파이썬 머신러닝 판다스 데이터분석

Lecture (6)

Dr. Heesuk Kim

목차

Part 1. 판다스 입문

2-2. 데이터프레임

• 행 선택

1) **loc**과 **iloc** 인덱서를 사용.

2) 인덱스 이름을 기준으로 행을 선택할 때는 loc을 이용하고, 정수형 위치 인덱스를 사용할 때는 iloc을 이용.

구 분	loc	iloc
탐색 대상	인덱스 이름(index label)	정수형 위치 인덱스(integer position)
범위 지정	가능(범위의 끝 포함) 예) ['a':'c'] → 'a',' b', 'c'	가능(범위의 끝 제외) 예) [3:7] → 3, 4, 5, 6 (* 7 제외)

[표 1-2] loc과 iloc

예제 1-9

① 1개의 행 선택

〈예제 1-9〉 행 선택 (File: example/part1/1.9_select_row.py)

```
1  # -*- coding: utf-8 -*-
2
3  import pandas as pd
4
5  # DataFrame() 함수로 데이터프레임 변환. 변수 df에 저장
6  exam_data = {'수학' : [ 90, 80, 70], '영어' : [ 98, 89, 95],
7               '음악' : [ 85, 95, 100], '체육' : [ 100, 90, 90]}
8
9  df = pd.DataFrame(exam_data, index=['서준', '우현', '인아'])
10 print(df)
11 print('\n')
12
13 # 행 인덱스를 사용하여 행 1개 선택
14 label1 = df.loc['서준']
15 position1 = df.iloc[0]
16 print(label1)
17 print('\n')
18 print(position1)
```

〈실행 결과〉 코드 1~18라인을 부분 실행

```
     수학  영어   음악   체육
서준  90   98    85   100
우현  80   89    95    90
인아  70   95   100    90


수학     90
영어     98
음악     85
체육    100
Name: 서준, dtype: int64


수학     90
영어     98
음악     85
체육    100
Name: 서준, dtype: int64
```

데이터프레임의 첫 번째 행에는 '서준' 학생의 과목별 점수 데이터가 입력되어 있다. '서준' 학생의 과목별 점수 데이터를 행으로 추출하면 시리즈 객체가 반환된다.

loc 인덱서를 이용하려면 '서준'이라는 인덱스 이름을 직접 입력하고, iloc을 이용할 때는 첫 번째 정수형 위치를 나타내는 0을 입력한다. 각각 반환되는 값을 label1 변수와 position1 변수에 저장, 출력하면 같은 결과를 갖는다.

2-2. 데이터프레임

② 여러 개의 행을 선택(인덱스 리스트 활용)

〈예제 1-9〉 행 선택 (File: example/part1/1.9_select_row.py(이어서 계속))

```
~  ~~~ (생략) ~~~

21  # 행 인덱스를 사용하여 2개 이상의 행 선택
22  label2 = df.loc[['서준', '우현']]
23  position2 = df.iloc[[0, 1]]
24  print(label2)
25  print('\n')
26  print(position2)
```

〈실행 결과〉 코드 21~26라인을 부분 실행

```
     수학  영어  음악  체육
서준  90   98   85   100
우현  80   89   95    90
     수학  영어  음악  체육
서준  90   98   85   100
우현  80   89   95    90
```

2개 이상의 행 인덱스를 배열로 입력하면, 매칭되는 모든 행 데이터를 동시에 추출한다.

데이터프레임 df의 첫번째와 두번째 행에 있는 '서준', '우현' 학생을 인덱싱으로 선택해 본다. loc 인덱서는 ['서준', '우현'] 과 같이 인덱스 이름을 배열로 전달하고, iloc을 이용할 때는 [0, 1] 과 같이 정수형 위치를 전달한다. 이때, label2 변수와 position2 변수에 저장된 값은 같다.

