

# 파이썬 머신러닝 판다스 데이터분석

Lecture (11)



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 2. 데이터 입출력

---

### 1. 외부파일 읽기

1-1. CSV 파일

1-2. Excel 파일

1-3. JSON 파일

### 2. 웹(web)에서 가져오기

2-1. HTML 웹 페이지에서 표 속성 가져오기

2-2. 웹 스크래핑

### 3. API 활용하여 데이터 수집하기

### 4. 데이터 저장하기

4-1. CSV 파일로 저장

4-2. JSON 파일로 저장

4-3. Excel 파일로 저장

4-4. 여러 개의 데이터프레임을 하나의 Excel 파일로 저장



# Part 2. 데이터 입출력

## 1-2. Excel 파일

남북한발전전력량.xlsx - Excel

kim hs

파일 홈 삽입 페이지 레이아웃 수식 데이터 검토 보기 개발 도구 도움말 어떤 작업을 원하시나요? 공유

클립보드 글꼴 맞춤 표시 형식 스타일 셀 조건부 서식 표 서식 셀 스타일 삽입 삭제 서식 정렬 및 필터 찾기 및 선택

C15

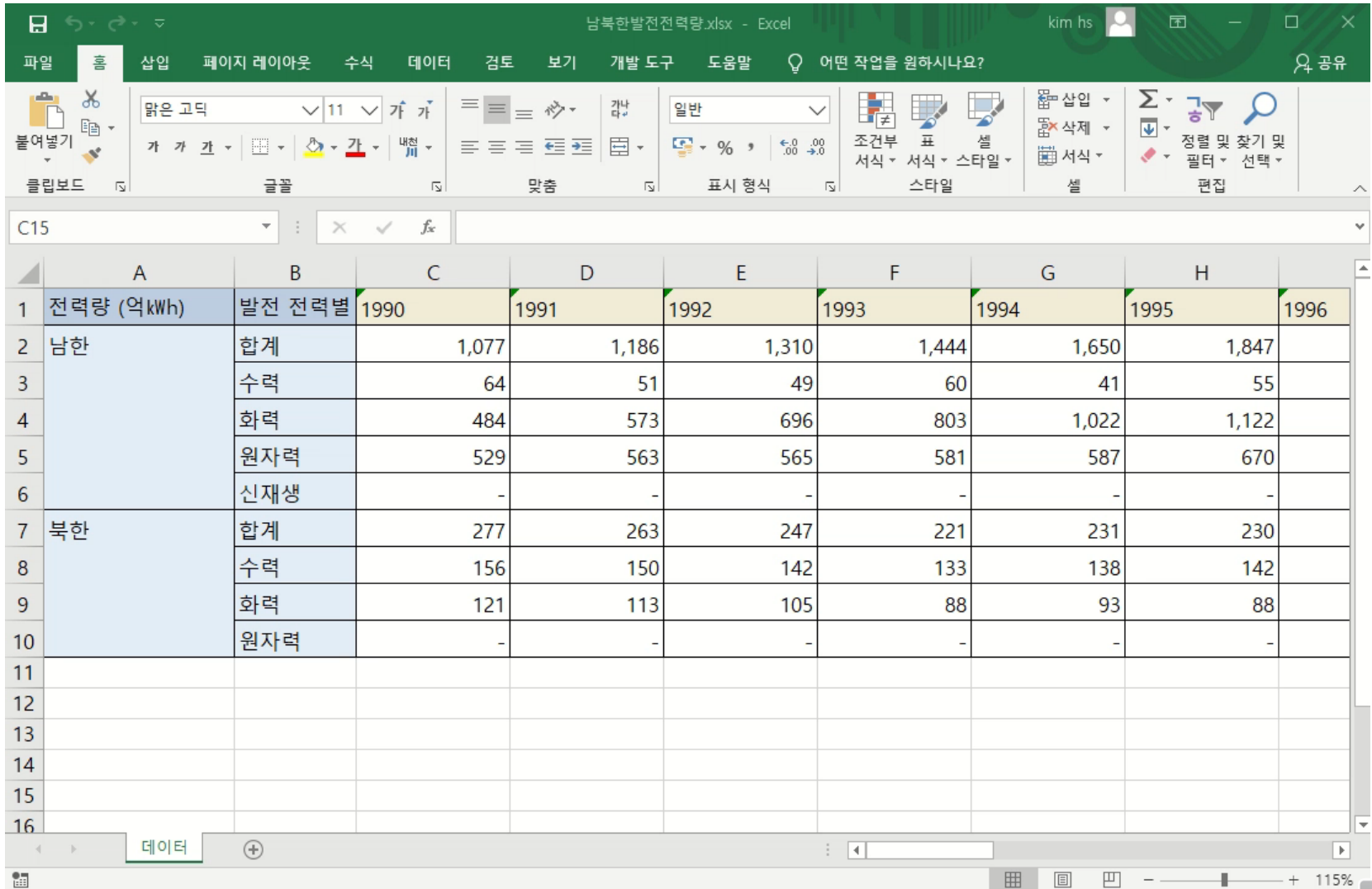
	A	B	C	D	E	F	G	H	
1	전력량 (억kWh)	발전 전력별	1990	1991	1992	1993	1994	1995	1996
2	남한	합계	1,077	1,186	1,310	1,444	1,650	1,847	
3		수력	64	51	49	60	41	55	
4		화력	484	573	696	803	1,022	1,122	
5		원자력	529	563	565	581	587	670	
6		신재생	-	-	-	-	-	-	
7	북한	합계	277	263	247	221	231	230	
8		수력	156	150	142	133	138	142	
9		화력	121	113	105	88	93	88	
10		원자력	-	-	-	-	-	-	
11									
12									
13									
14									
15									
16									

