PANDAS 데이터 파일 처리

데이터 파일 처리

◆ 파일 입력 & 출력 함수

- 다양한 형태의 외부 파일 읽어 **DataFrame** 변환
- DataFrame함수로 다양하게 데이터 분석 및 가공
- 파일 형식 : csv, json, xlsx, xml, html, yml 등등

데이터 파일 처리

◆ 파일 입력 & 출력 함수

파일 형식	READER	WRITER
CSV	read_csv( )	to_csv( )
JSON	read_json( )	to_json( )
HTML	read_html( )	to_html( )
Local clipboard	read_clipboard( )	to_clipboard( )
Ms Excel	read_excel( )	to_excel( )
HDF5 Format	read_hdf( )	to_hdf( )
SQL	read_sql( )	to_sql( )

데이터 파일 처리

◆ CSV 파일

- 데이터 값을 **쉼표(,)로 구분**하는 파일
- Comma Separated Values 약자
- **쉽표(,)로 열 구분, 줄바꿈으로 행(row) 구분**
- TSV(Tab), SSV(Space) 데이터 파일도 존재

데이터 파일 처리

◆ CSV 파일

형태 : 데이터, 데이터, 데이터

```
1,13.2,9.1,blue
1,98,2.8,gray
2,10.5,81.3,red
1,8.8,5.21,yellow
```

```
1;13.2;9.1;blue
1;98;2.8;gray
2;10.5;81.3;red
1;8.8;5.21;yellow
```

데이터 파일 처리

◆ CSV 파일

dataframe = Pandas.read_csv(“파일 경로(이름)”, 옵션 …)

옵션	기본값	기능
header	0	0번행을 열 이름 지정, None은 0번행 열 이름 지정 안함
Index_col	None	행 인덱스가 되는 열 번호 또는 열 이름 False : 인덱스 지정하지 않음 None : 0, 1, 2, …. 정수 인덱스
sep	,	텍스트 데이터 구분자 문자 지정
delimiterstr	None	텍스트 데이터 구분자 문자 지정
names		열 이름을 사용할 문자열 리스트
skiprows		처음 몇 줄을 skip할 것인지 설정
parse_dates	False	날짜/시간 데이터를 datetime64로 변환 여부 설정
Encoding		텍스트 인코딩 종류(UTF-X) 지정

데이터 파일 처리

◆ CSV 파일

```
import pandas as pd

# 2. CSV로 저장
csv_path = 'sample_data.csv'

# 3. read_csv() 시 parse_dates 적용
df1 = pd.read_csv(csv_path)
df2 = pd.read_csv(csv_path,
                  parse_dates=['date'])

# 4. 출력 확인
print("[df1]₩n", df1.dtypes)
print('------------')
print("[df2]₩n", df2.dtypes)
```

```
[df1]
date        object
product     object
price        int64
dtype: object

------------

[df2]
date        datetime64[ns]
product     object
price        int64
dtype: object
```

데이터 파일 처리

◆ EXCEL 파일 처리

```
dataframe = Pandas.read_excel( "파일 경로(이름)", 옵션 ... )
```

- read_csv함수와 옵션 일치

```
## "ImportError: Install xlrd "
      오류 발생시에는  xlrd 라이브러리 설치 필요
```

데이터 파일 처리

◆ JSON 파일

- JavaScript Object Notation 약자
- 자바스크립트에서 사용하는 객체 표기 방법
- **다양한 프로그래밍 언어에서 데이터 교환에 사용**
- **인코딩/디코딩 표준으로도 사용**

데이터 파일 처리

◆ JSON 파일

형태 : [{ 키:값, 키:값, 키:{ 키:값, 키:값 } }]

```
[ {  'id':1,
     'name':'jane',
     'data' : { 'major':'science', 'grade': 1 }
  }, { 'id':2,
      'name':'Tom',
      'data' : { 'major':'math', 'grade': 2 }
    } ]
```

데이터 파일 처리

◆ JSON 파일

▪ orient

- **JSON string의 foramt을 결정하는 방향 의미**

orient	전체 형태	요소 형태	구성	
columns	Dict	Dict	키 : 컬럼명	값 : 행 : 값
records	List	행단위 Dict	키 : 컬럼명	값 : 값
split	Dict	Dict	키 : columns index data	값 : [각 항목들]
values	List	List	값만 담기	
index	Dict	Dict	키 : 인덱스	값 : { 컬럼 : 값}

