

데이터 분석 with 파이썬

파일 입출력

목차

1. CSV 파일
2. 엑셀 파일

01

CSV 파일

01. CSV 파일

I. CSV 파일 이해하기

- 공공데이터포털(www.data.go.kr) 대부분의 공공데이터 파일은 CSV 형식

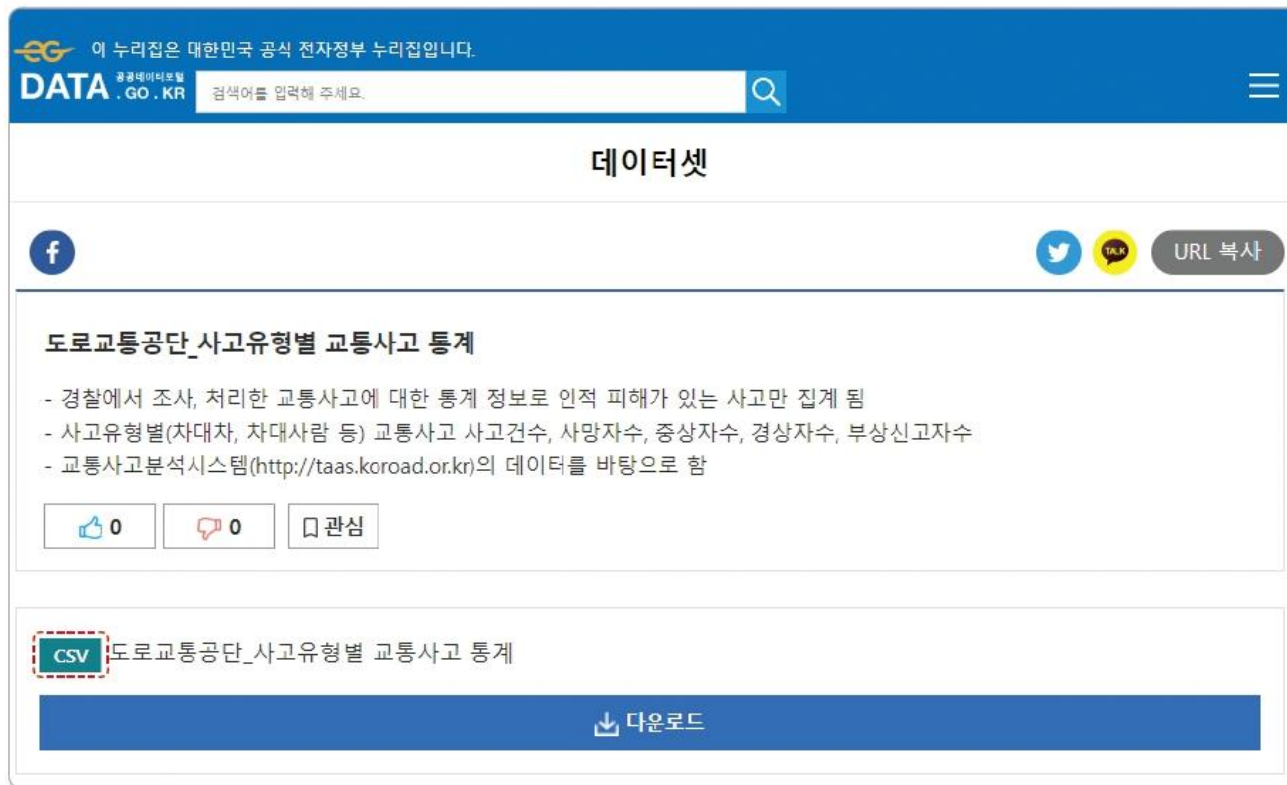


그림 3-1 공공데이터포털에서 제공하는 CSV 데이터셋

01. CSV 파일

I. CSV 파일 이해하기

- CSV(Comma-separated Value) 파일은 데이터를 쉼표(,)로 구분하는 텍스트 데이터 형식.
 - 메모장을 실행하고 첫 번째 행에는 **이름,국어,영어,수학**,
두 번째 행에는 **철수,100,90,80**, 세 번째 행에는 **영수,90,95,75**를 입력.
 - 파일 이름을 'score.csv'로 입력하고 파일 형식은 [모든 파일]을 선택.

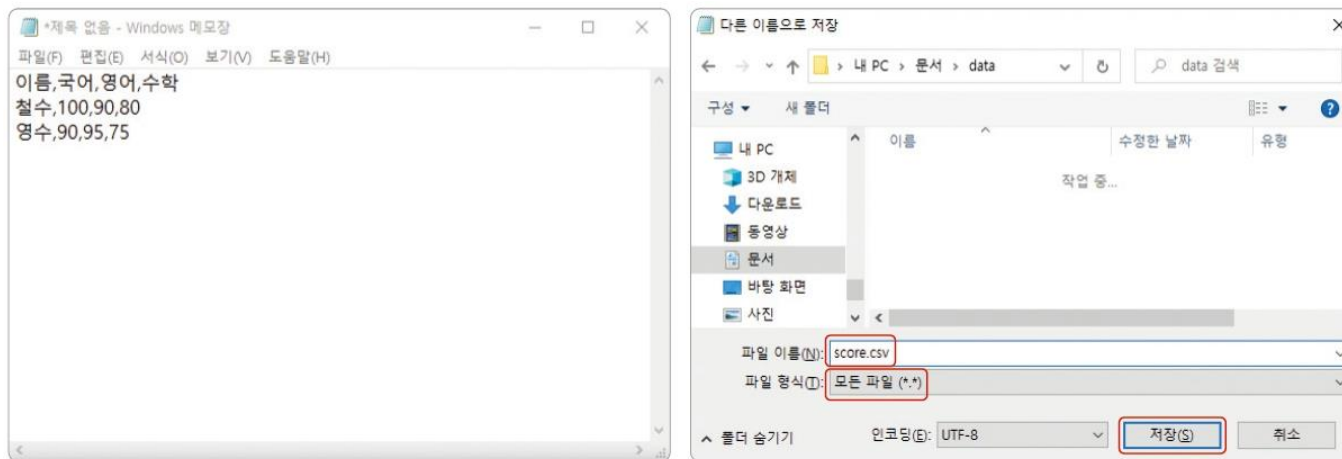


그림 3-2 메모장으로 CSV 파일 생성

01. CSV 파일

I. CSV 파일 이해하기

- CSV(Comma-separated Value) 파일은 데이터를 쉼표(,)로 구분하는 텍스트 데이터 형식.
 - PC에 엑셀 프로그램이 설치되어 있다면 'score.csv' 파일이 탐색기에서 엑셀 파일(xlsx) 아이콘과 비슷한 모양으로 나타남.

