

CSV & LOG관리

• CSV의 개념

- CSV(Comma Separate Values)는 콤마(,)를 기준으로 나누어진 값이라고 이해하면 된다.
- csv 파일은 어디서나 사용할 수 있도록 텍스트 데이터를 사용한다.

연도, 제조사, 모델, 설명, 가격

1997, Ford, E350, "ac, abs, moon", 3000.00

1999, Chevy, "Venture ""Extended Edition""", "", 4900.00

1999, Chevy, "Venture ""Extended Edition, Very Large""", "", 5000.00

1996, Jeep, Grand Cherokee, "MUST SELL! air, moon roof, loaded", 4799.00

➡ 위 데이터를 보면, 제일 상단에 필드(field), 헤더(header) 또는 열 이름(column name)이라고 부르는 텍스트 데이터가 입력되었다.

각 데이터는 콤마로 나뉘어진다.

두 번째 줄부터는 각 필드의 실제 데이터가 있는데, 각 행의 데이터가 인스턴스라고 이해하면 된다.

이러한 데이터를 행(row), 튜플(tuple), 인스턴스(instance) 등으로 부른다.

데이터의 분류는 분류 기준이 되는 문자에 따라 TSV(Tab Separate Values), SSV(Single-blank Separate Values) 등으로 구분한다.

- csv 파일 만들기

- csv 파일은 엑셀을 사용하여 간단히 만들 수 있다. 다음 순서대로 csv 파일을 만들어 보자.

① 'sample.xls' 파일을 다운로드 한다.

공공데이터 포털에서 데이터 엑셀파일을 받기.

서울 열린데이터 광장에서 엑셀 엑셀 받기.

LOCALDATA - 지방행정인허가데이터개방에서 엑셀 받기

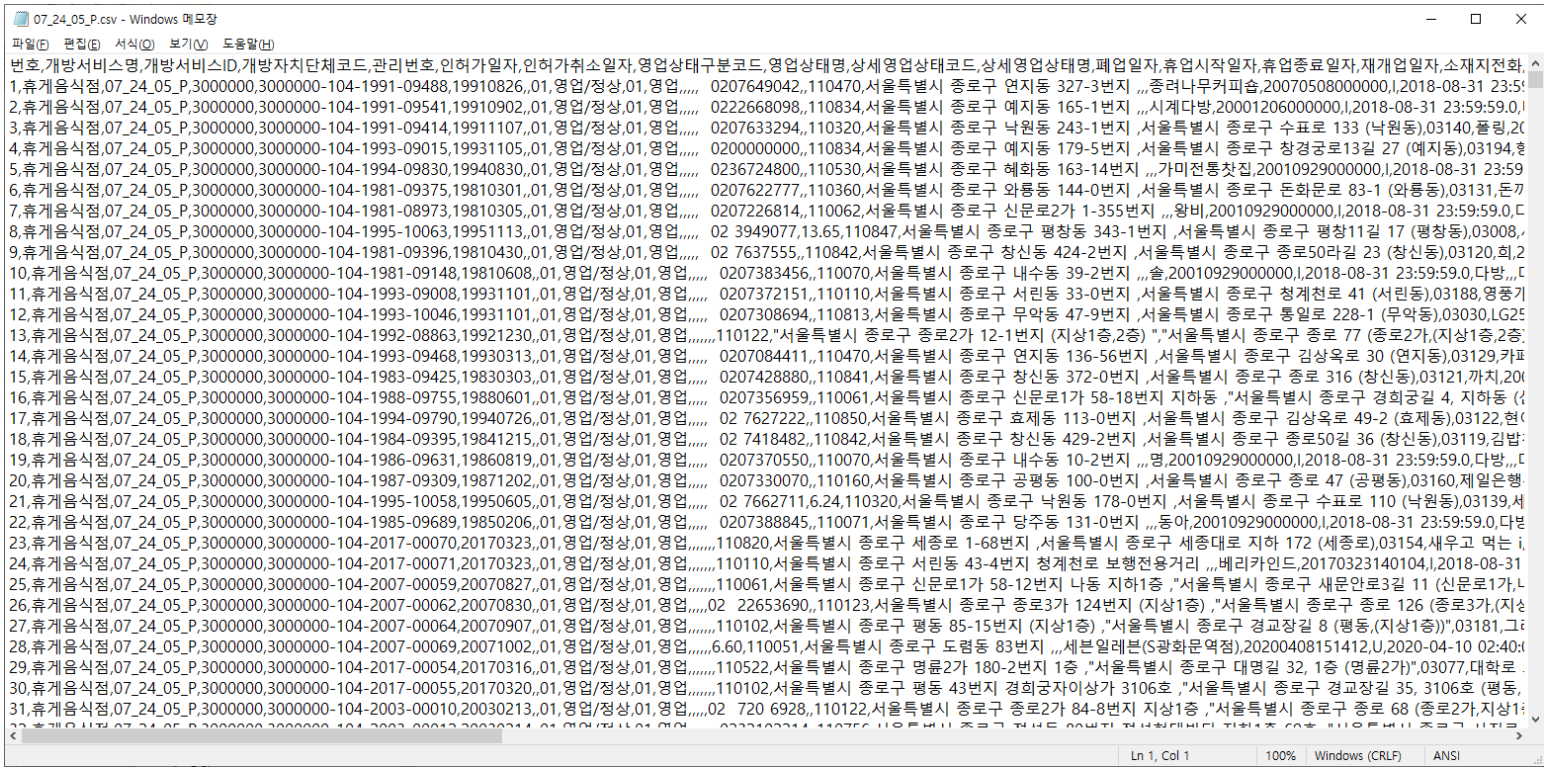
② 엑셀에서 다운로드한 파일을 연다.