데이터

115%

## Part 2. 데이터 입출력 [화면 녹화]

### 1-2. Excel 파일



남북한발전전력량.xlsx - Excel

kim hs

파일 홈 삽입 페이지 레이아웃 수식 데이터 검토 보기 개발 도구 도움말 어떤 작업을 원하시나요? 공유

붙여넣기 글꼴 맞춤 표시 형식 스타일 셀 정렬 및 필터 찾기 및 선택

C15

	A	B	C	D	E	F	G	H	
1	전력량 (억kWh)	발전 전력별	1990	1991	1992	1993	1994	1995	1996
2	남한	합계	1,077	1,186	1,310	1,444	1,650	1,847	
3		수력	64	51	49	60	41	55	
4		화력	484	573	696	803	1,022	1,122	
5		원자력	529	563	565	581	587	670	
6		신재생	-	-	-	-	-	-	
7	북한	합계	277	263	247	221	231	230	
8		수력	156	150	142	133	138	142	
9		화력	121	113	105	88	93	88	
10		원자력	-	-	-	-	-	-	
11									
12									
13									
14									
15									
16									

데이터

115%

# Part 2. 데이터 입출력

## 1-2. Excel 파일

Excel 파일(확장자: .xlsx)의 행과 열은 데이터프레임의 행, 열로 일대일 대응된다. read\_excel() 함수의 사용법은 앞에서 살펴본 read\_csv() 함수와 거의 비슷하다.

header, index\_col 등 대부분의 옵션을 그대로 사용할 수 있다.

Excel 파일 → 데이터프레임: `pandas.read_excel("파일 경로(이름) ")`

### ① Excel 파일 미리보기

〈Excel 파일〉 미리보기 (File: example/part2/남북한발전전력량.xlsx)

전력량 (kWh)	발전 전력별	1990	1991	1992	...	2012	2013	2014	2015	2016
남한	합계	1077	1186	1310	...	5096	5171	5220	5281	5404
NaN	수력	64	51	49	...	77	84	78	58	66
NaN	화력	484	573	696	...	3430	3581	3427	3402	3523
NaN	원자력	529	563	565	...	1503	1388	1564	1648	1620
NaN	신재생	-	-	-	...	86	118	151	173	195
북한	합계	277	263	247	...	215	221	216	190	239
NaN	수력	156	150	142	...	135	139	130	100	128
NaN	화력	121	113	105	...	80	82	86	90	111
NaN	원자력	-	-	-	...	-	-	-	-	-

