

Section 02 CSV 파일 처리 방법

■ CSV 파일 개념

- [그림 6-1]의 CSV 파일은 [표 6-1]과 같은 형태의 데이터를 입력한 것
- 첫 행은 각 열의 값의 이름을 표현한 헤더(Header), 두 번째 행부터는 데이터 값

표 6-1 CSV로 생성할 표 데이터

아이디	이름	인원	주소	국번	전화 번호	평균 키	데뷔 일자
TWC	트와이스	9	서울	02	11111111	167	2015.10.19
BLK	블랙핑크	4	경남	055	22222222	163	2016.08.08
WMN	여자친구	6	경기	031	33333333	166	2015.01.15
OMY	오마이걸	7	서울			160	2015.04.21
GRL	소녀시대	8	서울	02	44444444	168	2007.08.02
ITZ	있지	5	경남			167	2019.02.12
RED	레드벨벳	4	경북	054	55555555	161	2014.08.01
APN	에이핑크	6	경기	031	77777777	164	2011.02.10
SPC	우주소녀	13	서울	02	88888888	162	2016.02.25
MMU	마마무	4	전남	061	99999999	165	2014.06.19

Section 02 CSV 파일 처리 방법

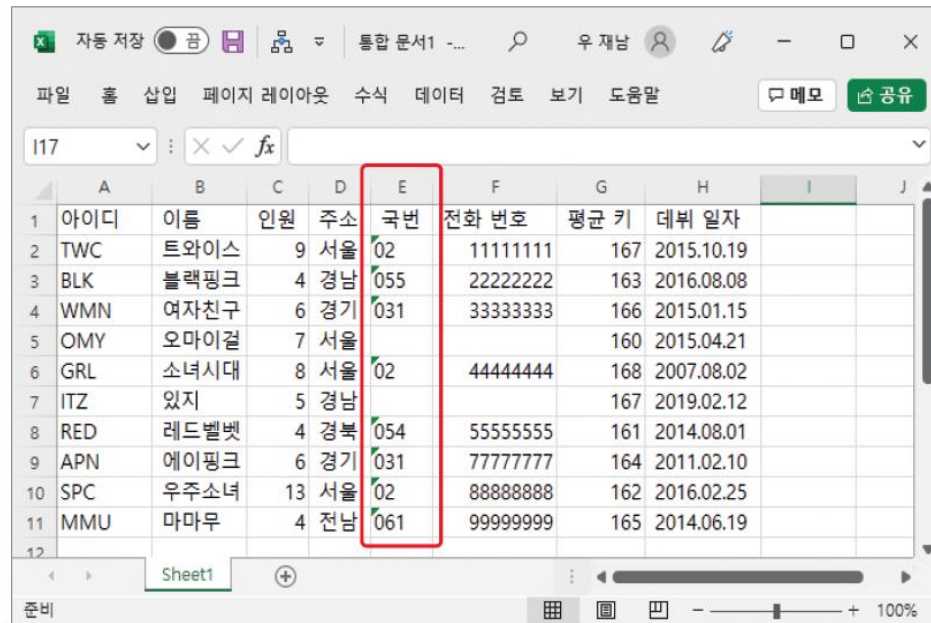
■ CSV 파일 개념

- CSV 파일을 메모장에서 직접 생성하거나 편집하는 것은 불편하고, 심표로 구분되어 있기 때문에 가독성이 떨어짐
- CSV를 직접 입력하여 생성할 때는 엑셀(Excel)을 사용하는 것이 편리함

Section 02 CSV 파일 처리 방법

■ CSV 파일 생성

- 엑셀을 실행해서 [표 6-1]을 다음과 같이 입력
- 국번 열의 경우 앞에 숫자 0이 들어가야 하므로 작은따옴표를 입력해야 한다는 점을 주의
(엑셀에서는 기본적으로 02를 입력해도 2로 저장되기 때문)



	A	B	C	D	E	F	G	H	I	J
1	아이디	이름	인원	주소	국번	전화 번호	평균 키	데뷔 일자		
2	TWC	트와이스	9	서울	02	11111111	167	2015.10.19		
3	BLK	블랙핑크	4	경남	055	22222222	163	2016.08.08		
4	WMN	여자친구	6	경기	031	33333333	166	2015.01.15		
5	OMY	오마이걸	7	서울			160	2015.04.21		
6	GRL	소녀시대	8	서울	02	44444444	168	2007.08.02		
7	ITZ	있지	5	경남			167	2019.02.12		
8	RED	레드벨벳	4	경북	054	55555555	161	2014.08.01		
9	APN	에이핑크	6	경기	031	77777777	164	2011.02.10		
10	SPC	우주소녀	13	서울	02	88888888	162	2016.02.25		
11	MMU	마마무	4	전남	061	99999999	165	2014.06.19		
12										