③ 메뉴 바에서 [파일]-[다른 이름으로 저장]을 선택한다.

④ [다른 이름으로 저장] 대화상자에서 '파일 형식'을 'CSV(쉼표로 분리)'로 선택한 후, [저장]을 클릭한다.

⑤ 엑셀을 종료한 후, 메모장에서 파일을 연다.

- csv 파일 만들기
- 새롭게 생성된 'csv 파일 만들기.csv' 파일은 메모장 같은 텍스트 에디터로 내용을 확인할 수 있다.



[새로 만든 csv 파일을 메모장에서 확인하기]

- csv 파일 다루기
 - csv 파일을 다루기 위해 먼저 파이썬으로 csv 파일을 열고, 내용을 사용하는 작업을 한다.
 - 파이썬으로 csv 파일을 다루는 방법에는 크게 세 가지가 있다.
 - ① 파일 객체 사용하기
 - ② csv 객체 사용하기
 - ③ PANDAS 객체 사용하기

- csv 파일 다루기 : 파일 객체를 사용하여 데이터 다루기
 - 파일 객체를 사용하여 데이터를 다루는 방법은 일반적인 텍스트 파일을 처리하듯 파일을 읽어 온 후, 한 줄씩 데이터를 처리하는 것이다.
 - '07_24_05_P.csv' 파일을 사용하여 실습한다.

- csv 파일 다루기 : 파일 객체를 사용하여 데이터 다루기

```

line_counter = 0          # 파일의 총 줄 수를 세는 변수
data_header = []         # 데이터의 필드값을 저장하는 리스트
sample_list = []         # shop의 개별 리스트를 저장하는 리스트

# 07_24_05_P.csv 파일을 smaple_data 객체에 저장
with open("07_24_05_P.csv") as smaple_data:
    while True:
        # 07_24_05_P.csv에 데이터 변수에 한 줄씩 저장
        data = smaple_data.readline()
        if not data:
            break          # 데이터가 없을 때, 반복문 종료
        if line_counter == 0:
            # 첫 번째 데이터는 데이터의 필드
            # 데이터의 필드는 data_header 리스트에 저장, 데이터 저장 시 ","로 분리
            data_header = data.split(",")
        else:
            # 일반 데이터는 sample_list 객체에 저장, 데이터 저장 시 ","로 분리
            sample_list.append(data.split(","))
        line_counter += 1

#print("Header:", data_header)          # 데이터 필드값 출력
print("Header:", data_header[0:5])      # 필드를 5개만 출력
for i in range(0, 10):                  # 데이터 출력(샘플 10개만)
    #print("Data",i,":",sample_list[i][0:5])
    print("Data",i,":",sample_list[i][0:5])    #데이터를 5개만 출력

# 전체 데이터 크기 출력
print("전체 데이터 사이즈 :", len(sample_list))