### ② Excel 파일 읽어오기

〈예제 2-2〉 Excel 파일 읽기 (File: example/part2/2.2\_read\_excel.py)

```
1 # -*- coding: utf-8 -*-
2
3 import pandas as pd
4
5 # read_excel() 함수로 데이터프레임 변환
6 df1 = pd.read_excel('./남북한발전전력량.xlsx') # header=0 (default 옵션)
7 df2 = pd.read_excel('./남북한발전전력량.xlsx', header=None) # header = None 옵션
```

```
9 # 데이터프레임 출력
10 print(df1)
11 print('\n')
12 print(df2)
```

〈실행 결과〉 코드 전부 실행

전력량 (kWh)	발전 전력별	1990	1991	1992	...	2012	2013	2014	2015	2016	
0	남한	합계	1077	1186	1310	...	5096	5171	5220	5281	5404
1	NaN	수력	64	51	49	...	77	84	78	58	66
2	NaN	화력	484	573	696	...	3430	3581	3427	3402	3523
3	NaN	원자력	529	563	565	...	1503	1388	1564	1648	1620
4	NaN	신재생	-	-	-	...	86	118	151	173	195
5	북한	합계	277	263	247	...	215	221	216	190	239
6	NaN	수력	156	150	142	...	135	139	130	100	128
7	NaN	화력	121	113	105	...	80	82	86	90	111
8	NaN	원자력	-	-	-	...	-	-	-	-	-

[9 rows x 29 columns]

0	1	2	3	4	...	24	25	26	27	28	
0	전력량 (kWh)	발전 전력별	1990	1991	1992	...	2012	2013	2014	2015	2016
1	남한	합계	1077	1186	1310	...	5096	5171	5220	5281	5404
2	NaN	수력	64	51	49	...	77	84	78	58	66
3	NaN	화력	484	573	696	...	3430	3581	3427	3402	3523
4	NaN	원자력	529	563	565	...	1503	1388	1564	1648	1620
5	NaN	신재생	-	-	-	...	86	118	151	173	195
6	북한	합계	277	263	247	...	215	221	216	190	239
7	NaN	수력	156	150	142	...	135	139	130	100	128
8	NaN	화력	121	113	105	...	80	82	86	90	111
9	NaN	원자력	-	-	-	...	-	-	-	-	-

[10 rows x 29 columns]

header 옵션을 추가하지 않은 경우에는 Excel 파일의 첫 행이 열 이름을 구성한다. 한편, header=None 옵션을 사용하면, 정수형 인덱스(0, 1, 2, ...)를 열 이름으로 자동 할당한다.

## Part 2. 데이터 입출력 [화면 녹화]

The screenshot displays the Spyder Python IDE interface. The main editor window shows a Python script with the following code:

```
1 # -*- coding: utf-8 -*-
2
3 import pandas as pd
4
5 # read_excel() 함수로 데이터프레임 변환
6 df1 = pd.read_excel('./남북한발전전력량.xlsx') # header=0 (default 옵션)
7 df2 = pd.read_excel('./남북한발전전력량.xlsx', header=None) # header=None 옵션
8
9 # 데이터프레임 출력
10 print(df1)
11 print('\n')
12 print(df2)
```

The Variable explorer on the right is empty. The IPython console at the bottom shows the following output:

```
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]:
In [1]:
```

The status bar at the bottom indicates: Permissions: RW, End-of-lines: CRLF, Encoding: UTF-8, Line: 1, Column: 1, Memory: 72.

# Part 2. 데이터 입출력

## 1-3. JSON 파일

JSON 파일(확장자: .json)은 JavaScript에서 유래한 데이터 공유를 목적으로 개발된 특수한 파일형식이다.

파이썬 딕셔너리와 비슷하게 'key : value' 구조를 갖는다.

read\_json() 함수를 사용하여, JSON 파일을 데이터프레임으로 변환한다.