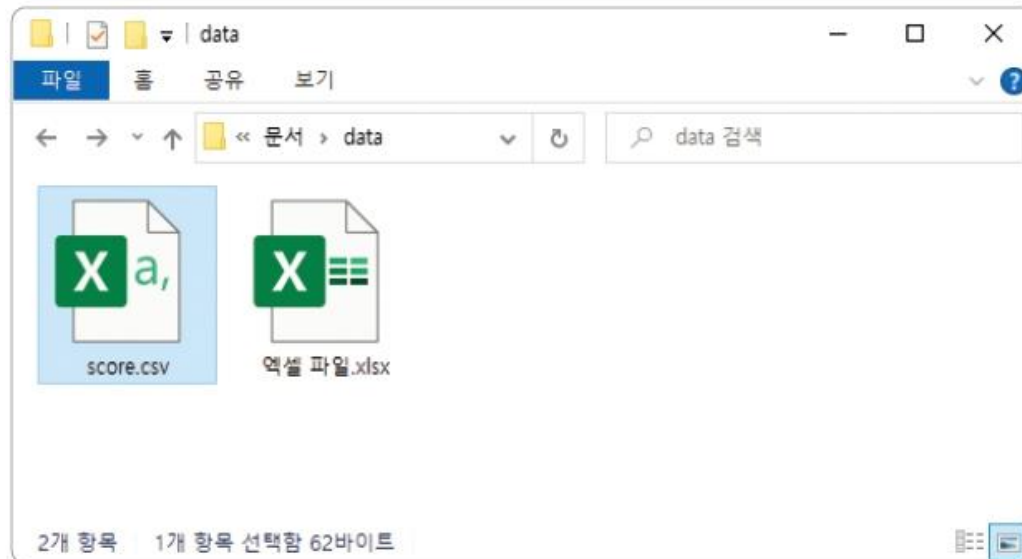


그림 3-3 CSV 파일과 엑셀 파일

01. CSV 파일

I. CSV 파일 이해하기

- 데이터에 한글이 포함된 CSV 파일
 - 엑셀 프로그램으로 'score.csv' 파일을 열고 내용을 확인해 보면 [그림 3-4]와 같이 한글이 깨져 보일 것 (인코딩 문제).
 - 인코딩(Encoding): 문자 데이터를 컴퓨터에게 전달하는 방식.

	A	B	C	D
1	?대쫘	援?뵤	?곡뵤	?샐뵤
2	泥쫘뵤	100	90	80
3	?곡뵤	90	95	75

그림 3-4 엑셀로 CSV 파일 열기(1)

01. CSV 파일

I. CSV 파일 이해하기

- 데이터에 한글이 포함된 CSV 파일
 - 메모장으로 'score.csv' 파일을 열고 단축키 [Ctrl] + [Shift] + [S] 를 눌러 [다른 이름으로 저장].
 - 인코딩을 [ANSI]로 변경하고 'score_a.csv' 파일로 저장.

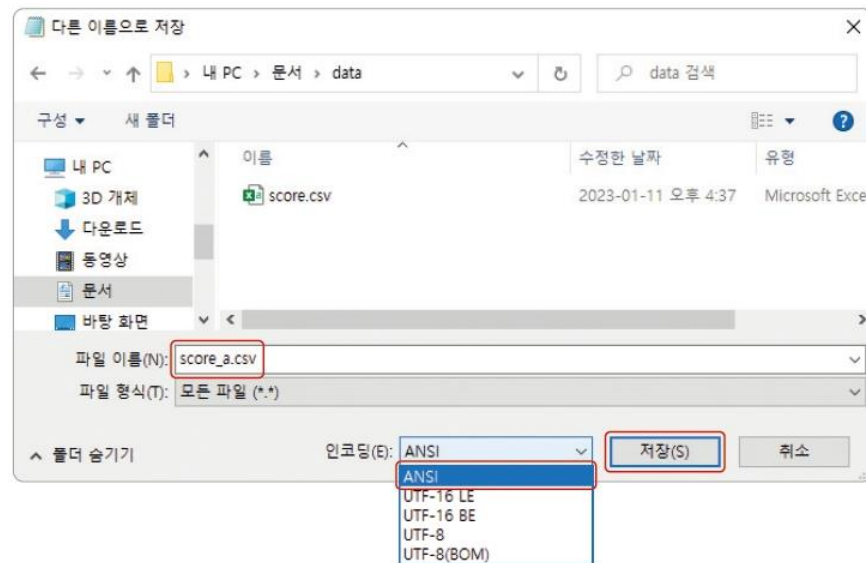


그림 3-5 메모장에서 인코딩 변경하기

01. CSV 파일

I. CSV 파일 이해하기

- 데이터에 한글이 포함된 CSV 파일
 - 새로 저장한 'score_a.csv' 파일을 엑셀 프로그램으로 다시 열어보면 [그림 3-6]과 같이 한글이 깨지지 않고 제대로 보임.