```

- csv 파일 다루기 : 파일 객체를 사용하여 데이터 다루기

```
PS C:\Users\Wjin\Documents\Github\aijr202003\python> &
C:/Users/jin/AppData/Local/Programs/Python/Python38-32/python.exe
c:/Users/jin/Documents/Github/aijr202003/python/csv_controll_object.py
Header: ['번호', '개방서비스명', '개방서비스ID', '개방자치단체코드', '관리번호']
Data 0 : ['1', '휴게음식점', '07_24_05_P', '3000000', '3000000-104-1991-09488']
Data 1 : ['2', '휴게음식점', '07_24_05_P', '3000000', '3000000-104-1991-09541']
Data 2 : ['3', '휴게음식점', '07_24_05_P', '3000000', '3000000-104-1991-09414']
Data 3 : ['4', '휴게음식점', '07_24_05_P', '3000000', '3000000-104-1993-09015']
Data 4 : ['5', '휴게음식점', '07_24_05_P', '3000000', '3000000-104-1994-09830']
Data 5 : ['6', '휴게음식점', '07_24_05_P', '3000000', '3000000-104-1981-09375']
Data 6 : ['7', '휴게음식점', '07_24_05_P', '3000000', '3000000-104-1981-08973']
Data 7 : ['8', '휴게음식점', '07_24_05_P', '3000000', '3000000-104-1995-10063']
Data 8 : ['9', '휴게음식점', '07_24_05_P', '3000000', '3000000-104-1981-09396']
Data 9 : ['10', '휴게음식점', '07_24_05_P', '3000000', '3000000-104-1981-09148']
전체 데이터 사이즈 : 300001
```


- csv 파일 다루기 : 파일 객체를 사용하여 데이터 다루기
 - 읽은 정보를 새로운 csv 파일을 생성합니다.
 - 필요한 정보만 따로 리스트 객체에 저장하여 csv 파일에 쓰는 방식으로 작성.

- csv 파일 다루기 : 파일 객체를 사용하여 데이터 다루기

```

line_counter = 0          # 파일의 총 줄 수를 세는 변수
data_header = []         # 데이터의 필드값을 저장하는 리스트
new_shop_list = []       # 새로운 shop의 개별 리스트를 저장하는 리스트
shop = None

# 07_24_05_P.csv 파일을 smaple_data 객체에 저장
with open("07_24_05_P.csv") as smaple_data:
    while True:
        # 07_24_05_P.csv에 데이터 변수에 한 줄씩 저장
        data = smaple_data.readline()
        if not data:
            break          # 데이터가 없을 때, 반복문 종료
        if line_counter == 0:
            # 첫 번째 데이터는 데이터의 필드
            # 데이터의 필드는 data_header 리스트에 저장, 데이터 저장 시 ","로 분리
            data_header = data.split(",")
        else:
            # 일반 데이터는 sample_list 객체에 저장, 데이터 저장 시 ","로 분리
            shop = data.split(",")
            if len(shop) == 28 and shop[18] == '커피숍' :
                new_shop_list.append(shop)      # 업태 필드가 "커피숍" 것만 new_shop_list에 저장
            line_counter += 1

with open("new_shop_coffe.csv", "w", encoding='utf-8') as new_shop_coffe:
    for shop in new_shop_list:
        new_shop_coffe.write(",".join(shop).strip('\n')+"\n")
        # new_shop_list 객체에 있는 데이터를 customers_USA_only.csv 파일에 쓰기

print('파일 작성 완료')