그림 6-2 엑셀에서 데이터 입력

Section 02 CSV 파일 처리 방법

■ CSV 파일 생성

- 입력을 완료하면 파일 형식을 CSV로 선택한 후 C:\CookAnalysis\CSV\singer1.csv로 저장 (폴더가 없다면 폴더를 먼저 생성)

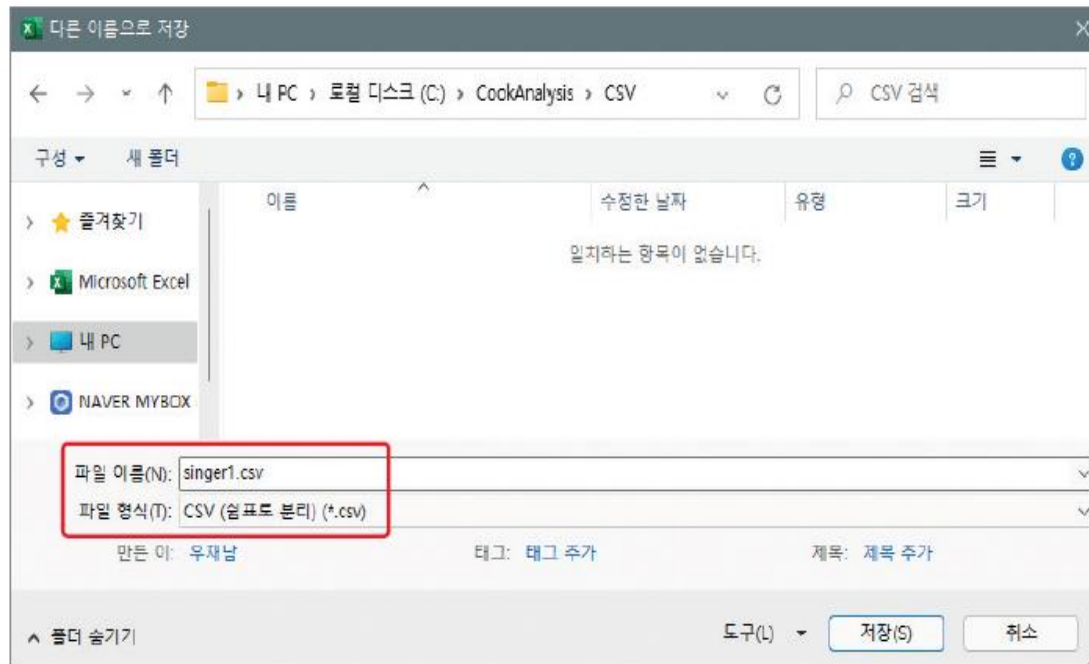


그림 6-3 CSV 파일로 저장

Section 02 CSV 파일 처리 방법

■ CSV 파일 생성

- 저장된 파일을 메모장에서 읽을 수 있음
- [파일 탐색기]에서 C:\₩CookAnalysis\₩CSV\₩singer1.csv 파일을 선택
- 마우스 오른쪽 버튼을 클릭 -> [연결 프로그램]-[메모장]을 선택

