

# 파이썬 머신러닝 판다스 데이터분석

## Lecture (5)



Dr. Heesuk Kim

Part 0. 개발환경 준비

**Part 1. 판다스 입문**

Part 2. 데이터 입출력

Part 3. 데이터 살펴보기

Part 4. 시각화 도구

Part 5. 데이터 사전처리

Part 6. 데이터프레임의 다양한 응용

Part 7. 머신러닝 데이터 분석



# Part 1. 판다스 입문

---

- 1. 데이터과학자가 판다스를 배우는 이유
- 2. 판다스 자료구조
  - 2-1. 시리즈
  - 2-2. 데이터프레임
- 3. 인덱스 활용
- 4. 산술 연산
  - 4-1. 시리즈 연산
  - 4-2. 데이터프레임 연산

# Part 1. 판다스 입문

## 2-2. 데이터프레임

### • 행 인덱스/열 이름 설정

: 데이터프레임의 행 인덱스와 열 이름을 사용자가 지정 가능.

행 인덱스/열 이름 설정: `pandas.DataFrame( 2차원 배열,  
index=행 인덱스 배열,  
columns=열 이름 배열 )`

#### 예제 1-5

##### ① 데이터프레임을 만들 때 설정

'3개의 원소를 갖는 리스트' 2개를 대상으로  
2차원 배열(리스트)로 데이터프레임을 만든다.

- index 옵션: ['준서', '예은'] 배열을 지정
- columns 옵션: ['나이', '성별', '학교'] 배열을 지정

〈예제 1-5〉 행 인덱스/열 이름 설정

(File: example/part1/1.5\_change\_df\_idx\_col.py)

```
1  # -*- coding: utf-8 -*-
2
3  import pandas as pd
4
5  # 행 인덱스/열 이름 지정하여 데이터프레임 만들기
6  df = pd.DataFrame([[15, '남', '덕영중'], [17, '여', '수리중']],
7                    index=['준서', '예은'],
8                    columns=['나이', '성별', '학교'])
9
10 # 행 인덱스, 열 이름 확인하기
11 print(df)           # 데이터프레임
12 print('\n')
13 print(df.index)     # 행 인덱스
14 print('\n')
15 print(df.columns)   # 열 이름
```

〈실행 결과〉 코드 1~15라인을 부분 실행

```
나이 성별 학교
준서  15  남  덕영중
예은  17  여  수리중

Index(['준서', '예은'], dtype='object')

Index(['나이', '성별', '학교'], dtype='object')
```



# Part 1. 판다스 입문

## 2-2. 데이터프레임

### ② 속성을 지정하여 변경하기

데이터프레임 df의 행 인덱스 배열을 나타내는 df.index와 열 이름 배열을 나타내는 df.columns에 새로운 배열을 할당하는 방식으로, 행 인덱스와 열 이름을 변경할 수 있다.

- 행 인덱스 변경: DataFrame 객체.index = 새로운 행 인덱스 배열
- 열 이름 변경: DataFrame 객체.columns = 새로운 열 이름 배열

<예제 1-5> 행 인덱스/열 이름 설정 (File: example/part1/1.5\_change\_df\_idx\_col.py(이어서 계속))

~~~~~ 생략 ~~~~~

```
18 # 행 인덱스, 열 이름 변경하기
19 df.index=['학생1', '학생2']
20 df.columns=['연령', '남녀', '소속']
21
22 print(df)          # 데이터프레임
23 print('\n')
24 print(df.index)    # 행 인덱스
25 print('\n')
26 print(df.columns)  # 열 이름
```

<실행 결과> 코드 1~15라인을 부분 실행

```
나이 성별 학교
준서  15  남  덕영중
예은  17  여  수리중

Index(['준서', '예은'], dtype='object')

Index(['나이', '성별', '학교'], dtype='object')
```

<실행 결과> 코드 18~26라인을 부분 실행

```
연령  남녀  소속
학생1  15   남  덕영중
학생2  17   여  수리중

Index(['학생1', '학생2'], dtype='object')

Index(['연령', '남녀', '소속'], dtype='object')
```



# Part 1. 판다스 입문

## 2-2. 데이터프레임

예제 1-5

The screenshot shows the Spyder Python IDE interface. The main editor window displays a Python script for creating and modifying a DataFrame. The script includes comments in Korean and uses the pandas library. The Variable explorer on the right shows the DataFrame 'df' with its structure. The IPython console at the bottom shows the execution of the script, including the creation of the DataFrame and the modification of its index and columns.