2-2. 데이터프레임

③ 여러 개의 원소를 선택(인덱스 범위 지정)

〈예제 1-9〉 행 선택 (File: example/part1/1.9_select_row.py(이어서 계속))

```
 ~  ~~~ (생략) ~~~

29  # 행 인덱스의 범위를 지정하여 행 선택
30  label3 = df.loc['서준':'우현']
31  position3 = df.iloc[0:1]
32  print(label3)
33  print('\n')
34  print(position3)
```

〈실행 결과〉 코드 29~34라인을 부분 실행

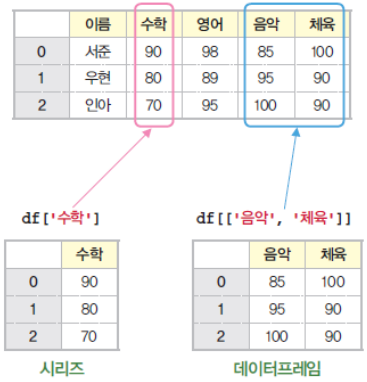
```
      수학  영어  음악  체육
서준   90    98    85   100
우현   80    89    95    90


      수학  영어  음악  체육
서준   90    98    85   100
```

단, 인덱스 이름을 범위로 지정한 label3의 경우에는 범위의 마지막 값인 '우현' 학생의 점수가 포함되지만,

정수형 위치 인덱스를 사용한 position3에는 범위의 마지막 값인 '우현' 학생의 점수가 제외된다.

2-2. 데이터프레임

- **열 선택**



[그림 1-11] 데이터프레임의 열 선택

예제 1-10

① 1개의 열 선택

열 1개 선택(시리즈 생성): DataFrame 객체["열 이름"] 또는 DataFrame 객체.열 이름

〈예제 1-10〉 열 선택 (File: example/part1/1.10_select_column.py)

```
1  # -*- coding: utf-8 -*-
2
3  import pandas as pd
4
5  # DataFrame() 함수로 데이터프레임 변환. 변수 df에 저장
6  exam_data = {'이름' : [ '서준', '우현', '인아'],
7               '수학' : [ 90, 80, 70],
8               '영어' : [ 98, 89, 95],
9               '음악' : [ 85, 95, 100],
10              '체육' : [ 100, 90, 90]}
11 df = pd.DataFrame(exam_data)
12 print(df)
13 print(type(df))
14 print('\n')
15
16 # '수학' 점수 데이터만 선택. 변수 math1에 저장
17 math1 = df['수학']
18 print(math1)
19 print(type(math1))
20 print('\n')
21
22 # '영어' 점수 데이터만 선택. 변수 english에 저장
23 english = df.영어
24 print(english)
25 print(type(english))
```

〈실행 결과〉 코드 1~25라인을 부분 실행

```
   이름  수학  영어  음악  체육
0  서준   90   98   85  100
1  우현   80   89   95   90
2  인아   70   95  100   90
<class 'pandas.core.frame.DataFrame'>


0    90
1    80
2    70
Name: 수학, dtype: int64
<class 'pandas.core.series.Series'>


0    98
1    89
2    95
Name: 영어, dtype: int64
<class 'pandas.core.series.Series'>
```

type( ) 함수를 사용하여, 데이터프레임에서 1개의 열을 선택할 때 반환되는 객체의 자료형을 확인하면 시리즈이다.

2-2. 데이터프레임

② n개의 열 선택 (리스트 입력)

열 n개 선택(데이터프레임 생성): `DataFrame 객체[ [ 열1, 열2 , … , 열n ] ]`

〈예제 1-10〉 열 선택 (File: example/part1/1.10_select_column.py(이어서 계속))

```
 ~  ~~~ (생략) ~~~

28  # '음악', '체육' 점수 데이터를 선택. 변수 music_gym에 저장
29  music_gym = df[['음악', '체육']]
30  print(music_gym)
31  print(type(music_gym))
32  print('\n')
33
34  # '수학' 점수 데이터만 선택. 변수 math2에 저장
35  math2 = df[['수학']]
36  print(math2)
37  print(type(math2))
```