	A	B	C	D
1	이름	국어	영어	수학
2	철수	100	90	80
3	영수	90	95	75

그림 3-6 엑셀로 CSV 파일 열기(2)

01. CSV 파일

I. CSV 파일 이해하기

- 데이터에 쉼표가 포함된 CSV 파일
 - 데이터에 쉼표(,)가 포함된 경우 유의해야 함.

표 3-1 쉼표가 포함된 데이터

품목	가격
딸기,사과	1,000
수박	5,000

- 메모장에서 다음과 같이 작성하여 'fruits.csv'로 저장.

```
품목,가격  
딸기,사과,1,000  
수박,5,000
```

01. CSV 파일

I. CSV 파일 이해하기

- 데이터에 쉼표가 포함된 CSV 파일
 - 같은 셀에 있어야 할 딸기와 사과가 쉼표 때문에 다른 셀에 나타나며 숫자들도 천 단위로 쉼표가 있어 분리됨.

	A	B	C	D
1	품목	가격		
2	딸기	사과	1	0
3	수박	5	0	

그림 3-7 쉼표가 포함된 데이터(1)

- 같은 셀에 넣을 데이터에 쉼표가 포함된 경우에는 따옴표로 묶어서 작성.

```
품목,가격  
"딸기,사과","1,000"  
수박,"5,000"
```

01. CSV 파일

I. CSV 파일 이해하기

- 데이터에 쉼표가 포함된 CSV 파일
 - 단축키 [Ctrl] + [Shift] + [S] 를 눌러 [다른 이름으로 저장] 대화상자를 열고 'fruits_q.csv' 파일로 저장.

	A	B
1	품목	가격
2	딸기,사과	1,000
3	수박	5,000

그림 3-8 쉼표가 포함된 데이터(2)

✓하나 더 알기: 데이터의 포맷

CSV만큼이나 자주 쓰이는 데이터 파일 형식이 XML과 JSON. XML과 JSON 파일은 웹상의 데이터를 나타내는데, 네트워크를 통해 데이터를 주고받을 때 데이터에 마크업(Mark-up). 즉, 문서나 데이터의 구조를 밝힐 수 있어서 데이터를 정제하거나 분석하는데 매우 효율적.

01. CSV 파일

II. 파이썬으로 CSV 파일 다루기

- 구글 Colab에서 파이썬으로 CSV 파일을 다루보기
 - 메모장을 열고 다음 내용을 CSV 파일에 작성하여 'characters.csv' 파일로 저장.

```
ID, 이름, 상징색, 취미, 특징
001, 뽀로로, 파랑색, 낚시, 펭귄
002, 에디, 주황색, 과학실험, 사막여우
003, 크롱, 초록색, 눈싸움, 공룡
004, 루피, 분홍색, 요리, 비버
005, 패티, 보라색, 운동, 펭귄
```

- 구글 Colab에 파일을 업로드하려면 먼저 import 명령어로 files 라이브러리를 로드해야 함.

[코드 3-1] files 라이브러리

```
from google.colab import files

#파일 업로드 함수 호출하기
f = files.upload( )
```

01. CSV 파일

II. 파이썬으로 CSV 파일 다루기

- 구글 Colab에서 파이썬으로 CSV 파일을 다루보기
 - [코드 3-1]을 실행하면 PC에 저장된 로컬 파일을 업로드하는 [파일 선택] 버튼이 나타남.
 - 버튼을 누르고 파일을 업로드하면 파일이 변수 f에 담김.

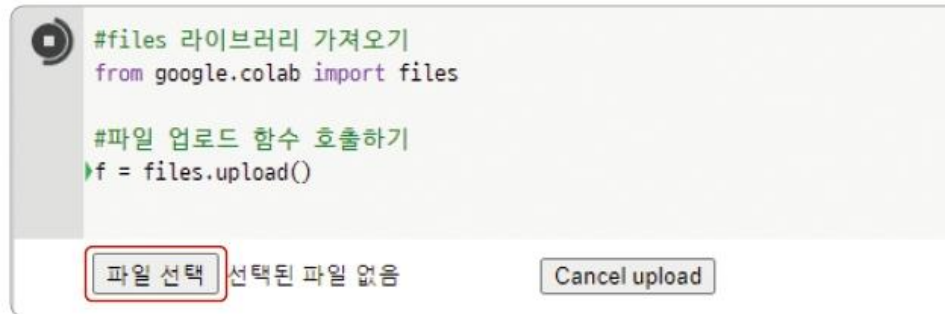


그림 3-9 파일 업로드

01. CSV 파일

II. 파이썬으로 CSV 파일 다루기

- csv 라이브러리
 - 업로드한 CSV 파일은 csv 라이브러리를 이용하여 다룰 수 있다.

[코드 3-2] CSV 파일 불러오기

```
f = open('characters.csv', 'r', encoding = 'cp949')
```

- open() 함수를 호출하여 로컬에서 업로드한 파일을 불러와 f 변수에 담기.
- CSV 파일을 윈도우 운영체제에서 만들었다면 cp949를 입력.

01. CSV 파일

II. 파이썬으로 CSV 파일 다루기

- csv 라이브러리

[코드 3-3] CSV 파일 읽기

```
import csv

rdr = csv.reader(f)

#반복문으로 한 행씩 출력하기
for line in rdr:
    print(line)
```

[코드 3-3] 실행결과

```
['ID', '이름', '상징색', '취미', '특징']
['001', '뽀로로', '파랑색', '냠시', '펭귄']
['002', '에디', '주황색', '과학실험', '사막여우']
['003', '크롱', '초록색', '눈싸움', '공룡']
['004', '루피', '분홍색', '요리', '비버']
['005', '패티', '보라색', '운동', '펭귄']
```


01. CSV 파일

II. 파이썬으로 CSV 파일 다루기

- CSV와 리스트

– 인덱스를 이용하여 'characters.csv' 파일의 데이터 중 두 번째 열의 이름만 출력.