```

• csv 파일 다루기 : 파일 객체를 사용하여 데이터 다루기

new_shop_coffe.csv - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

25,휴게음식점,07_24_05_P,3000000,3000000-104-2007-00059,20070827,,01,영업/정상,01,영업,110061,서울특별시	중로구	신문로1가 58-12번지 나동 지하1층 ,03175,커피온더힐,20070827100301,I,2018-08-31 23:59:59.0
26,휴게음식점,07_24_05_P,3000000,3000000-104-2007-00062,20070830,,01,영업/정상,01,영업,110123,서울특별시	중로구	종로3가 124번지 (지상1층) ,03192,이디야커피,20160624150745,I,2018-08-31 23:59:59.0
27,휴게음식점,07_24_05_P,3000000,3000000-104-2007-00064,20070907,,01,영업/정상,01,영업,110102,서울특별시	중로구	평동 85-15번지 (지상1층) ,03181,그라찌에 중앙대학교 평동캠퍼스점,20170106103827,I,2018-08-31 23:59:59.0
28,휴게음식점,07_24_05_P,3000000,3000000-104-2007-00069,20071002,,01,영업/정상,01,영업,110051,서울특별시	중로구	도렴동 83번지 ,,세븐일레븐(S광화문역점),20200408151412,U,2020-04-10 02:40:00.0,커피온
29,휴게음식점,07_24_05_P,3000000,3000000-104-2017-00054,20170316,,01,영업/정상,01,영업,110522,서울특별시	중로구	영륜2가 180-2번지 1층 ,03077,대학로 와플,20170316135735,I,2018-08-31 23:59:59.0,커피온
30,휴게음식점,07_24_05_P,3000000,3000000-104-2017-00055,20170320,,01,영업/정상,01,영업,110102,서울특별시	중로구	평동 43번지 경희궁자이상가 3106호 ,03180,커피베이 경희궁자이점,20170320094427,I,2018-08-31 23:59:59.0
32,휴게음식점,07_24_05_P,3000000,3000000-104-2003-00012,20030214,,01,영업/정상,01,영업,110756,서울특별시	중로구	적선동 80번지 적선현대빌딩 지하1층 69호 ,03170,에스프레소인뉴룩,20070305000000,I,2018-08-31 23:59:59.0
37,휴게음식점,07_24_05_P,3000000,3000000-104-2017-00074,20170323,,01,영업/정상,01,영업,110111,서울특별시	중로구	관철동 16-5번지 2층 ,03189,(주) 커피빈코리아 강각역점,20170323154537,I,2018-08-31 23:59:59.0
40,휴게음식점,07_24_05_P,3000000,3000000-104-2017-00038,20170223,,01,영업/정상,01,영업,110160,서울특별시	중로구	공평동 144번지 1층 105호,03158,빌리엔젤(강각역점),20170223102718,I,2018-08-31 23:59:59.0
41,휴게음식점,07_24_05_P,3000000,3000000-104-2017-00039,20170223,,01,영업/정상,01,영업,110160,서울특별시	중로구	공평동 144번지 1층103호 ,03158,디초콜릿커피&종로구청점,20170223145524,I,2018-08-31 23:59:59.0
42,휴게음식점,07_24_05_P,3000000,3000000-104-2017-00040,20170223,,01,영업/정상,01,영업,110040,서울특별시	중로구	통의동 108번지 1층 ,03044,마상스(Ma Chance),20170223165549,I,2018-08-31 23:59:59.0
44,휴게음식점,07_24_05_P,3000000,3000000-104-2018-00094,20180601,,01,영업/정상,01,영업,110796,서울특별시	중로구	신문로1가 24번지 고려빌딩 ,03182,(주)커피빈코리아 새문안로점,20180601095304,I,2018-08-31 23:59:59.0
46,휴게음식점,07_24_05_P,3000000,3000000-104-2018-00096,20180605,,01,영업/정상,01,영업,110833,서울특별시	중로구	예지동 25-2번지 ,03195,커피온리 광장시장점,20200522133758,U,2020-05-24 02:40:00.0,커피온
48,휴게음식점,07_24_05_P,3000000,3000000-104-2018-00100,20180611,,01,영업/정상,01,영업,110809,서울특별시	중로구	동숭동 1-93번지 ,03086,마스터키즈쿠링,20180611120752,I,2018-08-31 23:59:59.0,커피온
52,휴게음식점,07_24_05_P,3000000,3000000-104-2017-00047,20170228,,01,영업/정상,01,영업,110524,서울특별시	중로구	영륜4가 83-1번지 2~3층 ,03078,까페랑송,20200211182008,U,2020-02-13 02:40:00.0,커피온
53,휴게음식점,07_24_05_P,3000000,3000000-104-2018-00102,20180612,,01,영업/정상,01,영업,110035,서울특별시	중로구	육인동 153-1번지 ,03034,118COFFEE,20180629165616,I,2018-08-31 23:59:59.0,커피숍,1