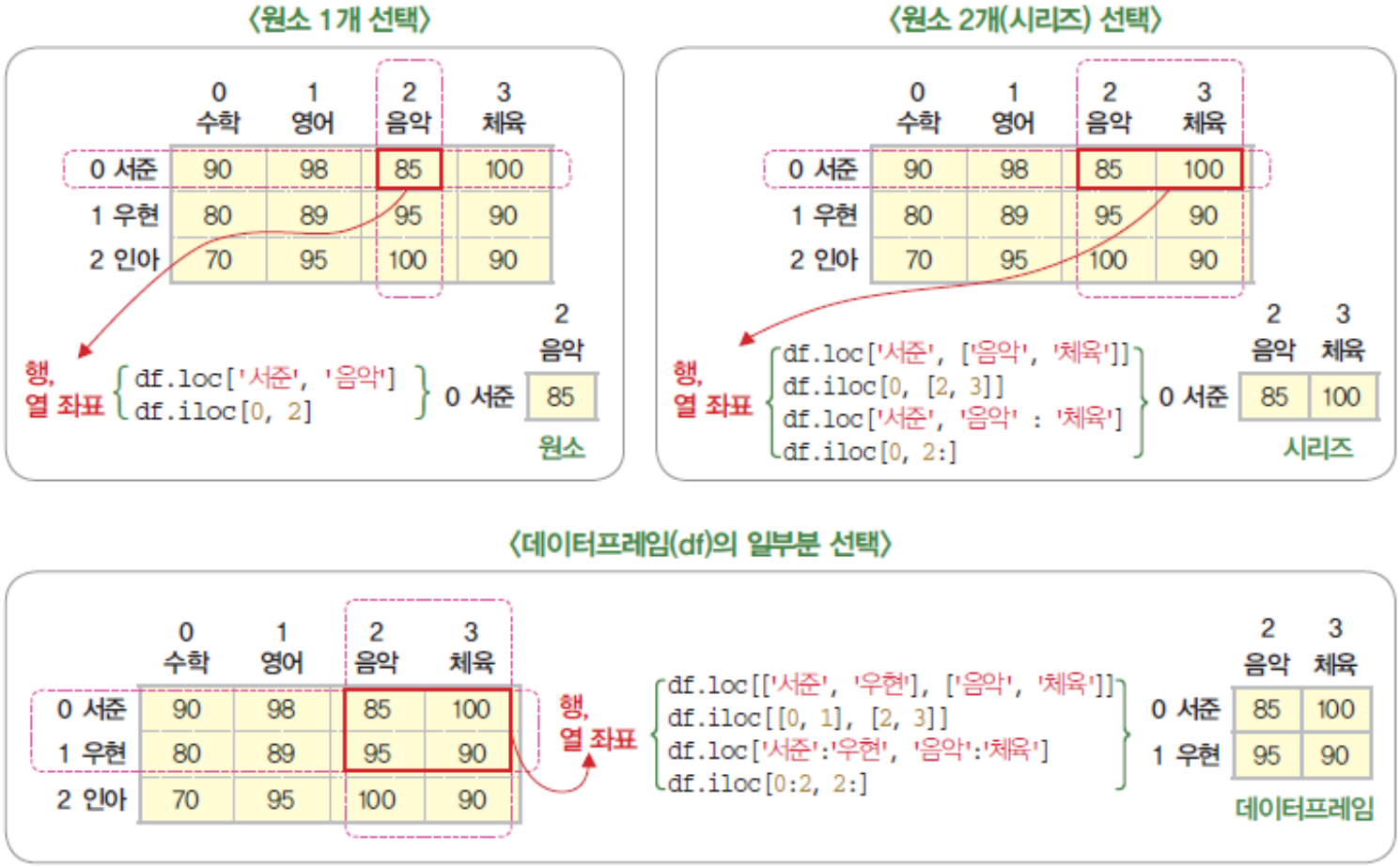
〈실행 결과〉 코드 28~37라인을 부분 실행

```
   음악  체육
0   85  100
1   95   90
2  100   90
<class 'pandas.core.frame.DataFrame'>


   수학
0  90
1  80
2  70
<class 'pandas.core.frame.DataFrame'>
```

이 때, 반환되는 객체의 자료형을 확인하면 데이터프레임이다.