표 3-2 'characters.csv' 파일의 데이터

ID	이름	상징색	취미	특징
001	뽀로로	파랑색	낚시	펭귄
002	에디	주황색	과학실험	사막여우
003	크롱	초록색	눈싸움	공룡
004	루피	분홍색	요리	비버
005	패티	보라색	운동	펭귄

01. CSV 파일

II. 파이썬으로 CSV 파일 다루기

- CSV와 리스트
 - 인덱스를 이용하여 'characters.csv' 파일의 데이터 중 두 번째 열의 이름만 출력.

[코드 3-4] 특정 열 출력

```
f = open('characters.csv', 'r', encoding = 'cp949')
rdr = csv.reader(f)

#두 번째 열을 반복문으로 한 행씩 출력하기

for line in rdr:
    print(line[1])
```

[코드 3-4] 실행결과

```
이름
뽀로로
에디
크롱
루피
패티
```

01. CSV 파일

II. 파이썬으로 CSV 파일 다루기

- CSV와 리스트
 - 인덱스를 이용하여 'characters.csv' 파일의 데이터 중 두 번째 열의 이름만 출력.

[코드 3-5] 특정 열 출력

```
f.close( )
```

- 읽기가 끝났어도 변수 f에는 파일을 읽어온 내용이 남아있음.
- 다 사용하고 난 후에는 close() 함수를 호출하여 자원을 반환해야 함.

01. CSV 파일

II. 파이썬으로 CSV 파일 다루기

[코드 3-6] CSV 파일 작성

```
#csv 라이브러리 가져오기
import csv

#CSV 파일 불러오기
f = open('write.csv', 'w', encoding='cp949', newline='')

#CSV 파일 작성할 준비하기
wr = csv.writer(f)

#데이터를 한 행씩 작성하기
wr.writerow(['ID', '이름', '상징색', '취미', '특징'])
wr.writerow(['001', '뽀로로', '파랑색', '늑시', '펭귄'])
wr.writerow(['002', '에디', '주황색', '과학실험', '사막여우'])
wr.writerow(['003', '크롱', '초록색', '눈싸움', '공룡'])
wr.writerow(['004', '루피', '분홍색', '요리', '비버'])
wr.writerow(['005', '패티', '보라색', '운동', '펭귄'])

#f 변수의 자원 반환하기
f.close()
```

- open() 함수로 빈 파일을 불러오기.
- csv.writer() 함수를 호출하여 wr 변수에 CSV 파일을 작성할 준비를 한 다음 writerow() 함수를 호출하여 행별로 데이터를 작성.
- f.close()를 호출하여 자원을 반환

01. CSV 파일

II. 파이썬으로 CSV 파일 다루기

- 파이썬으로 CSV 파일 작성하기
 - [코드 3-6]을 실행하면 Colab 탐색 창의 디렉토리에 'write.csv' 파일 생성됨.

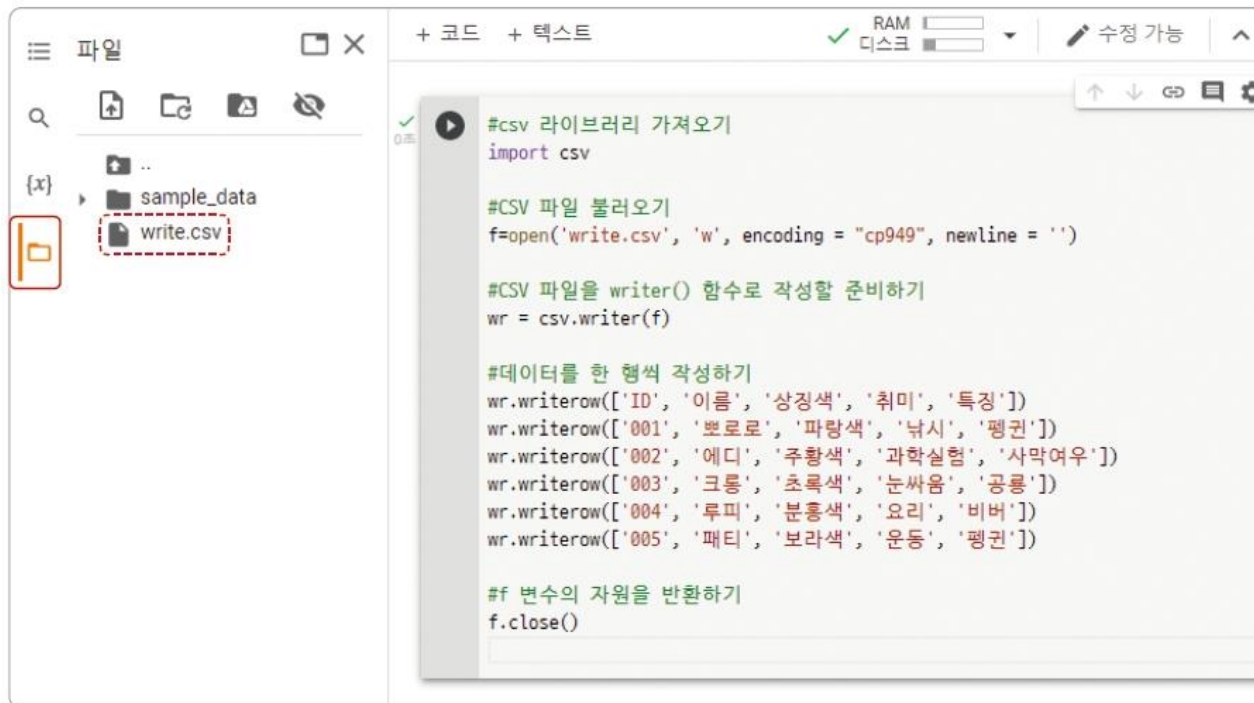


그림 3-10 디렉토리에 'write.csv' 파일 생성

01. CSV 파일

II. 파이썬으로 CSV 파일 다루기

- 데이터 추가하기
 - CSV 파일 마지막 행 아래에 한 행을 추가하여 캐릭터 데이터를 입력.