데이터 파일 처리

◆ JSON 파일

▪ orient

- JSON string의 foramt을 결정하는 방향 의미

```
[{"name":"Jack","age":26},{"name":"Ace","age":87}]
```

키 – 컬럼명
값 – 행 : 값

columns

```
{ "name" : {"0" : "Jack" ,"1" : "Ace" } , "age" : {"0" : 26, "1" : 87 } }
```

데이터 파일 처리

◆ JSON 파일

- orient

```
[{"name":"Jack","age":26},{"name":"Ace","age":87}]
```

행 단위
{ 컬럼명 : 값 }

records

```
[ {"name":"Jack","age":26} , {"name":"Ace","age":87} ]
```

데이터 파일 처리

◆ JSON 파일

- orient

```
[{"name":"Jack","age":26},{"name":"Ace","age":87}]
```

키 : columns, index, data
값 : 각 항목 리스트로

split

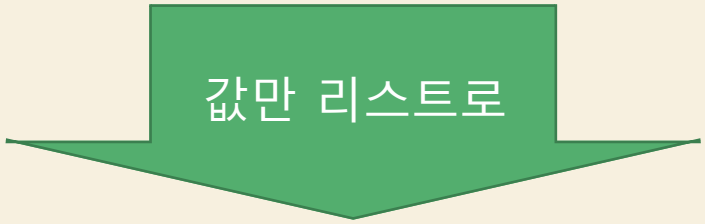
```
{ "columns":["name","age"],
  "index“  :[0,1],
  "data“   :[["Jack",26],["Ace",87]]}
```

데이터 파일 처리

◆ JSON 파일

- orient

```
[{"name":"Jack","age":26},{"name":"Ace","age":87}]
```



values

```
[["Jack",26],["Ace",87]]
```

데이터 파일 처리

◆ JSON 파일

- orient

```
[{"name":"Jack","age":26},{"name":"Ace","age":87}]
```

키 : index
값 : { 컬럼 : 값 }

index

```
{ "0“ : {"name":"Jack","age":26},
  "1“ : {"name":"Ace","age":87}}
```

데이터 파일 처리

◆ JSON 파일 처리

dataframe = Pandas.read_json(“파일 경로(이름)”, 옵션 …)

- 매개변수들
  - → path_or_buf : 읽을 JSON 파일의 경로 또는 JSON 문자열
  - → orient : JSON의 구조 지정. 기본값 None
  - → typ : 반환할 객체 유형, 기본값 데이터프레임
  - → lines : JSON 데이터 줄바꿈으로 나누어져 있을 경우 True 설정

데이터 파일 처리

◆ JSON 파일 처리

```
import pandas as pd

# JSON 파일 경로
json_file = 'data_lines.json'

# JSON 파일 읽기
data = pd.read_json(json_file, , lines=True)

# 데이터프레임 출력
print(data)
```

데이터 파일 처리

◆ HTML 파일

- HyperText Markup Language 약자
- 웹 페이지의 구조를 정의하는 마크업 언어
- **프로그래밍 언어는 아니며, 웹 페이지의 뼈대 구성**
- **인코딩/디코딩 표준으로도 사용**

데이터 파일 처리

◆ HTML 파일 구조

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
 <HEAD>
  <TITLE> New Document </TITLE>
  <META NAME="Generator" CONTENT="EditPlus">
  <META NAME="Author" CONTENT="">
  <META NAME="Keywords" CONTENT="">
  <META NAME="Description" CONTENT="">
 </HEAD>

 <BODY bgcolor='yellow'>
  <h1> HEADER </h1>
  <a href="http://www.naver.com">GO~! NAVER</a>

  <div>
   SPACE 1
  </div>

  <div>
   SPACE 2
  </div>

 </BODY>
</HTML>
```

태그(Tag)

-HTML 구성 요소
-사용자에게 보여줄 여러가지 요소
-형식 : < 태그명 > </태그명>
-역할에 따라 다양한 속성 존재

데이터 파일 처리

◆ HTML 파일

dataframe = Pandas.read_html("파일 경로(이름)", 옵션 ...)

- Web Page에 **<table> 태그의 표 형식 데이터**를 읽어옴
- 테이블 데이터는 각각 별도의 DataFrame으로 변환
- DataFrame 리스트 반환

"ImportError: Install lxml "

오류 발생시에는 lxml 라이브러리 설치 필요

➔ **pip install lxml**