2-2. 데이터프레임

- **원소 선택**



[그림 1-12] 데이터프레임의 [행, 열] 데이터 선택

2-2. 데이터프레임

예제 1-11

〈예제 1-11〉 원소 선택 (File: example/part1/1.11_select_element.py)

```
1  # -*- coding: utf-8 -*-
2
3  import pandas as pd
4
5  # DataFrame() 함수로 데이터프레임 변환. 변수 df에 저장
6  exam_data = {'이름' : [ '서준', '우현', '인아'],
7               '수학' : [ 90, 80, 70],
8               '영어' : [ 98, 89, 95],
9               '음악' : [ 85, 95, 100],
10              '체육' : [ 100, 90, 90]}
11 df = pd.DataFrame(exam_data)
12
13 # '이름' 열을 새로운 인덱스로 지정하고, df 객체에 변경 사항 반영
14 df.set_index('이름', inplace=True)
15 print(df)
```

```
  이름  수학  영어   음악   체육
0  서준  90  98   85  100
1  우현  80  89   95   90
2   인아  70  95  100   90
```

```
      수학  영어   음악   체육
이름
서준  90    98     85    100
우현  80    89     95     90
인아  70    95    100     90
```

① 1개의 원소를 선택

〈예제 1-11〉 원소 선택 (File: example/part1/1.11_select_element.py(이어서 계속))

```
~  ~~~ 생략 ~~~

18 # 데이터프레임 df의 특정 원소 1개 선택('서준'의 '음악' 점수)
19 a = df.loc['서준', '음악']
20 print(a)
21 b = df.iloc[0, 2]
22 print(b)
```

〈실행 결과〉 코드 18~22라인을 부분 실행

```
85
85
```

2-2. 데이터프레임

② 2개 이상의 원소를 선택 (시리즈)

〈예제 1-11〉 원소 선택 (File: example/part1/1.11_select_element.py(이어서 계속))

```
 ~  ~~~ 생략 ~~~

25  # 데이터프레임 df의 특정 원소 2개 이상 선택('서준'의 '음악', '체육' 점수)
26  c = df.loc['서준', ['음악', '체육']]
27  print(c)
28  d = df.iloc[0, [2, 3]]
29  print(d)
30  e = df.loc['서준', '음악':'체육']
31  print(e)
32  f = df.iloc[0, 2:]
33  print(f)
```

〈실행 결과〉 코드 25~33라인을 부분 실행

```
음악     85
체육    100
Name: 서준, dtype: int64
음악     85
체육    100
Name: 서준, dtype: int64
음악     85
체육    100
Name: 서준, dtype: int64
음악     85
체육    100
Name: 서준, dtype: int64
```

③ 2개 이상의 원소를 선택 (데이터프레임)

〈예제 1-11〉 원소 선택 (File: example/part1/1.11_select_element.py(이어서 계속))

```
 ~  ~~~ 생략 ~~~

36  # df 2개 이상의 행과 열에 속하는 원소들 선택('서준', '우현'의 '음악', '체육' 점수)
37  g = df.loc[['서준', '우현'], ['음악', '체육']]
38  print(g)
39  h = df.iloc[[0, 1], [2, 3]]
40  print(h)
41  i = df.loc['서준':'우현', '음악':'체육']
42  print(i)
43  j = df.iloc[0:2, 2:]
44  print(j)
```

〈실행 결과〉 코드 36~44라인을 부분 실행

```
     음악   체육
이름
서준  85  100
우현  95   90
     음악   체육
이름
서준  85  100
우현  95   90
     음악   체육
이름
서준  85  100
우현  95   90
     음악   체육
이름
서준  85  100
우현  95   90
```