표 3-3 캐릭터 추가

ID	이름	상징색	취미	특징
001	뿌로로	파랑색	낚시	펭귄
002	에디	주황색	과학실험	사막여우
003	크롱	초록색	눈싸움	공룡
004	루피	분홍색	요리	비버
005	패티	보라색	운동	펭귄
006	포비	흰색	춤	북극곰

01. CSV 파일

II. 파이썬으로 CSV 파일 다루기

- 데이터 추가하기

[코드 3-7] CSV 파일에 데이터 추가

```
f = open('write.csv', 'a', encoding='cp949', newline='')
wr = csv.writer(f)

wr.writerow(['006', '포비', '흰색', '춤', '북극곰'])

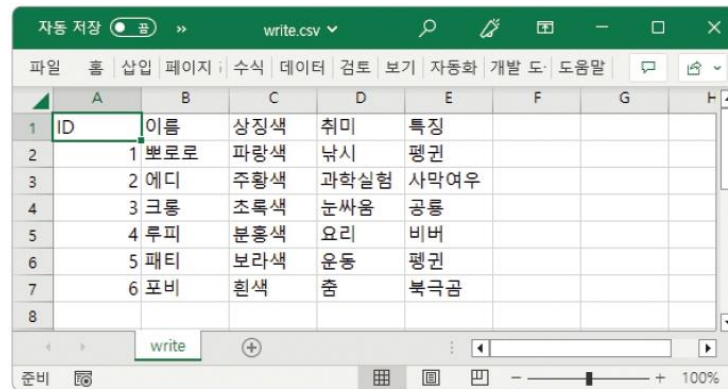
#f 변수의 자원 반환하기
f.close( )
```

- open() 함수의 두 번째 인자로 a(추가하기)를 사용.
- writerow() 함수로 내용을 작성하면 wr의 맨 끝에 추가됨.
- 데이터를 추가하는 동작이 끝나면 close() 함수를 호출하여 객체의 자원을 반환.

01. CSV 파일

II. 파이썬으로 CSV 파일 다루기

- 인덱스를 이용하여 'characters.csv' 파일의 데이터 중 두 번째 열의 이름만 출력.
 - 디렉토리에서 'write.csv' 파일을 다운로드하여 엑셀 프로그램으로 내용을 확인해 보면 데이터가 추가되어 있음



The screenshot shows an Excel window titled 'write.csv'. The spreadsheet contains the following data:

	A	B	C	D	E	F	G	H
1	ID	이름	상징색	취미	특징			
2		1 뽀로로	파랑색	낚시	펭귄			
3		2 에디	주황색	과학실험	사막여우			
4		3 크롱	초록색	눈싸움	공룡			
5		4 루피	분홍색	요리	비버			
6		5 패티	보라색	운동	펭귄			
7		6 포비	흰색	춤	북극곰			
8								

그림 3-11 추가된 데이터 확인

✓하나 더 알기: 판다스

판다스를 이용하면 데이터를 직관적으로 읽을 수 있고, 데이터를 정제하는 작업, 통계 및 데이터 분석 작업을 손쉽게 할 수 있다.

LAB. 식재료 관리 프로그램

데이터를 구글 Colab에서 CSV 파일로 작성하여 'weather.csv' 파일로 저장하기

날짜	요일	날씨	최저기온	최고기온	강수량
20230101	월	맑음	-4.3	3.8	0
20230102	화	맑음	-7.4	0.4	0
20230103	수	맑음	-9.0	0.6	0

1. import 키워드로 csv 라이브러리를 가져옴.

```
import csv
```

2. CSV 파일을 불러오기. 함수 writer()로 데이터를 작성할 준비.

```
f = open('weather.csv','w', newline='')  
wr = csv.writer(f)
```

LAB. 식재료 관리 프로그램

데이터를 구글 Colab에서 CSV 파일로 작성하여 'weather.csv' 파일로 저장하기

3. 함수 `writerow()`로 데이터를 한 행씩 추가.

```
wr.writerow(['날짜', '요일', '날씨', '최저기온', '최고기온', '강수량'])  
wr.writerow(['20230101', '월', '맑음', -4.3, 3.8, 0])  
wr.writerow(['20230102', '화', '맑음', -7.4, 0.4, 0])  
wr.writerow(['20230103', '수', '맑음', -9.0, 0.6, 0])
```

4. 객체의 자원을 반환.

```
f.close( )
```

02

엑셀 파일

02. 엑셀 파일

I. 워크시트와 워크북

- 엑셀은 CSV보다 많은 기능을 제공하는 데이터 형식.
 - 엑셀 파일에서 각 칸을 셀(Cell)이라고 부름.
 - 셀들이 모여 이루는 한 화면은 워크시트(Worksheet).
 - 워크시트가 모여 워크북(Workbook)이 되며, 워크북 단위로 엑셀 파일을 저장할 수 있음.