54,휴게음식점,07_24_05_P,3000000,3000000-104-2018-00103,20180618,,01,영업/정상,01,영업,110150,서울특별시	중로구	중학동 19번지 더케이트원타워 ,03142,고디바케이트원타워,20180618155220,I,2018-08-31 23:59:59.0
55,휴게음식점,07_24_05_P,3000000,3000000-104-2018-00104,20180620,,01,영업/정상,01,영업,110045,서울특별시	중로구	체부동 209번지 ,03041,너만 모르는 밀크티,20190627093328,U,2019-06-29 02:40:00.0,커피온
58,휴게음식점,07_24_05_P,3000000,3000000-104-2018-00111,20180627,,01,영업/정상,01,영업,110800,서울특별시	중로구	계동 2-49번지 ,03057,부스갤러리(Booth gallery),20180627140744,I,2018-08-31 23:59:59.0
61,휴게음식점,07_24_05_P,3000000,3000000-104-2018-00114,20180629,,01,영업/정상,01,영업,110820,서울특별시	중로구	세종로 1-68번지 5호선 광화문역 ,03154,커피앤밀스,20181212160351,U,2018-12-14 02:40:00.0
63,휴게음식점,07_24_05_P,3000000,3000000-104-2018-00109,20180625,,01,영업/정상,01,영업,110530,서울특별시	중로구	혜화동 74-38번지 성인빌딩 ,03076,카페 브릿지(Cafe BRIDGE),20190111175232,U,2019-01-11 02:40:00.0
64,휴게음식점,07_24_05_P,3000000,3000000-104-2018-00110,20180625,,01,영업/정상,01,영업,110530,서울특별시	중로구	혜화동 179-4번지 ,03084,커피스미스 대학로 서경대점,20180806151218,I,2018-08-31 23:59:59.0
71,휴게음식점,07_24_05_P,3000000,3000000-104-2007-00049,20070709,,01,영업/정상,01,영업,110220,서울특별시	중로구	팔판동 72번지 (지하1층) ,03054,cafe ICAM,20101001154148,I,2018-08-31 23:59:59.0,커피온
73,휴게음식점,07_24_05_P,3000000,3000000-104-2017-00035,20170220,,01,영업/정상,01,영업,110863,서울특별시	중로구	송인동 107-5번지 ,03113,요거 프레스,20170220173813,I,2018-08-31 23:59:59.0,커피숍,2
76,휴게음식점,07_24_05_P,3000000,3000000-104-2018-00116,20180702,,01,영업/정상,01,영업,110035,서울특별시	중로구	육인동 149-1번지 ,03034,카페 봄,20181120105100,U,2018-11-22 02:37:36.0,커피숍,1970
77,휴게음식점,07_24_05_P,3000000,3000000-104-2018-00117,20180702,,01,영업/정상,01,영업,110521,서울특별시	중로구	영륜1가 56-4번지 ,03068,더런드리카페,20180702175358,I,2018-08-31 23:59:59.0,커피온
87,휴게음식점,07_24_05_P,3000000,3000000-104-2019-00209,20191031,,01,영업/정상,01,영업,110340,서울특별시	중로구	익선동 154-1번지 ,03133,빠송송,20191031154442,I,2019-11-02 00:23:04.0,커피숍,19909
97,휴게음식점,07_24_05_P,3000000,3000000-104-2020-00002,20200106,,01,영업/정상,01,영업,110122,서울특별시	중로구	종로2가 6번지 종로타워 ,03161,뉴룩송사탕,20200106114807,I,2020-01-08 00:23:39.0,커피온
98,휴게음식점,07_24_05_P,3000000,3000000-104-2020-00003,20200107,,01,영업/정상,01,영업,110801,서울특별시	중로구	계동 131-2번지 1층 ,03057,주식회사 하프 커피 로스터스,20200107145012,I,2020-01-09 00:23:39.0
105,휴게음식점,07_24_05_P,3000000,3000000-104-2019-00235,20191213,,01,영업/정상,01,영업,110030,서울특별시	중로구	청운동 108-5번지 ,03047,청운동 108,20200131152926,U,2020-02-02 02:40:00.0,커피숍
106,휴게음식점,07_24_05_P,3000000,3000000-104-2019-00213,20191113,,01,영업/정상,01,영업,110817,서울특별시	중로구	부암동 269-8번지 ,03022,서촌베이킹 스튜디오,20191113100525,I,2019-11-15 00:23:15
109,휴게음식점,07_24_05_P,3000000,3000000-104-2019-00212,20191112,,01,영업/정상,01,영업,110470,서울특별시	중로구	연지동 135번지 ,03128,카페온 바이 온라인프,20191112150900,I,2019-11-14 00:23:14.0
110,휴게음식점,07_24_05_P,3000000,3000000-104-2019-00100,20191011,,01,영업/정상,01,영업,110813,서울특별시	중로구	관동동 16번지 경희궁 롯데캐슬 2020010111141600,I,2020-01-01 00:23:14.0