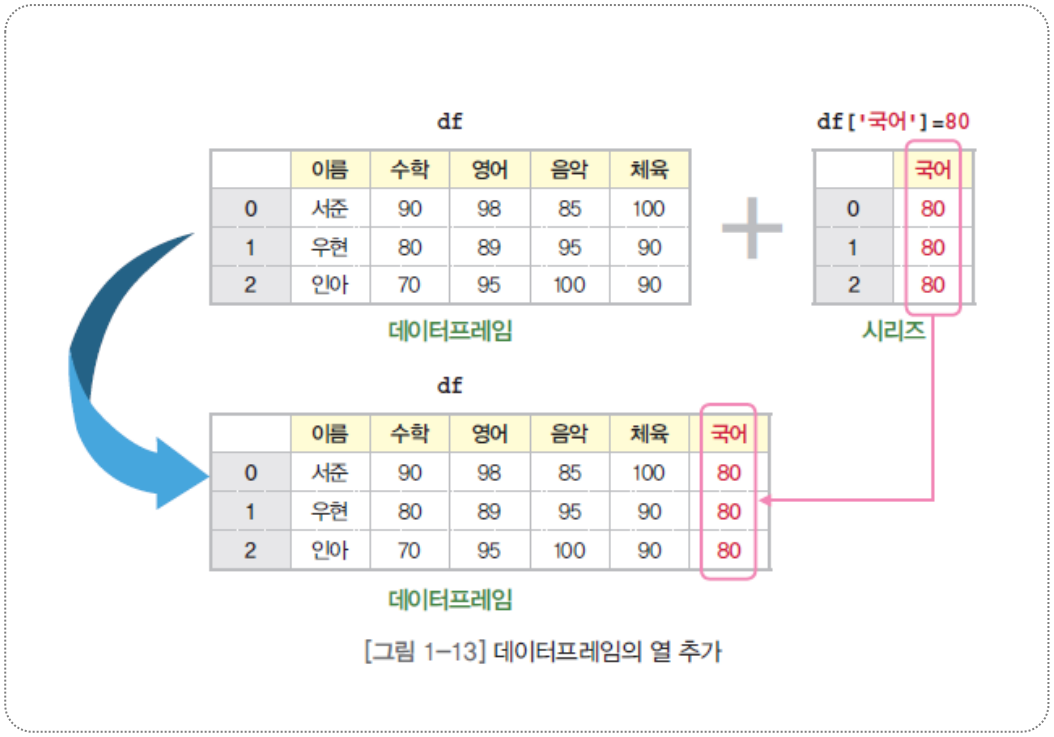
2-2. 데이터프레임

- **열 추가**

1) 추가하려는 열 이름과 데이터 값을 입력. 마지막 열에 덧붙이듯 새로운 열을 추가.

```
열 추가: DataFrame 객체[ '추가하려는 열 이름' ] = 데이터 값
```

2) 이때 모든 행에 동일한 값이 입력되는 점에 유의.



	이름	수학	영어	음악	체육
0	서준	90	98	85	100
1	우현	80	89	95	90
2	인아	70	95	100	90

	국어
0	80
1	80
2	80

	이름	수학	영어	음악	체육	국어
0	서준	90	98	85	100	80
1	우현	80	89	95	90	80
2	인아	70	95	100	90	80

[그림 1-13] 데이터프레임의 열 추가

2-2. 데이터프레임

예제 1-12

다음 예제에서 '국어' 열을 새로 추가하는데, 모든 학생들의 국어 점수가 동일하게 80점으로 입력되는 과정을 보여준다.

〈예제 1-12〉 열 추가 (File: example/part1/1.12_add_column.py)

```
1  # -*- coding: utf-8 -*-
2
3  import pandas as pd
4
5  # DataFrame() 함수로 데이터프레임 변환. 변수 df에 저장
6  exam_data = {'이름' : [ '서준', '우현', '인아'],
7               '수학' : [ 90, 80, 70],
8               '영어' : [ 98, 89, 95],
9               '음악' : [ 85, 95, 100],
10              '체육' : [ 100, 90, 90]}
11 df = pd.DataFrame(exam_data)
12 print(df)
13 print('\n')
14
15 # 데이터프레임 df에 '국어' 점수 열(column) 추가. 데이터 값은 80 지정
16 df['국어'] = 80
17 print(df)
```