```
1 #-*- coding: utf-8 -*-
2
3 import pandas as pd
4
5 # 행 인덱스/열 이름 지정하여, 데이터프레임 만들기
6 df = pd.DataFrame([[15, '남', '덕영중'], [17, '여', '수리중']],
7                   index=['준서', '예은'],
8                   columns=['나이', '성별', '학교'])
9
10 # 행 인덱스, 열 이름 확인하기
11 print(df)           #데이터프레임
12 print('\n')
13 print(df.index)     #행 인덱스
14 print('\n')
15 print(df.columns)   #열 이름
16 print('\n')
17
18 # 행 인덱스, 열 이름 변경하기
19 df.index=['학생1', '학생2']
20 df.columns=['연령', '남녀', '소속']
21
22 print(df)           #데이터프레임
23 print('\n')
24 print(df.index)     #행 인덱스
25 print('\n')
26 print(df.columns)   #열 이름
27
28 print(df)
```

Variable explorer

| Name | Type      | Size | Value |
|------|-----------|------|-------|
| df   | DataFrame | 1000 |       |

IPython console

```
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.

IPython 7.6.1 -- An enhanced Interactive Python.

In [1]:
```

IPython console | History log

Permissions: RW | End-of-lines: CRLF | Encoding: UTF-8 | Line: 28 | Column: 10 | Memory: 77%

# Part 1. 판다스 입문

## 2-2. 데이터프레임

### ③ rename 메소드 사용

rename() 메소드를 적용하면, 행 인덱스 또는 열 이름의 일부를 선택하여 변경 가능. 객체를 리턴.

\* 원본 객체를 변경하려면, `inplace=True` 옵션을 사용.

- 행 인덱스 변경: `DataFrame 객체.rename(index={기존 인덱스:새 인덱스, ... })`
- 열 이름 변경: `DataFrame 객체.rename(columns={기존 이름:새 이름, ... })`



# Part 1. 판다스 입문

## 2-2. 데이터프레임

### ③ rename 메소드 사용

#### 예제 1-6

〈예제 1-6〉 행 인덱스/열 이름 변경

(File: example/part1/1.6\_change\_df\_idx\_col2.py)