JSON 파일 → 데이터프레임: `pandas.read_json("파일 경로(이름) ")`

### ① JSON 파일 미리보기

〈JSON 파일〉 미리보기 (File: example/part2/read\_json\_sample.json)

```
1 {
2   "name": {"pandas": "",
3            "NumPy": "",
4            "matplotlib": ""},
5
6   "year": {"pandas": 2008,
7            "NumPy": 2006,
8            "matplotlib": 2003},
9
10  "developer": {"pandas": "Wes McKinney",
11               "NumPy": "Travis Oliphant",
12               "matplotlib": "John D. Hunter"},
13
14  "opensource": {"pandas": "True",
15                "NumPy": "True",
16                "matplotlib": "True"}
17 }
```

JSON 파일에는 주요 파이썬 패키지의 출시년도, 개발자, 오픈소스 정보가 들어있다.

### ② JSON 파일 읽어오기

〈예제 2-3〉 JSON 파일 읽기

(File: example/part2/2.3\_read\_json.py)

```
1  # -*- coding: utf-8 -*-
2
3  import pandas as pd
4
5  # read_json() 함수로 데이터프레임 변환
6  df = pd.read_json('./read_json_sample.json')
7  print(df)
8  print('\n')
9  print(df.index)
```

〈실행 결과〉 코드 전부 실행

name	year	developer	opensource
NumPy	2006	Travis Oliphant	True
matplotlib	2003	John D. Hunter	True
pandas	2008	Wes McKinney	True

Index(['NumPy', 'matplotlib', 'pandas'], dtype='object')

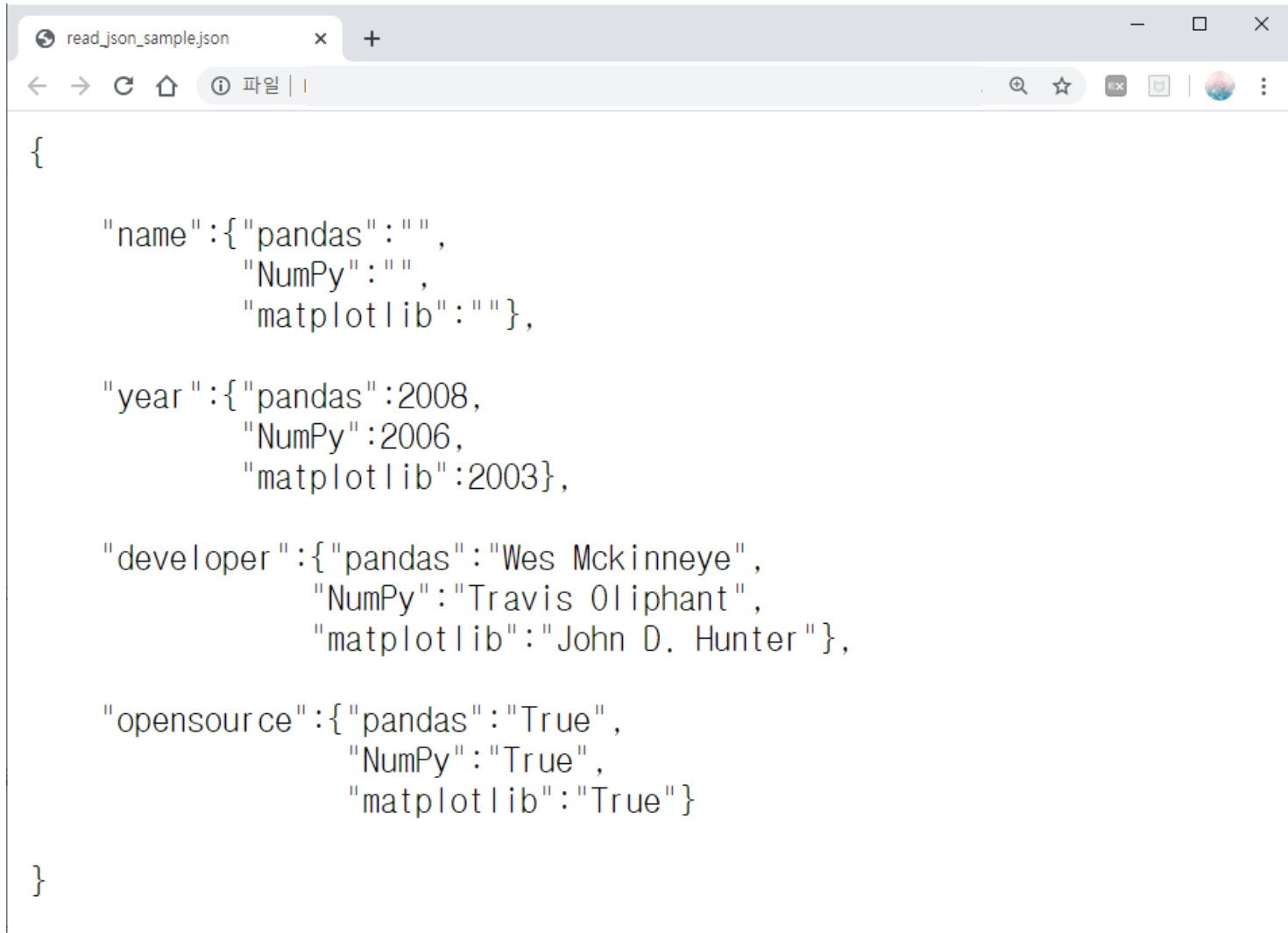
JSON 파일의 "name" 데이터("pandas", "NumPy", "matplotlib")가 인덱스로 지정된다.