〈실행 결과〉 코드 전부 실행

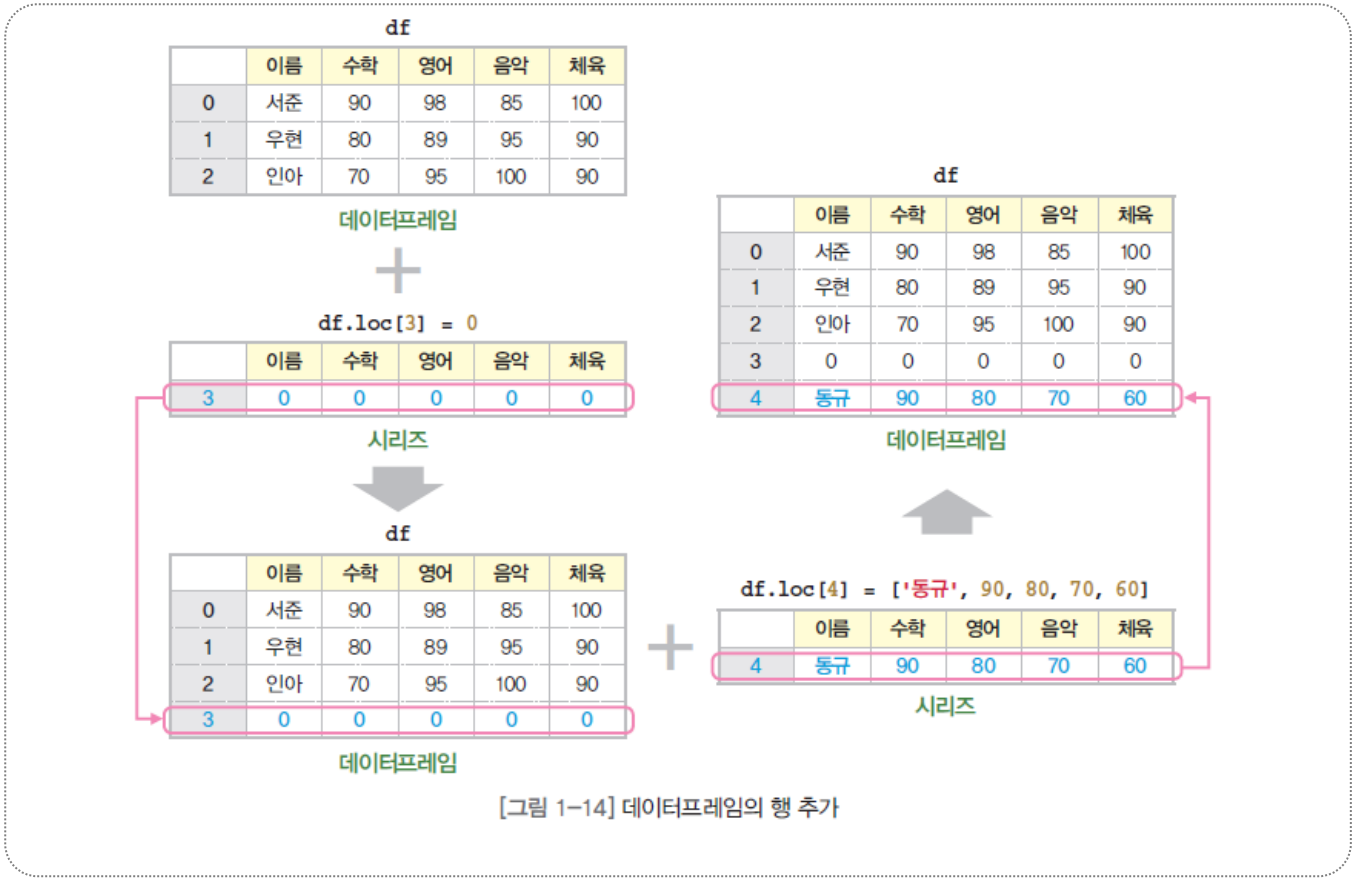
```
   이름  수학  영어  음악  체육
0  서준    90    98    85   100
1  우현    80    89    95    90
2  인아    70    95   100    90


   이름  수학  영어  음악  체육  국어
0  서준    90    98    85   100    80
1  우현    80    89    95    90    80
2  인아    70    95   100    90    80
```

2-2. 데이터프레임

- **행 추가**

: 행 인덱스와 데이터 값을 loc 인덱서를 사용하여 입력.

```
행 추가: DataFrame.loc[ '새로운 행 이름' ] = 데이터 값 (또는 배열)
```