```
1  # -*- coding: utf-8 -*-
2
3  import pandas as pd
4
5  # 행 인덱스/열 이름 지정하여 데이터프레임 만들기
6  df = pd.DataFrame([[15, '남', '덕영중'], [17, '여', '수리중']],
7                     index=['준서', '예은'],
8                     columns=['나이', '성별', '학교'])
9
10 # 데이터프레임 df 출력
11 print(df)
12 print("\n")
13
14 # 열 이름 중, '나이'를 '연령'으로, '성별'을 '남녀'로, '학교'를 '소속'으로 바꾸기
15 df.rename(columns={'나이': '연령', '성별': '남녀', '학교': '소속'}, inplace=True)
16
17 # df의 행 인덱스 중에서, '준서'를 '학생1'로, '예은'을 '학생2'로 바꾸기
18 df.rename(index={'준서': '학생1', '예은': '학생2'}, inplace=True)
19
20 # df 출력 (변경 후)
21 print(df)
```

원본 객체를 변경하려면, **inplace=True** 옵션을 사용.

각 리스트가 행으로 변환되는 점에 유의한다.

〈실행 결과〉 코드 전부 실행

|    | 나이 | 성별 | 학교  |
|----|----|----|-----|
| 준서 | 15 | 남  | 덕영중 |
| 예은 | 17 | 여  | 수리중 |

|     | 연령 | 남녀 | 소속  |
|-----|----|----|-----|
| 학생1 | 15 | 남  | 덕영중 |
| 학생2 | 17 | 여  | 수리중 |

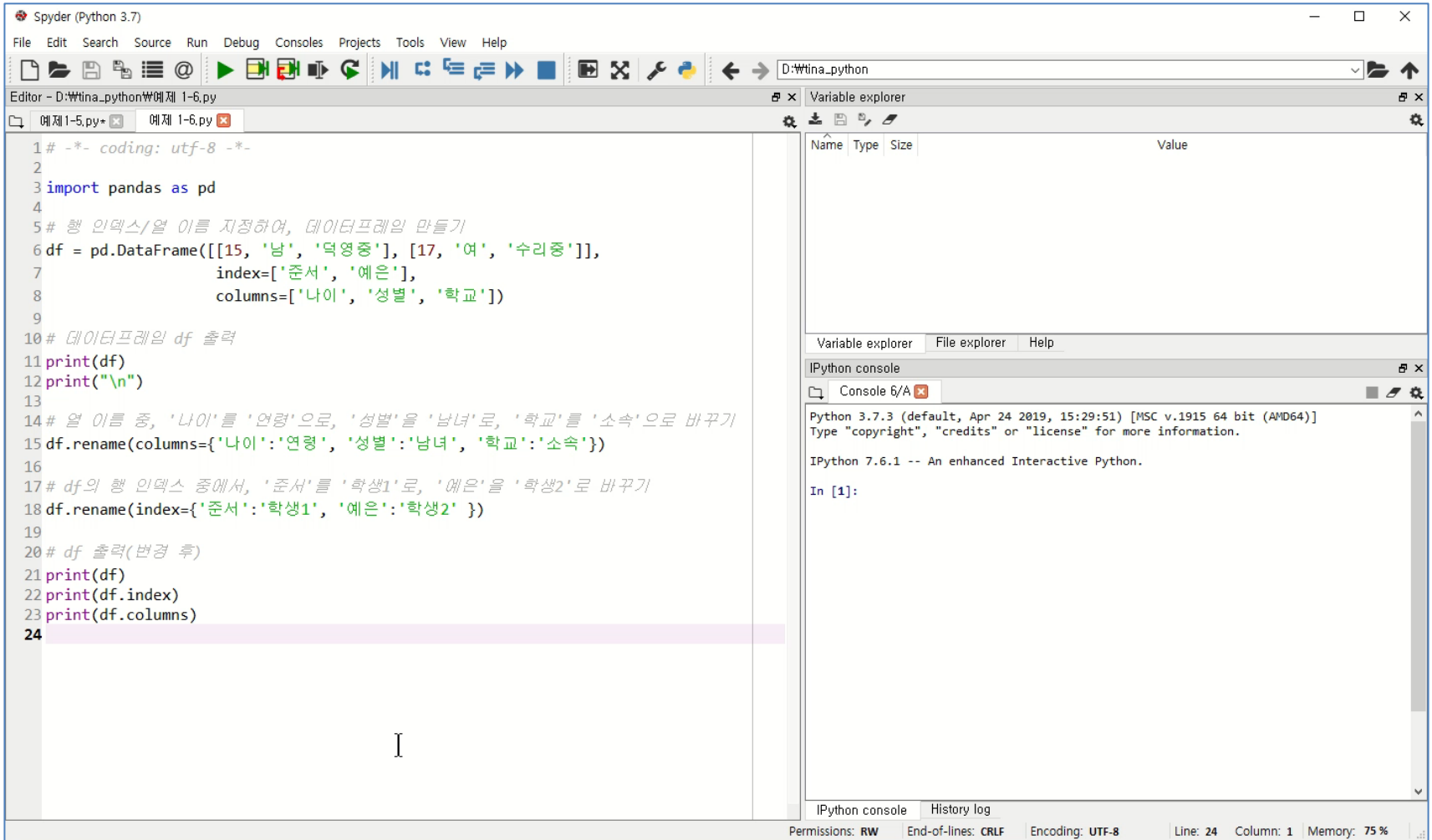




# Part 1. 판다스 입문

## 2-2. 데이터프레임

### 예제 1-6



The screenshot shows the Spyder Python IDE interface. The main editor window displays a Python script for creating and manipulating a DataFrame. The script includes comments in Korean and uses pandas to create a DataFrame with two rows and three columns. It then renames the columns and the index, and finally prints the DataFrame and its index and columns.