## Part 2. 데이터 입출력

### 1-3. JSON 파일



```
{
  "name": {
    "pandas": "",
    "NumPy": "",
    "matplotlib": ""
  },
  "year": {
    "pandas": 2008,
    "NumPy": 2006,
    "matplotlib": 2003
  },
  "developer": {
    "pandas": "Wes Mckinney",
    "NumPy": "Travis Oliphant",
    "matplotlib": "John D. Hunter"
  },
  "opensource": {
    "pandas": "True",
    "NumPy": "True",
    "matplotlib": "True"
  }
}
```



## Part 2. 데이터 입출력 [화면 녹화]

The screenshot displays the Spyder Python IDE interface. The main editor window shows a Python script with the following code:

```
1 # -*- coding: utf-8 -*-
2
3 import pandas as pd
4
5 # read_json() 함수로 데이터프레임 변환
6 df = pd.read_json('./read_json_sample.json')
7 print(df)
8 print('\n')
9 print(df.index)
```

The script is saved in a file named `2,3_read_json.py`. The Variable explorer on the right shows no variables. The IPython console at the bottom displays the following output:

```
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 indicates the following settings: Permissions: RW, End-of-lines: CRLF, Encoding: UTF-8, Line: 9, Column: 16, Memory: 77 %.

## Part 2. 데이터 입출력 [화면 녹화]

### 2-1. HTML 웹 페이지에서 표 속성 가져오기

The screenshot shows the Spyder Python IDE interface. The top pane displays a web browser window with a file named 'sample.html'. The browser shows a table with the following content:

c0	c1	c2	c3
0	0	1	4
1	1	2	5
2	2	3	6

Below the table, there is a table with the following content:

	name	year	developer	opensource
0	NumPy	2006	Travis Oliphant	True
1	matplotlib	2003	John D. Hunter	True
2	pandas	2008	Wes Mckinney	True

The bottom pane shows a Python script in the editor. The script is as follows:

```
1 # -*- coding: utf-8 -*-
2
3 import pandas as pd
4
5 # Read JSON data
6 df = pd.read_json('data.json')
7 print(df)
8 print('\n')
9 print(df.to_html())
```

The bottom status bar shows the following information: Permissions: RW, End-of-lines: CRLF, Encoding: UTF-8, Line: 9, Column: 16, Memory: 78%.

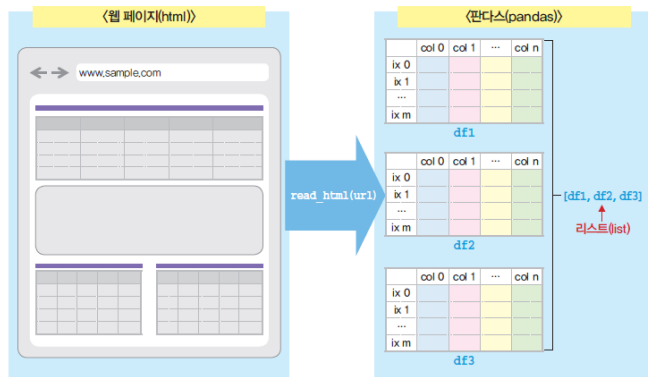
## Part 2. 데이터 입출력

### 2. 웹(web)에서 가져오기

#### 2-1. HTML 웹 페이지에서 표 속성 가져오기

read\_html() 함수는 HTML 웹 페이지에 있는 <table> 태그에서 표 형식의 데이터를 모두 찾아서 데이터프레임으로 변환한다.

HTML 표 속성 읽기 : `pandas.read_html( "웹 주소(URL)" 또는 "HTML 파일 경로(이름)" )`



[그림 2-3] HTML 페이지의 표 가져오기

<예제 2-4> 웹에서 표 정보 읽기

(File: example/part2/2.4\_read\_html.py)

```
1 # -*- coding: utf-8 -*-
2
3 import pandas as pd
4
5 # HTML 파일 경로 or 웹 페이지 주소를 url 변수에 저장
6 url = './sample.html'
7
8 # HTML 웹페이지의 표(table)를 가져와서 데이터프레임으로 변환
9 tables = pd.read_html(url)
10
11 # 표(table)의 개수 확인
12 print(len(tables))
```

```
15 # tables 리스트의 원소를 iteration하면서 각각 화면 출력
16 for i in range(len(tables)):
17     print("tables[%s]" % i)
18     print(tables[i])
19     print('\n')
20
21 # 파이썬 패키지 정보가 들어 있는 두 번째 데이터프레임을 선택하여 df 변수에 저장
22 df = tables[1]
23
24 # 'name' 열을 인덱스로 지정
25 df.set_index(['name'], inplace=True)
26 print(df)
```

<실행 결과> 코드 전부 실행

2

tables[0]

Unnamed: 0	c0	c1	c2	c3	
0	0	0	1	4	7
1	1	1	2	5	8
2	2	2	3	6	9

tables[1]

	name	year	developer	opensource
0	NumPy	2006	Travis Oliphant	True
1	matplotlib	2003	John D. Hunter	True
2	pandas	2008	Wes McKinney	True

	year	developer	opensource
name			
NumPy	2006	Travis Oliphant	True
matplotlib	2003	John D. Hunter	True
pandas	2008	Wes McKinney	True

표 데이터들은 각각 별도의 데이터프레임으로 변환되기 때문에, 여러 개의 데이터프레임(표)을 원소로 갖는 리스트가 반환된다.

## Part 2. 데이터 입출력 [화면 녹화]

Spyder (Python 3.7)

File Edit Search Source Run Debug Consoles Projects Tools View Help

2,1\_read\_csv.py 2,2\_read\_excel.py 2,3\_read\_json.py 2,4\_read\_html.py

```
1 #-*- coding: utf-8 -*-
2
3 import pandas as pd
4
5 # HTML 파일 경로 or 웹 페이지 주소를 url 변수에 저장
6 url = './sample.html'
7
8 # HTML 웹페이지의 표(table)를 가져와서 데이터프레임으로 변환
9 tables = pd.read_html(url)
10
11 # 표(table)의 개수 확인
12 print(len(tables))
13 print('\n')
14
15 # tables 리스트의 원소를 iteration하면서 각각 화면 출력
16 for i in range(len(tables)):
17     print("tables[%s]" % i)
18     print(tables[i])
19     print('\n')
20
21 # 파이썬 패키지 정보가 들어 있는 두 번째 데이터프레임을 선택하여 df 변수에 저장
22 df = tables[1]
23
24 # 'name' 열을 인덱스로 지정
25 df.set_index(['name'], inplace=True)
26 print(df)
```

Variable explorer

Name	Type	Size	Value
------	------	------	-------

IPython console

Console 13/A

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: 1 Column: 1 Memory: 81%

# Any Question?

Thank you.

