

[ ] # (1) 먼저다음과같이하여, test 라는이름의 DataFrame 을만드시오.

```
import numpy as np
import pandas as pd

s = np.linspace(40,95,12)
ss = pd.DataFrame(s.reshape(4,3))
nn = pd.Series(['apple','banana','apricot','avocado'])
```

[48] test = pd.concat([nn,ss],axis=1)
test.columns=['name','Kor','Eng','Math']

[49] test

	name	Kor	Eng	Math
0	apple	40.0	45.0	50.0
1	banana	55.0	60.0	65.0
2	apricot	70.0	75.0	80.0
3	avocado	85.0	90.0	95.0

[ ] # (2) 다음과같이, melt 메소드를이용하여 long 테이블로 변환하여 test\_1 을만드시오.

[50] test\_1 = test.melt(id\_vars=['name'], var\_name='subject', value\_name='score')
test\_1

	name	subject	score
0	apple	Kor	40.0
1	banana	Kor	55.0
2	apricot	Kor	70.0
3	avocado	Kor	85.0
4	apple	Eng	45.0
5	banana	Eng	60.0
6	apricot	Eng	75.0
7	avocado	Eng	90.0
8	apple	Math	50.0
9	banana	Math	65.0
10	apricot	Math	80.0
11	avocado	Math	95.0

[ ] # (3) 다음과같이, pivot 메소드를이용하여다시 wide 테이블로변환하여 test\_1w 를만드시오.

[51] test\_1w = test\_1.pivot(index='name', columns='subject')

[52] test\_1w

	score			
	subject	Eng	Kor	Math
	name			
	apple	45.0	40.0	50.0
	apricot	75.0	70.0	80.0
	avocado	90.0	85.0	95.0

```
banana    60.0   55.0   65.0
```

✓ [53] test  
0초

	name	Kor	Eng	Math
0	apple	40.0	45.0	50.0
1	banana	55.0	60.0	65.0
2	apricot	70.0	75.0	80.0
3	avocado	85.0	90.0	95.0

✓ [54] test.columns  
0초

```
Index(['name', 'Kor', 'Eng', 'Math'], dtype='object')
```

✓ [55] test\_lw.columns # MultiIndex로 인덱스화 되어 있다.  
0초

```
MultiIndex([('score', 'Eng'),  
            ('score', 'Kor'),  
            ('score', 'Math')],  
           names=[None, 'subject'])
```

✓ [56] test.index  
0초

```
RangeIndex(start=0, stop=4, step=1)
```

✓ [57] test\_lw.index  
0초

```
Index(['apple', 'apricot', 'avocado', 'banana'], dtype='object', name='name')
```

```
[ ] # (4) 다음과같은점을살펴보고, test 와 test_lw 를비교하시오.  
# test_lw.columns : melt 했다가 다시 pivot 되면서 멀티인덱스로 변경됨  
# test_lw.index : index가 과일이름으로 되어있는 반면에 test는 0~4까지 숫자 인덱스로 이뤄짐  
# 과일이름의순서 : 과일 이름 오름차순 정렬  
# 과목명어순서 : 과목명 오름차순 정렬
```

✓ [58] test\_lw.columns = [ u[1] for u in test\_lw.columns ]

✓ [59] test\_lw  
0초

	Eng	Kor	Math
name			
apple	45.0	40.0	50.0
apricot	75.0	70.0	80.0
avocado	90.0	85.0	95.0
banana	60.0	55.0	65.0

✓ [60] test  
0초

	name	Kor	Eng	Math
0	apple	40.0	45.0	50.0
1	banana	55.0	60.0	65.0
2	apricot	70.0	75.0	80.0
3	avocado	85.0	90.0	95.0

```
[ ] # (5) 다음과같이하여, test_lw 에서 MultiIndex 를제거하고,  
# 그결과를 test 와비교하시오.  
# => multiIndex 는 제거 되었으나 name 속성이 함께 없음  
# => 여전히 index는 숫자인덱스가 아니라 과일이름
```

✓ [61] test\_lw.columns  
0초

```
Index(['Eng', 'Kor', 'Math'], dtype='object')
```

```
✓ [62] test_lw.index
0초
Index(['apple', 'apricot', 'avocado', 'banana'], dtype='object', name='name')
```

```
✓ [63] test.columns
0초
Index(['name', 'Kor', 'Eng', 'Math'], dtype='object')
```

```
✓ [64] test.index
0초
RangeIndex(start=0, stop=4, step=1)
```

```
[ ] # (6) 다음과같이하여, test_lw 의과목명의순서를바로잡아,
# test 와과목명의순서가동일한 test_lwk 를만드시오.
```

```
✓ [65] test.columns
0초
Index(['name', 'Kor', 'Eng', 'Math'], dtype='object')
```

```
✓ [66] test_lwk=test_lw[test.columns[1:]]
0초
test_lwk
```

	Kor	Eng	Math
name			
apple	40.0	45.0	50.0
apricot	70.0	75.0	80.0
avocado	85.0	90.0	95.0
banana	55.0	60.0	65.0

```
[ ] # (7) 다음사항을확인하여, test 와 test_lwk 의차이를비교하시오.
# => 과목명 순서는 바뀌었으나 여전히 test_lwk는 인덱스가 과목명으로 되어있고 컬럼에는 과목명 존재하지 않음
```

```
✓ [67] test.index
0초
RangeIndex(start=0, stop=4, step=1)
```

```
✓ [68] test_lwk.index
0초
Index(['apple', 'apricot', 'avocado', 'banana'], dtype='object', name='name')
```

```
✓ [69] test.columns
0초
Index(['name', 'Kor', 'Eng', 'Math'], dtype='object')
```

```
✓ [70] test_lwk.columns
0초
Index(['Kor', 'Eng', 'Math'], dtype='object')
```

```
[ ] # (8) 다음과같이하여연계되는 test_A 와 test 를비교하시오.
# => 컬럼과 인덱스 모두 동일하다
# => 과일이름 순서는 test_A 여전히 오름차순
```

```
✓ [71] test_A = test_lwk.reset_index()
```

```
✓ [72] test_A
0초
```

	name	Kor	Eng	Math
0	apple	40.0	45.0	50.0
1	apricot	70.0	75.0	80.0
2	avocado	85.0	90.0	95.0
3	banana	55.0	60.0	65.0

```
✓ [73] test
```

	name	Kor	Eng	Math
0	apple	40.0	45.0	50.0
1	banana	55.0	60.0	65.0
2	apricot	70.0	75.0	80.0
3	avocado	85.0	90.0	95.0

✓ [26] test\_A.columns  
0초  
Index(['name', 'Kor', 'Eng', 'Math'], dtype='object')

✓ [27] test\_A.index  
0초  
RangeIndex(start=0, stop=4, step=1)

✓ [28] test.columns  
0초  
Index(['name', 'Kor', 'Eng', 'Math'], dtype='object')

✓ [29] test.index  
0초  
RangeIndex(start=0, stop=4, step=1)

[ ] # (9) test\_A 의과일이름순서를 test 와동일하게맞추고자한다.  
# 이를위하여, 다음과같이 test[['name']] 과 test\_A 에  
# merge 메소드를적용하려한다.  
# 다음파이썬문장에서 ??? 에적당한내용을넣어,  
# 명령을완성하고, 그결과를얻고, test 와비교하시오.

✓ [74] test[['name']].merge(test\_A, on='name')  
0초

	name	Kor	Eng	Math
0	apple	40.0	45.0	50.0
1	banana	55.0	60.0	65.0
2	apricot	70.0	75.0	80.0
3	avocado	85.0	90.0	95.0

