

In [3]:

```
import pandas as pd
df = pd.read_csv('./data/scores.csv')
df = df.head(2)
df
```

Out[3]:

	name	kor	eng	math
0	Aiden	100.0	90.0	95.0
1	Charles	90.0	80.0	75.0

## 1. 열을 행으로 보내기( column -> row )

### 1.1. 모든 열 melt

```
df.melt()
pd.melt(df)
```

In [4]:

```
df.melt()
# (name, name[0]), (name, name[1]), (kor, kor[0]), (kor, kor[1]), , , ,
```

Out[4]:

	variable	value
0	name	Aiden
1	name	Charles
2	kor	100.0
3	kor	90.0
4	eng	90.0
5	eng	80.0
6	math	95.0
7	math	75.0

### 1.2. 고정할 컬럼 지정하여 melt

id\_vars = [열 이름 리스트] --> 위치를 그대로 유지할 열 이름

In [5]:

```
# name 고정
df.melt(id_vars = 'name')
# name 컬럼은 고정된 채로 다른 컬럼만 넘어간다.
```

Out[5]:

	name	variable	value
0	Aiden	kor	100.0
1	Charles	kor	90.0
2	Aiden	eng	90.0

	name	variable	value
3	Charles	eng	80.0
4	Aiden	math	95.0
5	Charles	math	75.0

In [6]:

```
# 'name', 'kor' 고정
df.melt(id_vars = ['name', 'kor'])
```

Out[6]:

	name	kor	variable	value
0	Aiden	100.0	eng	90.0
1	Charles	90.0	eng	80.0
2	Aiden	100.0	math	95.0
3	Charles	90.0	math	75.0

### 1.3. 행으로 위치를 변경할 열 고정

```
values_vars = [열 이름 리스트]
```

In [7]:

```
# kor
df.melt(id_vars = 'name', value_vars = 'kor')
# 둘 다 설정되지 않은 항목은 삭제된다. 기존 값과 원하는 value 값이 모두 설정되었으니 이외의 값은 모두 삭제
```

Out[7]:

	name	variable	value
0	Aiden	kor	100.0
1	Charles	kor	90.0

In [8]:

```
# kor, eng
df.melt(id_vars = 'name', value_vars = ['kor', 'eng'])
```

Out[8]:

	name	variable	value
0	Aiden	kor	100.0
1	Charles	kor	90.0
2	Aiden	eng	90.0
3	Charles	eng	80.0

### 1.4 컬럼명 변경하기

`` var\_name = 컬럼명 // value\_vars로 위치를 변경한 열 이름 value\_name = var\_name // 으로 위치를 변경한 열의 데이터 저장한 열 이름

In [11]:

```
# subject, score
df.melt(id_vars = 'name', value_vars = ['kor', 'eng'], var_name='subject', value_name='score')
```

```
# variable 부분과 value 부분이 수정되었다.
```

Out[11]:

	name	subject	score
0	Aiden	kor	100.0
1	Charles	kor	90.0
2	Aiden	eng	90.0
3	Charles	eng	80.0

## 2. 열을 행으로 보내기( pivot )

In [12]:

```
df = pd.read_csv('./data/scores.csv')
df = df.head(2)
df = df.melt(id_vars = 'name', var_name = 'subject', value_name='score')
```

```
def get_grade(x):
    if x >= 90: grade = 'A'
    elif x >= 80: grade = 'B'
    elif x >= 70: grade = 'C'
    elif x >= 60: grade = 'D'
    else: grade = 'F'
    return grade
```

```
df['grade'] = df['score'].apply(get_grade) # 함수 적용
df = df.sort_values('name') # 이름 단위로 정렬
df
```

Out[12]:

	name	subject	score	grade
0	Aiden	kor	100.0	A
2	Aiden	eng	90.0	A
4	Aiden	math	95.0	A
1	Charles	kor	90.0	A
3	Charles	eng	80.0	B
5	Charles	math	75.0	C

```
df.pivot(
    index = 인덱스로 사용할 컬럼,
    columns = 컬럼으로 사용할 컬럼,
    values = 값으로 사용할 컬럼
)
```

In [13]:

```
# name, subject, score
df.pivot(index = 'name', columns = 'subject', values = 'score')
# 각 columns로 subject의 각 값들이 각 컬럼으로 들어간다.
# 그리고 각 column의 전체 인덱스로 subject가 된다.

# 그리고 df의 각 값에 들어갈 내용이 score 행의 내용이다.

# 각 행의 인덱스가 바로 name에 들어가는 내용이다.
```

Out[13]:

subject	eng	kor	math
name			
Aiden	90.0	100.0	95.0
Charles	80.0	90.0	75.0

In [15]:

```
# name, subject, grade
df.pivot(index = 'name', columns = 'subject', values = 'grade')
```

Out[15]:

subject	eng	kor	math
name			
Aiden	A	A	A
Charles	B	A	C

In [17]:

```
# name, subject, [grade, score]
df.pivot(index = 'name', columns = 'subject', values = ['grade', 'score'])
# values로 들어가는 값의 범주가 2가지이면, 각각의 범주마다
# 각각의 columns가 만들어지고, 2종류의 df를 합친 모습이 된다.
```

Out[17]:

grade				score		
subject	eng	kor	math	eng	kor	math
name						
Aiden	A	A	A	90.0	100.0	95.0
Charles	B	A	C	80.0	90.0	75.0

In [18]:

```
# name, subject
df.pivot(index = 'name', columns = 'subject')
# 위의 결과와 같다. name, subject을 제외한 모든 항목이 values에 들어간다.
```

Out[18]:

score				grade		
subject	eng	kor	math	eng	kor	math
name						
Aiden	90.0	100.0	95.0	A	A	A
Charles	80.0	90.0	75.0	B	A	C

### 3. 행과 열 바꾸기

```
df.transpose()
```

In [19]:

```
df.head()
```

Out[19]:

	name	subject	score	grade
0	Aiden	kor	100.0	A

2	name	subject	score	grade
	Aiden	eng	90.0	A
4	Aiden	math	95.0	A
1	Charles	kor	90.0	A
3	Charles	eng	80.0	B

In [20]:

```
df.transpose()
# column의 맨 처음 컬럼 항목부터 묶어서 차례로 보여준다.
```

Out[20]:

	0	2	4	1	3	5
name	Aiden	Aiden	Aiden	Charles	Charles	Charles
subject	kor	eng	math	kor	eng	math
score	100.0	90.0	95.0	90.0	80.0	75.0
grade	A	A	A	A	B	C