[그림 1-14] 데이터프레임의 행 추가

2-2. 데이터프레임

예제 1-13

〈예제 1-13〉 행 추가 (File: example/part1/1.13_add_row.py)

```
1  # -*- coding: utf-8 -*-
2
3  import pandas as pd
4
5  # DataFrame() 함수로 데이터프레임 변환. 변수 df에 저장
6  exam_data = {'이름' : ['서준', '우현', '인아'],
7               '수학' : [ 90, 80, 70],
8               '영어' : [ 98, 89, 95],
9               '음악' : [ 85, 95, 100],
10              '체육' : [ 100, 90, 90]}
11 df = pd.DataFrame(exam_data)
12 print(df)
13 print('\n')
14
15 # 새로운 행(row) 추가 - 같은 원소 값 입력
16 df.loc[3] = 0
17 print(df)
18 print('\n')
19
20 # 새로운 행(row) 추가 - 원소 값 여러 개의 배열 입력
21 df.loc[4] = ['동규', 90, 80, 70, 60]
22 print(df)
23 print('\n')
24
25 # 새로운 행(row) 추가 - 기존 행 복사
26 df.loc['행5'] = df.loc[3]
27 print(df)
```

〈실행 결과〉 코드 전부 실행

```
   이름  수학  영어   음악   체육
0  서준  90  98   85  100
1  우현  80  89   95   90
2  인아  70  95  100   90

   이름  수학  영어   음악   체육
0  서준  90  98   85  100
1  우현  80  89   95   90
2  인아  70  95  100   90
3   0   0   0    0    0   ←

   이름  수학  영어   음악   체육
0  서준  90  98   85  100
1  우현  80  89   95   90
2  인아  70  95  100   90
3   0   0   0    0    0
4  동규  90  80   70   60   ←

   이름  수학  영어   음악   체육
0  서준  90  98   85  100
1  우현  80  89   95   90
2  인아  70  95  100   90
3   0   0   0    0    0
4  동규  90  80   70   60
행5  0   0   0    0    0   ←
```

2-2. 데이터프레임

- **원소 값 변경**

: 원소를 선택하고 새로운 데이터 값을 지정.

원소 값 변경: DataFrame 객체의 일부분 또는 원소를 선택 = 새로운 값

① 1개의 원소를 변경 예제 1-14

〈예제 1-14〉 원소 값 변경 (File: example/part1/1.14_modify_dataframe_element.py)

```
1  # -*- coding: utf-8 -*-
2
3  import pandas as pd
4
5  # DataFrame() 함수로 데이터프레임 변환. 변수 df에 저장
6  exam_data = {'이름' : [ '서준', '우현', '인아'],
7               '수학' : [ 90, 80, 70],
8               '영어' : [ 98, 89, 95],
9               '음악' : [ 85, 95, 100],
10              '체육' : [ 100, 90, 90]}
11 df = pd.DataFrame(exam_data)
12
13 # '이름' 열을 새로운 인덱스로 지정하고, df 객체에 변경사항 반영
14 df.set_index('이름', inplace=True)
15 print(df)
16 print('\n')
17
18 # 데이터프레임 df의 특정 원소를 변경하는 방법: '서준'의 '체육' 점수
19 df.iloc[0][3] = 80
20 print(df)
21 print('\n')
22
23 df.loc['서준']['체육'] = 90
24 print(df)
25 print('\n')
26
27 df.loc['서준', '체육'] = 100
28 print(df)
```

〈실행 결과〉 코드 1~28라인을 부분 실행

```
     수학  영어  음악  체육
이름
서준  90  98  85  100
우현  80  89  95   90
인아  70  95  100   90


     수학  영어  음악  체육
이름
서준  90  98  85   80
우현  80  89  95   90
인아  70  95  100   90


     수학  영어  음악  체육
이름
서준  90  98  85   90
우현  80  89  95   90
인아  70  95  100   90


     수학  영어  음악  체육
이름
서준  90  98  85  100
우현  80  89  95   90
인아  70  95  100   90
```

앞의 예제에서 '서준' 학생의 '체육' 점수를 선택하는 여러 방법을 시도하였다. 각 방법을 비교하기 위해, 각기 다른 점수를 새로운 값으로 입력하여 원소를 변경하였다. (변경된 값을 원으로 표시)

