

In [1]:

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

1. 함수로 데이터 변경하기

- 함수에 파라미터가 없는 경우 `def 함수명:,,, 함수 내용,,, return 결과값`

함수 적용

`df[컬럼].apply(함수명)`

- 함수에 파라미터가 있는 경우 `def 함수명(매개변수):,,, 함수 내용,,, return 결과값`

함수 적용

`df[컬럼].apply(함수명, 매개변수 = 값)`

In [2]:

```
df_copy.head()
```

Out[2]:

	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

In [4]:

```
# df['math']의 모든 점수에 5점 더하기 ( --> 105점? )
df['math'] = df['math'] + 5
df['math'].values
```

Out[4]:

```
array([105.,  85., 110., 110.,  70.])
```

In [5]:

```
# 함수 만들기
def plus5(x):
    score = x + 5
    if score >= 100:
        return 100
    return score
```

In [8]:

```
# df['math']의 모든 점수에 5점 더하기 : 100점을 넘을 수 없다.
df['math'].apply(plus5)
```

Out[8]:

```
array([100.,  85., 110., 110.,  70.])
```

```
0      100.0
1      90.0
2     100.0
3     100.0
4      75.0
Name: math, dtype: float64
```

In [9]:

```
# 위의 내용은 출력만 한 것이고, 이는 값변경 적용이 안된다.
df['math'] = df['math'].apply(plus5)
```

In [10]:

```
df['eng']
```

Out[10]:

```
0      90.0
1      80.0
2     100.0
3     100.0
4      35.0
Name: eng, dtype: float64
```

In [12]:

```
def plusn(x, n):
    score = x + n
    if score >= 100:
        return 100
    else:
        return score
```

In [13]:

```
# df['eng']의 모든 점수에 n점 더하기 : 100점을 넘을 수 없다.
n = 3
df['eng'] = df['eng'].apply(plusn, n = n)
df['eng']
```

Out[13]:

```
0      93.0
1      83.0
2     100.0
3     100.0
4      38.0
Name: eng, dtype: float64
```

3. 함수로 행/열의 데이터 집계하기

~~ **df.apply(함수명, axis = 0)** : 열 단위로 함수가 적용된다. **df.apply(함수명, axis = 1)** : 행 단위로 함수가 적용된다.

In [18]:

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

Out[18]:

	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
   name kor eng math
```

```
In [19]:
```

```
df.index = df.name
df.drop(columns = ['name'], inplace = True)
df_copy = df.copy()
```

```
In [20]:
```

```
df.index
```

```
Out[20]:
```

```
Index(['Aiden', 'Charles', 'Danial', 'Evan', 'Henry'], dtype='object', name='name')
```

```
In [21]:
```

```
df_copy.head()
```

```
Out[21]:
```

	kor	eng	math
name			
Aiden	100.0	90.0	95.0
Charles	90.0	80.0	75.0
Danial	95.0	100.0	100.0
Evan	100.0	100.0	100.0
Henry	NaN	35.0	60.0

```
In [22]:
```

```
# 행단위 데이터 출력
def print_me(x):
    print(x)
```

```
df.apply(print_me, axis = 1)
```

```
kor      100.0
eng       90.0
math      95.0
Name: Aiden, dtype: float64
kor       90.0
eng       80.0
math      75.0
Name: Charles, dtype: float64
kor       95.0
eng      100.0
math      100.0
Name: Danial, dtype: float64
kor      100.0
eng      100.0
math      100.0
Name: Evan, dtype: float64
kor       NaN
eng       35.0
math      60.0
Name: Henry, dtype: float64
```

```
Out[22]:
```

```
name
Aiden      None
Charles    None
Danial      None
Evan       None
Henry      None
dtype: object
```

In [31]:

```
# 학생 별 합계 구하기
def get_sum(x):
    return x.sum()

df_copy['sum'] = df.apply(get_sum, axis = 1) # 10이 행단위
df_copy['sum']
```

Out[31]:

```
name
Aiden      570.0
Charles    490.0
Danial     590.0
Evan       600.0
Henry      190.0
sum         0.0
Name: sum, dtype: float64
```

In [32]:

```
df_copy
```

Out[32]:

	kor	eng	math	sum
name				
Aiden	100.0	90.0	95.0	570.0
Charles	90.0	80.0	75.0	490.0
Danial	95.0	100.0	100.0	590.0
Evan	100.0	100.0	100.0	600.0
Henry	NaN	35.0	60.0	190.0
sum	385.0	405.0	430.0	0.0

In [33]:

```
# 과목 별 점수 합 구하기
df.apply(get_sum, axis = 0)
```

Out[33]:

```
kor      385.0
eng      405.0
math     430.0
sum     1220.0
dtype: float64
```

In [34]:

```
df_copy.loc['sum'] = df.apply(get_sum, axis = 0)
df_copy
```

Out[34]:

	kor	eng	math	sum
name				
Aiden	100.0	90.0	95.0	570.0
Charles	90.0	80.0	75.0	490.0
Danial	95.0	100.0	100.0	590.0
Evan	100.0	100.0	100.0	600.0
Henry	NaN	35.0	60.0	190.0

sum	385.0	405.0	430.0	1220.0
	kor	eng	math	sum