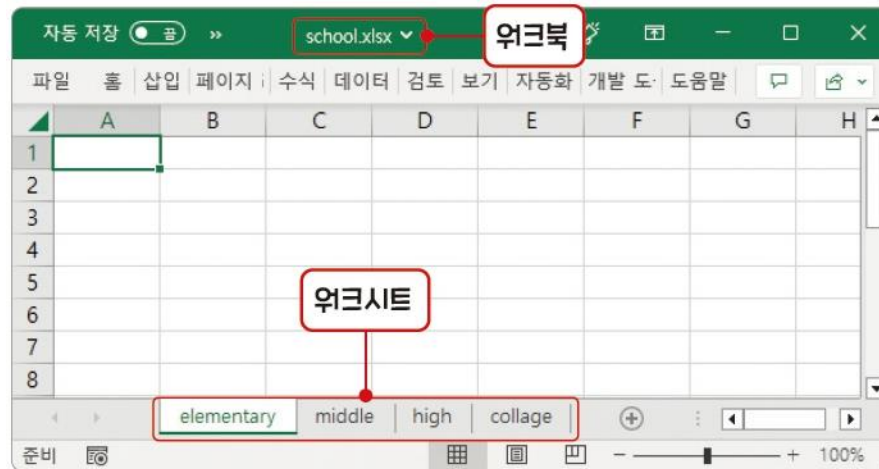
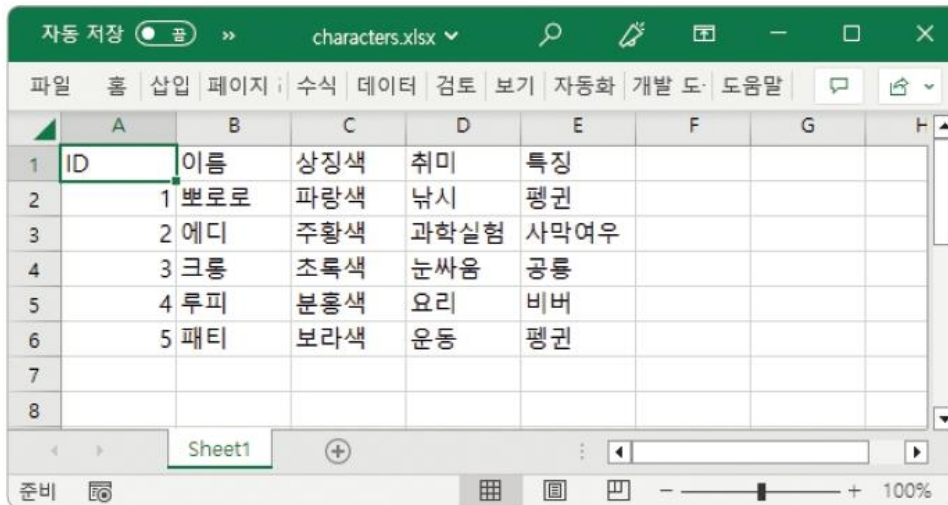


그림 3-12 워크시트와 워크북(1)

02. 엑셀 파일

I. 워크시트와 워크북

- 엑셀은 CSV보다 많은 기능을 제공하는 데이터 형식.
 - 워크시트와 워크북의 관계는 파일 박스와 문서의 관계에 비유할 수 있음.
 - 워크시트가 파일 박스 안에 담긴 내용물이라면 도서나 문서를 담은 파일 박스를 워크북이라고 할 수 있음.
 - 엑셀에서 [그림 3-14]의 데이터를 작성하고 'characters.xlsx'로 저장



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H
1	ID	이름	상징색	취미	특징			
2		1 뽀로로	파랑색	낙시	펭귄			
3		2 에디	주황색	과학실험	사막여우			
4		3 크롱	초록색	눈싸움	공룡			
5		4 루피	분홍색	요리	비버			
6		5 패티	보라색	운동	펭귄			
7								
8								

그림 3-14 데이터를 'characters.xlsx'로 저장

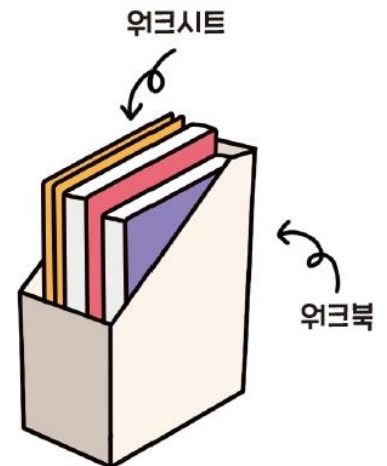


그림 3-13 워크시트와 워크북(2)

02. 엑셀 파일

II. 파이썬으로 엑셀 파일 읽기

- openpyxl 라이브러리를 활용하여 파이썬에서 엑셀 파일 다루기.

[코드 3-8] 엑셀 파일 업로드

```
from google.colab import files  
f = files.upload( )
```

- 실행 결과에서 [파일 선택] 버튼을 클릭하여 'characters.xlsx' 업로드.

[코드 3-8] 엑셀 파일 업로드

```
#openpyxl 라이브러리 설치하기  
!pip install openpyxl  
#openpyxl 라이브러리 가져오기  
import openpyxl  
  
wb = openpyxl.load_workbook('characters.xlsx')
```

- pip로 openpyxl 라이브러리를 설치, import 명령어로 가져오기.
- openpyxl 라이브러리의 load_workbook() 함수를 사용하여 파일을 불러오고 변수 wb에 이 엑셀 파일 전체를 담기.

02. 엑셀 파일

II. 파이썬으로 엑셀 파일 읽기

- openpyxl 라이브러리를 활용하여 파이썬에서 엑셀 파일 다루기.

[코드 3-10] 특정 열 출력

```
print(wb.sheetnames)
```

[코드 3-10] 실행결과

```
이름  
[ 'Sheet1' ]
```

- sheetnames 속성으로 wb 변수에 담긴 엑셀 파일의 모든 워크시트 이름을 출력.

02. 엑셀 파일

II. 파이썬으로 엑셀 파일 읽기

- openpyxl 라이브러리를 활용하여 파이썬에서 엑셀 파일 다루기.

[코드 3-11] 특정 위치의 데이터 읽기

```
sheet1 = wb['Sheet1']  
print(sheet1['A1'].value)
```

[코드 3-11] 실행결과

ID

- Sheet1 워크시트를 변수 sheet1에 따로 담기
- 변수 sheet1에 담긴 워크시트의 A1 셀에 저장된 값을 불러와 print() 함수로 출력.