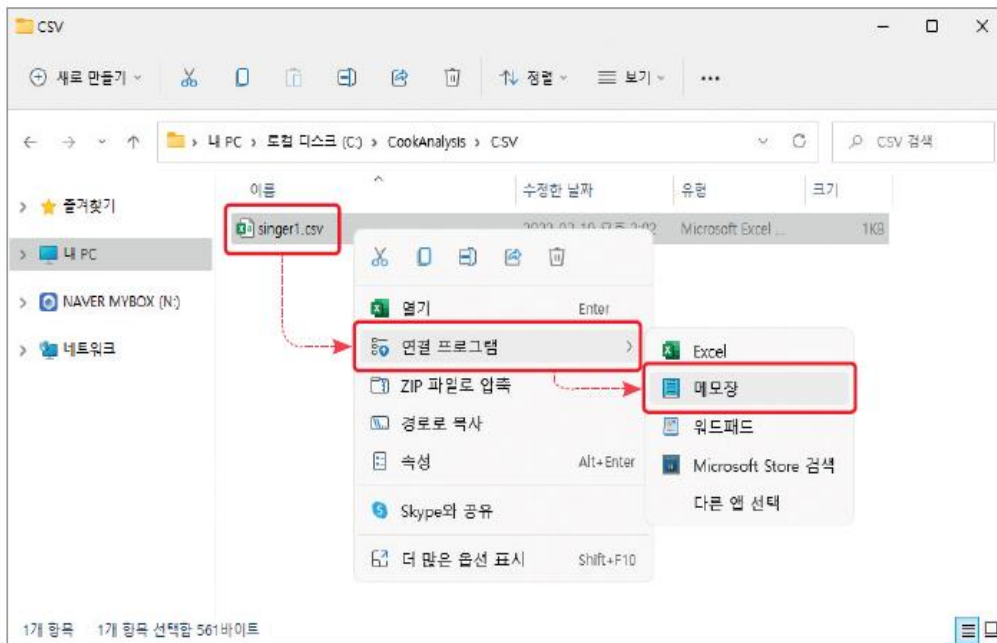


그림 6-4 CSV 파일을 메모장에서 열기

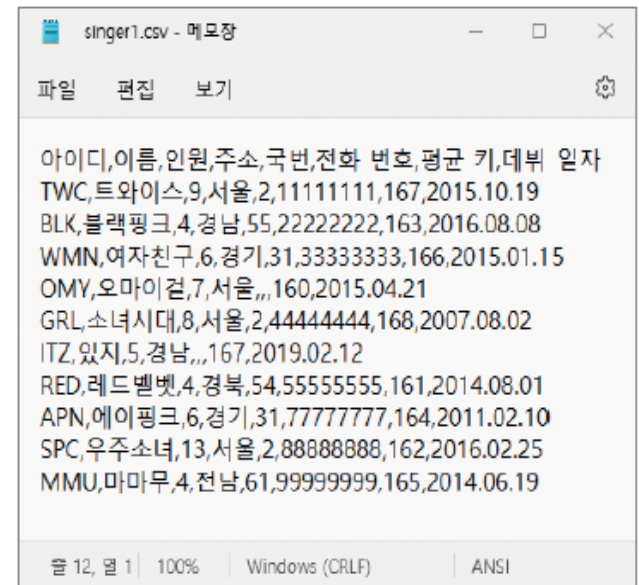


그림 6-5 메모장에서 열린 singer1.csv 파일

Section 02 CSV 파일 처리 방법

■ CSV 파일 처리

- with 예약어를 사용하면 Code06-01.py 9행의 파일을 닫는 과정이 생략되므로 편리함

Code06-02.py

```
0  with open("C:/CookAnalysis/CSV/singer1.csv", "r")
1
0      inStr = inFp.readline()
2      print(inStr, end = "")
0
3      inStr = inFp.readline()
0      print(inStr, end = "")
4
0
5
0
6
0
7
```

Section 02 CSV 파일 처리 방법

■ CSV 파일 처리

- CSV는 한 줄을 모두 처리하는 것이 아니라 각 데이터를 처리해야 의미가 있음
- 즉, TWC, 트와이스, 9 ... 등을 리스트에 분리해서 저장해야 이후 계산이 가능함

Section 02 CSV 파일 처리 방법

■ CSV 파일 처리

- CSV 파일의 헤더를 별도로 먼저 읽어서 처리한 후 나머지 모든 행을 리스트로 저장하고 각 항목을 분리해서 출력하는 코드

Code06-03.py

```
01 def printList(pList) :  
02     for data in pList :  
03         print(data, end='\t')  
04     print()  
05  
06 with open("C:/CookAnalysis/CSV/singer1.csv", "r") as inFp :  
07     header = inFp.readline()  
08     header = header.strip()  
09     header_list = header.split(',')  
10     printList(header_list)  
11     for inStr in inFp:  
12         inStr = inStr.strip()  
13         row_list = inStr.split(',')  
14         printList(row_list)
```


Section 02 CSV 파일 처리 방법

■ CSV 파일 처리

- 문자열에서 일부 문자를 변경하는 코드 (replace 함수를 사용)

```
myStr = "2025-5-5"  
myStr.replace('-', '#')
```

실행 결과

```
'2025#5#5'
```

- format()을 이용하여 정수 데이터 형식을 소수점이 들어간 문자열 형식으로 변경할 수 있음
- 두 번째 행은 "{순번:형식} {순번:형식}".format(값, 값, ...) 형식
- 형식은 정수인 경우엔 d, 실수인 경우엔 f를 사용
- 0.0f는 소수점 아래 자릿수가 0이라는 의미
- 0.2f는 소수점 아래 자릿수를 2자리까지 설정하라는 의미

```
num1, num2 = 1234.555, 100  
"{0:0.0f} {1:0.2f}".format(num1, num2)
```

실행 결과

```
'1235 100.00'
```

Section 02 CSV 파일 처리 방법

■ CSV 파일 처리

- 리스트를 문자열로 변경하려면 '구분자'.join()을 사용
- 리스트의 내용을 해시(#)로 구분해서 묶어줌

```
myList = ['파이썬', '데이터', '분석']  
'#'.join(myList)
```

실행 결과

```
'파이썬#데이터#분석'
```

- 숫자 리스트의 경우 join을 하기 전에 모두 문자열로 변경해야 함
- map(함수, 리스트)로 리스트에 함수가 한 번에 적용되는 방식을 사용하면 편리함
- map()은 myList의 각 항목마다 str() 함수를 한꺼번에 적용

```
myList = [100, 200, 300]  
list(map(str, myList))
```

실행 결과

```
['100', '200', '300']
```

Section 02 CSV 파일 처리 방법

■ CSV 파일 처리

- join()과 map()을 모두 사용하여 숫자 리스트를 문자열로 묶을 수 있음

```
myList = [2025, 8, 8]  
'/'.join(map(str, myList))
```

실행 결과

```
'2025/8/8'
```