1. 공통 데이터만으로 연결

`how = 'inner' ( default )`

In [45]:

```
pd.merge(df7, df8, on = 'name')
# name 컬럼에 항목이 겹치는 Charles, Danial만 추출된 후 조인된다.
```

Out[45]:

	name	eng	math
0	Charles	80.0	75.0
1	Danial	100.0	100.0

In [46]:

```
pd.merge(df7, df8, on = 'name', how = 'inner')
```

Out[46]:

	name	eng	math
0	Charles	80.0	75.0
1	Danial	100.0	100.0

2.2. 모든 행 연결

In [47]:

```
pd.merge(df7, df8, on = 'name', how = 'outer')
# 공통되지 않는, NaN 값이 존재하는 Evan, Henry도 추출되어 모두 다 join된다.
```

Out[47]:

	name	eng	math
0	Charles	80.0	75.0
1	Danial	100.0	100.0
2	Evan	100.0	NaN
3	Henry	NaN	60.0

2.3. 왼쪽 데이터베이스 기준으로 연결

In [48]:

```
pd.merge(df7, df8, on='name', how='left')
```

Out[48]:

	name	eng	math
0	Charles	80.0	75.0
1	Danial	100.0	100.0
2	Evan	100.0	NaN

2.4. 오른쪽 데이터베이스 기준으로 연결

In [49]:

```
pd.merge(df7, df8, on='name', how='right')
```

Out[49]:

	name	eng	math
0	Charles	80.0	75.0
1	Danial	100.0	100.0
2	Henry	NaN	60.0