02. 엑셀 파일

III. 파이썬으로 엑셀 파일 작성하기

- 워크북과 워크시트 생성하기

[코드 3-12] 새 워크북 생성

```
wb = openpyxl.Workbook( )
```

- 엑셀 파일에는 최소 1개의 워크시트가 존재해야 하므로 wb 안에 워크시트 Sheet가 자동 생성.



그림 3-15 워크북 생성

02. 엑셀 파일

III. 파이썬으로 엑셀 파일 작성하기

- 워크북과 워크시트 생성하기

[코드 3-13] 새 워크시트 생성

```
#새 워크시트 생성하기  
wb.create_sheet('Sheet2')  
#시트 이름 출력하기  
print(wb.sheetnames)
```

[코드 3-13] 실행결과

```
['Sheet', 'Sheet2']
```

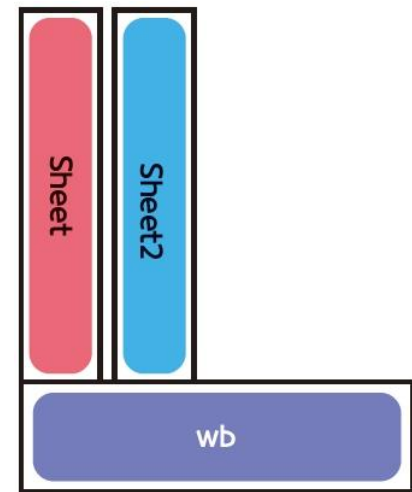


그림 3-16 워크시트 생성

- 함수 `create_sheet()`를 사용하여 워크북 `wb`에 새 워크시트 `Sheet2`를 생성.
- 객체 `wb`의 `sheetnames` 속성을 출력하여 현재 워크시트 목록을 확인.

02. 엑셀 파일

III. 파이썬으로 엑셀 파일 작성하기

- 워크시트 이름 변경하기

[코드 3-14] 워크시트 이름 변경

```
sheet1 = wb['Sheet']  
sheet2 = wb['Sheet2']  
  
sheet1.title = '캐릭터 명단'  
sheet2.title = '인기도 조사'  
  
print(wb.sheetnames)
```

[코드 3-14] 실행결과

```
['Sheet', 'Sheet2']
```



그림 3-17 워크시트 이름 변경

- 두 워크시트 Sheet와 Sheet2를 각각 변수 sheet1과 sheet2에 담기.
- title 속성을 변경 후, 현재 워크시트 목록을 출력하여 변경된 워크시트 이름을 확인.

02. 엑셀 파일

III. 파이썬으로 엑셀 파일 작성하기

- 워크시트에 데이터 입력하기

[코드 3-15] 워크시트 이름 변경

```
sheet2['B1'] = '인기도 조사결과'  
print(sheet2['B1'].value)
```

[코드 3-15] 실행결과

인기도 조사결과

- 변수 sheet2에 담긴 워크시트 Sheet2의 B1 셀에 값을 입력.

02. 엑셀 파일

III. 파이썬으로 엑셀 파일 작성하기

- 워크시트 복제하기

[코드 3-16] 워크시트 복제

```
copysheet = wb.copy_worksheet(sheet2)
print(wb.sheetnames)

#복제 시트의 이름 변경하기
copysheet.title = 'copy'

print(wb.sheetnames)
print(copysheet['B1'].value)
```

[코드 3-16] 실행결과

```
['캐릭터 명단', '인기도 조사', '인기도 조사 Copy']
['캐릭터 명단', '인기도 조사', 'copy']
인기도 조사결과
```

- 변수 sheet2를 copy_worksheet() 함수의 인자로 넣고 함수가 반환하는 워크시트는 변수 copysheet에 담기.
- 시트 이름을 출력하여 확인.
- 변수 copysheet에 담긴 워크시트의 title 속성을 copy로 변경.
- 다시 시트 이름을 출력해 보면 변수 copysheet에 담긴 워크시트 이름이 copy.
- 워크시트 내용까지 복제되었는지 확인하기 위하여 value 함수로 B1 셀의 값을 출력.

02. 엑셀 파일

III. 파이썬으로 엑셀 파일 작성하기

- 워크시트 복제하기



그림 3-18 워크시트 복사



그림 3-19 워크시트 복사

02. 엑셀 파일

III. 파이썬으로 엑셀 파일 작성하기

- 워크시트 삭제하기

[코드 3-17] 워크시트 삭제

```
del wb['인기도 조사']  
print(wb.sheetnames)
```

[코드 3-17] 실행결과

```
['캐릭터 명단', 'copy']
```

- 워크북 안에 워크시트가 두 개로 줄었고 변수 sheet2는 빈 그릇으로 남음.



그림 3-20 워크시트 삭제



그림 3-21 워크시트 삭제

02. 엑셀 파일

III. 파이썬으로 엑셀 파일 작성하기

- 엑셀 파일 저장하기

[코드 3-18] 엑셀 파일 저장

```
wb.save('complete.xlsx')
```

- 구글 Colab 탐색 창의 디렉토리에 'complete.xlsx' 파일이 생성됨.

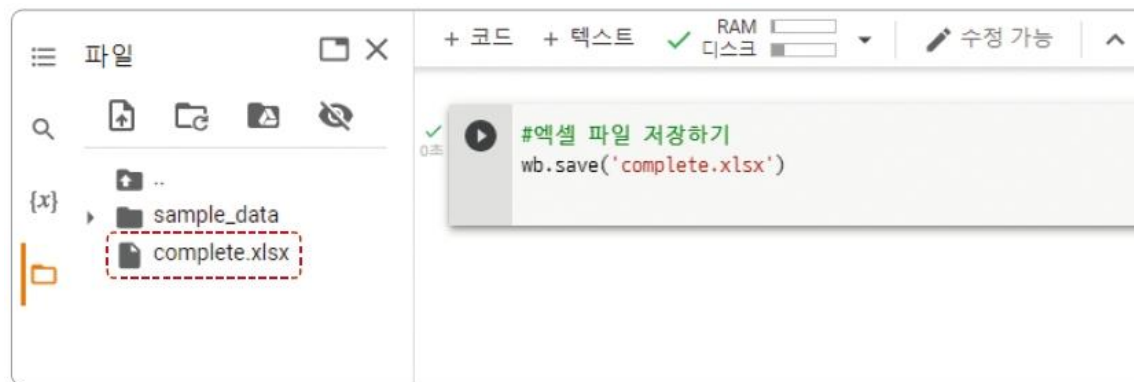


그림 3-22 디렉토리에 'complete.xlsx' 파일 생성

실전분석 교통 혼잡도 데이터 수집

[문제]

공공데이터포털(www.data.go.kr)에 접속하고
검색어로 **혼잡도**를 입력하여 데이터를 조회
합니다.