```
1 # -*- coding: utf-8 -*-
2
3 import pandas as pd
4
5 # 행 인덱스/열 이름 지정하여, 데이터프레임 만들기
6 df = pd.DataFrame([[15, '남', '덕영중'], [17, '여', '수리중']],
7                   index=['준서', '예은'],
8                   columns=['나이', '성별', '학교'])
9
10 # 데이터프레임 df 출력
11 print(df)
12 print("\n")
13
14 # 열 이름 중, '나이'를 '연령'으로, '성별'을 '남녀'로, '학교'를 '소속'으로 바꾸기
15 df.rename(columns={'나이': '연령', '성별': '남녀', '학교': '소속'})
16
17 # df의 행 인덱스 중에서, '준서'를 '학생1'로, '예은'을 '학생2'로 바꾸기
18 df.rename(index={'준서': '학생1', '예은': '학생2' })
19
20 # df 출력(변경 후)
21 print(df)
22 print(df.index)
23 print(df.columns)
24
```

The Variable explorer on the right shows an empty table with columns Name, Type, Size, and Value. The IPython console at the bottom shows the output of the script, including the DataFrame and its index and columns.

Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)]  
Type "copyright", "credits" or "license" for more information.

IPython 7.6.1 -- An enhanced Interactive Python.

In [1]:

IPython console History log

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 24 Column: 1 Memory: 75 %

# Part 1. 판다스 입문

## 2-2. 데이터프레임

### • 행/열 삭제

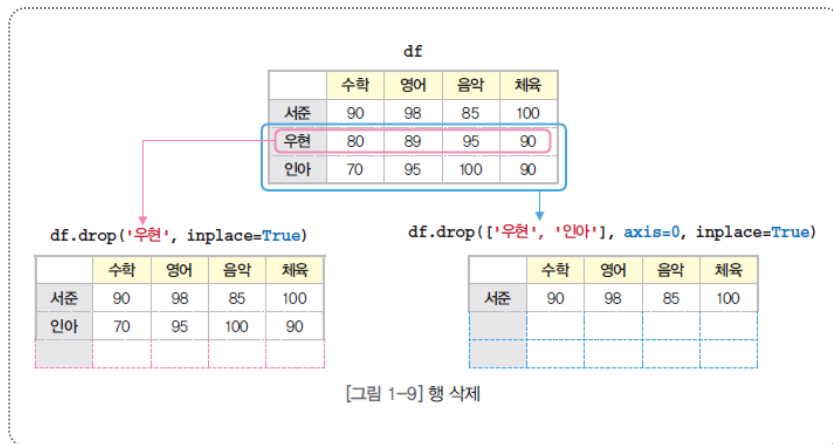
drop() 메소드에 행을 삭제할 때는 축(axis) 옵션으로 axis=0을 입력하거나, 별도로 입력하지 않는다.

반면, 축옵션으로 axis=1을 입력하면 열을 삭제한다. 동시에 여러 개의 행 또는 열을 삭제하려면, 리스트 형태로 입력한다.

- 행 삭제: DataFrame 객체.drop(행 인덱스 또는 배열, axis=0)
- 열 삭제: DataFrame 객체.drop(열 이름 또는 배열, axis=1)

한편, drop() 메소드는 기존 객체를 변경하지 않고 새로운 객체를 반환하는 점에 유의한다. 따라서, 원본 객체를 직접 변경하기 위해서는, inplace=True 옵션을 추가한다.

#### ① 행 삭제



# Part 1. 판다스 입문

## 2-2. 데이터프레임

### 예제 1-7

〈예제 1-7〉 행 삭제

(File: example/part1/1.7\_remove\_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 # 데이터프레임 df를 복제하여 변수 df2에 저장. df2의 1개 행(row) 삭제
14 df2 = df[:]
15 df2.drop('우현', inplace=True)
16 print(df2)
17 print('\n')
18
19 # 데이터프레임 df를 복제하여 변수 df3에 저장. df3의 2개 행(row) 삭제
20 df3 = df[:]
21 df3.drop(['우현', '인아'], axis=0, inplace=True)
22 print(df3)
```