2-2. 데이터프레임

② 1개 이상의 원소를 변경

〈예제 1-14〉 원소 값 변경 (File: example/part1/1.14_modify_dataframe_element.py(이어서 계속))

```
~  ~~~ (생략) ~~~

31  # 데이터프레임 df의 원소 여러 개를 변경하는 방법: '서준'의 '음악', '체육' 점수
32  df.loc['서준', ['음악', '체육']] = 50
33  print(df)
34  print('\n')
35
36  df.loc['서준', ['음악', '체육']] = 100, 50
37  print(df)
```

〈실행 결과〉 코드 31~37라인을 부분 실행

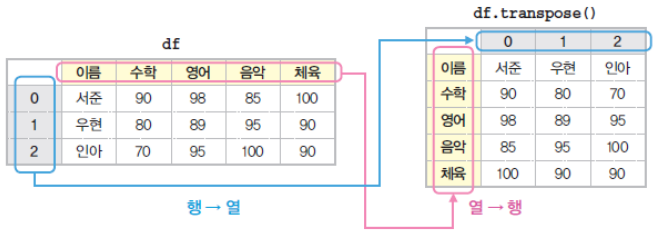
```
      수학  영어  음악  체육
이름
서준  90    98    50    50
우현  80    89    95    90
인아  70    95    100   90


      수학  영어  음악  체육
이름
서준  90    98    100   50
우현  80    89    95    90
인아  70    95    100   90
```

2-2. 데이터프레임

- **행, 열의 위치 바꾸기**

  데이터프레임의 행과 열을 서로 맞바꾸는 방법이다. 전치의 결과로 새로운 객체를 반환하므로, 기존 객체를 변경하기 위해서는 df = df.transpose() 또는 df = df.T 와 같이 입력한다.

```
행, 열 바꾸기: DataFrame 객체.transpose() 또는 DataFrame 객체.T
```



	이름	수학	영어	음악	체육
0	서준	90	98	85	100
1	우현	80	89	95	90
2	인아	70	95	100	90

	0	1	2
이름	서준	우현	인아
수학	90	80	70
영어	98	89	95
음악	85	95	100
체육	100	90	90

[그림 1-15] 행, 열 바꾸기

2-2. 데이터프레임

예제 1-15

〈예제 1-15〉 행, 열 바꾸기 (File: example/part1/1.15_transpose.py)

```
1  # -*- coding: utf-8 -*-
2
3  import pandas as pd
4
5  # DataFrame() 함수로 데이터프레임 변환. 변수 df에 저장
6  exam_data = {'이름' : [ '서준', '우현', '인아'],
7               '수학' : [ 90, 80, 70],
8               '영어' : [ 98, 89, 95],
9               '음악' : [ 85, 95, 100],
10              '체육' : [ 100, 90, 90]}
11 df = pd.DataFrame(exam_data)
12 print(df)
13 print('\n')
14
15 # 데이터프레임 df를 전치하기(메소드 활용)
16 df = df.transpose()
17 print(df)
18 print('\n')
19
20 # 데이터프레임 df를 다시 전치하기(클래스 속성 활용)
21 df = df.T
22 print(df)
```

〈실행 결과〉 코드 전부 실행

```
   이름  수학  영어   음악   체육
0  서준  90  98   85  100
1  우현  80  89   95   90
2  인아  70  95  100   90


      0   1    2
이름  서준  우현  인아
수학   90  80   70
영어   98  89   95
음악   85  95  100
체육  100  90   90


   이름  수학  영어   음악   체육
0  서준  90  98   85  100
1  우현  80  89   95   90
2  인아  70  95  100   90
```

Homework

1. 예제 1-9 을 스파이더에서 실습 후 화면 캡처
2. 예제 1-14 를 스파이더에서 실습 후 화면 캡처
3. 예제 1-15 를 스파이더에서 실습 후 화면 캡처

- **위의 화면 캡처 내용 3개를 hwp 파일에 넣어서 제출하세요.**
- 파일명 : **학과_Lecture(6)과제_학번_성명.hwp**

  예시) AI융합학과_Lecture(6)과제_230000_홍길동.hwp

과제 파일 제출 메일 주소 : idistina@daum.net

메일 제목 : **학과_Lecture(6)과제_학번_성명**
예시) AI융합학과_Lecture(6)과제_230000_홍길동

메일 내용 : hwp 파일 첨부

Any Question?

idishskim@naver.com

Thank you.