경기도 의정부시와 대전교통공사, 서울교통
공사에서 경전철 및 지하철 혼잡도 데이터를
CSV 파일로 제공합니다.

세 도시의 혼잡도 데이터를 파일 하나에 정
리하면 분석할 때 편리할 것입니다.

이 누리집은 대한민국 공식 전자정부 누리집입니다.

DATA 공공데이터포털 .GO . KR 검색어를 입력해 주세요.

전체 파일데이터 Open API

파일데이터	철도	자치행정기관
CSV	경기도 의정부시_의정부경전철 혼잡도	
수정일	2022-01-27	조회수 2143 다운로드 664

파일데이터	철도	공공기관
CSV	대전교통공사_열차 혼잡도 분석	
수정일	2022-12-09	조회수 3917 다운로드 1653

오픈 API	도로	자치행정기관
JSON	서울특별시_V2X 버스 승강장 혼잡 정보	
수정일	2022-07-25	조회수 771 활용신청 298

파일데이터	물류등기타	자치행정기관
HWP	서울특별시_지하철 혼잡도 통계	
수정일	2021-08-02	조회수 4243 다운로드 1081

파일데이터	철도	공공기관
CSV	서울교통공사_지하철혼잡도정보	
수정일	2022-04-11	조회수 24612 다운로드 4708

그림 3-23 공공데이터포털

실전분석 예금 이자 계산기

[해결]

1. 공공데이터포털에서 파일을 다운로드하고 차례로 'data1.csv', 'data2.csv', 'data3.csv'로 저장, 이를 다시 구글 Colab 디렉토리에 업로드.

```
from google.colab import files
```

```
f = files.upload( )
```

파일 선택 파일 3개

- **data3.csv**(text/csv) - 337792 bytes, last modified: 2022. 12. 17. - 100% done
- **data2.csv**(text/csv) - 1824 bytes, last modified: 2022. 12. 17. - 100% done
- **data1.csv**(text/csv) - 8196 bytes, last modified: 2022. 12. 17. - 100% done

Saving data3.csv to data3.csv

Saving data2.csv to data2.csv

Saving data1.csv to data1.csv

실전분석 예금 이자 계산기

[해결]

2. 먼저, 빈 워크북을 만들어 wb 변수에 담고 처음 워크북을 생성하면 기본 워크시트가 하나 만들어짐. 이 활성화된 워크시트를 ws라고 지정.

```
import openpyxl

wb = openpyxl.Workbook( )
ws = wb.active
```

3. csv 라이브러리의 reader() 함수로 'data1.csv' 파일의 내용을 읽습니다. 읽은 내용을 변수 reader에 담고, for 반복문으로 변수 reader의 행을 하나씩 워크시트 ws에 추가.

```
import csv

with open('data1.csv', encoding = 'cp949') as f:
    reader = csv.reader(f)
    for row in reader:
        ws.append(row)
```

실전분석 예금 이자 계산기

[해결]

4. openpyxl 라이브러리의 create_sheet() 함수로 워크북 wb에 두 번째 워크시트 ws2를 생성합니다. ws2에 'data2.csv'의 내용을 작성.

```
ws2 = wb.create_sheet('Sheet2')

with open('data2.csv', encoding = 'cp949') as f:
    reader = csv.reader(f)
    for row in reader:
        ws2.append(row)
```

5. 워크북 wb에 세 번째 워크시트 ws3를 생성하고 'data3.csv'의 내용을 작성.

```
ws3 = wb.create_sheet('Sheet3')

with open('data3.csv', encoding = 'cp949') as f:
    reader = csv.reader(f)
    for row in reader:
        ws3.append(row)
```

실전분석 예금 이자 계산기

[해결]

6. openpyxl 라이브러리의 함수 title을 사용하여 워크시트에 데이터 종류를 알기 쉽게 제목을 붙임. 그리고 save() 함수를 사용하여 엑셀 파일로 저장.

```
ws.title = '경기도 의정부시_의정부경전철 혼잡도'  
ws2.title = '대전교통공사_열차 혼잡도 분석'  
ws3.title = '서울특별시_지하철 혼잡도 정보'  
  
wb.save('traffic.xlsx')
```

7. 'traffic.xlsx' 파일을 다운로드하여 엑셀 프로그램으로 열기.

년월	요일	상_하행	열차번호	출발역	출발시간	도착역	도착시간	혼잡도	재차인원	영업일
2023-01	금요일	상행	1164	반석	17:55:00	판암	18:35:00	140	665	2022-0
2023-01	월요일	하행	1049	판암	08:18:00	반석	08:58:00	112	536	2022-0
2023-02	금요일	상행	1164	반석	17:55:00	판암	18:35:00	131	619	2022-0
2023-02	월요일	하행	1047	판암	08:52:00	반석	08:52:00	101	480	2022-0
2023-03	금요일	상행	1164	반석	17:55:00	판암	18:35:00	124	590	2022-0
2023-03	월요일	하행	1171	판암	17:55:00	반석	18:35:00	112	533	2022-0
2023-04	금요일	상행	1164	반석	17:55:00	판암	18:35:00	136	645	2022-0