〈실행 결과〉 코드 전부 실행

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

|    | 수학 | 영어 | 음악  | 체육  |
|----|----|----|-----|-----|
| 서준 | 90 | 98 | 85  | 100 |
| 인아 | 70 | 95 | 100 | 90  |

|    | 수학 | 영어 | 음악 | 체육  |
|----|----|----|----|-----|
| 서준 | 90 | 98 | 85 | 100 |



# Part 1. 판다스 입문

## 2-2. 데이터프레임

### ② 열 삭제

#### 예제 1-8

df

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

`df.drop('수학', axis=1, inplace=True)`

|    | 영어 | 음악  | 체육  |
|----|----|-----|-----|
| 서준 | 98 | 85  | 100 |
| 우현 | 89 | 95  | 90  |
| 인아 | 95 | 100 | 90  |

`df.drop(['영어', '음악'], axis=1, inplace=True)`

|    | 수학 | 체육  |
|----|----|-----|
| 서준 | 90 | 100 |
| 우현 | 80 | 90  |
| 인아 | 70 | 90  |

[그림 1-10] 열 삭제

<예제 1-8> 열 삭제

(File: example/part1/1.8\_remove\_column.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 # 데이터프레임 df를 복제하여 변수 df4에 저장. df4의 1개 열(column) 삭제
14 df4 = df[:]
15 df4.drop('수학', axis=1, inplace=True)
16 print(df4)
17 print('\n')
```

```
19 # 데이터프레임 df를 복제하여 변수 df5에 저장. df5의 2개 열(column) 삭제
20 df5 = df[:]
21 df5.drop(['영어', '음악'], axis=1, inplace=True)
22 print(df5)
```

<실행 결과> 코드 전부 실행

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

|    | 영어 | 음악  | 체육  |
|----|----|-----|-----|
| 서준 | 98 | 85  | 100 |
| 우현 | 89 | 95  | 90  |
| 인아 | 95 | 100 | 90  |

|    | 수학 | 체육  |
|----|----|-----|
| 서준 | 90 | 100 |
| 우현 | 80 | 90  |
| 인아 | 70 | 90  |



# Part 0. 개발환경 준비

## 2-2. 데이터프레임

예제 1-7

The screenshot shows the Spyder Python IDE interface. The main editor window displays a Python script for creating and manipulating a DataFrame. The script is as follows:

```
1 import pandas as pd
2
3 exam_data = {'수학' : [ 90, 80, 70], '영어' : [ 98, 89, 95],
4             '음악' : [ 85, 95, 100], '체육' : [ 100, 90, 90]}
5
6 df = pd.DataFrame(exam_data, index=['서준', '우현', '인아'])
7 print(df)
8 print('\n')
9
10
11 df2 = df[: ]
12 print(df2)
13 df2.drop('우현')
14 print(df2)
15
16 df22 = df2.drop('우현')
17 print(df22)
18 print('\n')
19 print(df2)
20
21
22
23 print('\n')
24 df3 = df[: ]
25 df3.drop(['우현', '인아'], axis=0, inplace=True)
26 print(df3)
```

The right-hand side of the IDE contains the Variable explorer, File explorer, and IPython console. The Variable explorer is currently empty. The IPython console shows the output of the script, which is the DataFrame created in line 6:

```
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.6.1 -- An enhanced Interactive Python.

In [1]:
```

The status bar at the bottom of the IDE shows the following information: Permissions: RW, End-of-lines: CRLF, Encoding: UTF-8, Line: 26, Column: 11, Memory: 83%.

# Any Questions?

Thank you.