Ln 1, Col 1 100% Windows (CRLF) UTF-8

- csv 파일 다루기 : csv 객체를 사용하여 데이터 다루기
 - 파이썬에서는 기본적으로 파일 객체가 아닌 csv 객체를 이용하여 csv 파일 형태의 데이터를 다루기 쉽다.
 - 'new_shop_coffe.csv' 파일을 사용한다.

- csv 파일 다루기 : csv 객체를 사용하여 데이터 다루기
 - csv 객체를 열기 위해서는 먼저 import문을 사용하여 csv 객체를 부르고, reader() 함수를 열어 각 속성(attribute)에 값을 넣으면 된다.

- csv 파일 다루기 : csv 객체를 사용하여 데이터 다루기

```
import csv # csv 객체 호출

seoul_location_data = [ ] # 기본 변수명 선언
header = [ ]
rownum = 0
# 불러들일 데이터를 선언함, 한글 처리를 위한 encoding = "utf-8"
with open("new_shop_coffe.csv", "r", encoding = "utf-8") as p_file:
    csv_data = csv.reader(p_file) # csv 객체를 이용해 csv_data 읽기, 특별히 데이터를 나누
    는 기준을 정하지 않음
    for row in csv_data: # 읽어 온 데이터를 한 줄씩 처리
        if rownum == 0: # 첫 번째 줄은 데이터 필드로 따로 저장
            header = row
            location = row[12]
            # '소재지 주소' 필드 데이터 추출, 한글 처리로 유니코드 데이터를 UTF-8로 변환
            if location.find(u"서울특별시")!=-1:
                # "행정구역" 데이터에 서울특별시가 들어가 있으면 seoul_location_data List에 추가
                seoul_location_data.append(row)
            rownum +=1

with open("new_shop_coffe_seoul_floating_data.csv", "w", encoding="utf8") as s_p_file:
    writer = csv.writer(s_p_file, delimiter='\t', quotechar='"', quoting=csv.QUOTE_ALL)
    # csv.writer를 사용해 csv 파일 만들기
    # delimiter 필드 구분자,
    # quotechar는 필드 각 데이터를 묶는 문자,
    # quoting은 묶는 범위
    writer.writerow(header) # 제목 필드 파일에 쓰기
    for row in seoul_location_data:
        writer.writerow(row) # seoul_location_data에 있는 정보를 리스트에 쓰기
    print('쓰기완료')
```

- csv 파일 다루기 : csv 객체를 사용하여 데이터 다루기

new_shop_coffe_seoul_floating_data.csv - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

'번호'	'개방서비스명'	'개방서비스ID'	'개방자치단체코드'	'관리번호'	'인가일자'	'인가취소일자'	'영업상태구분코드'	'영업상태명'	'상세영업상태코드'	'상세영업상태명'	'소재지'
'25'	'휴게음식점'	'07_24_05_P'	'3000000'	'3000000-104-2007-00059'	'20070827'	'"	'01'	'영업/정상'	'01'	'영업'	'110061' '서울특별시 종로구 신문로1가 56'
'26'	'휴게음식점'	'07_24_05_P'	'3000000'	'3000000-104-2007-00062'	'20070830'	'"	'01'	'영업/정상'	'01'	'영업'	'110123' '서울특별시 종로구 종로3가 124'
'27'	'휴게음식점'	'07_24_05_P'	'3000000'	'3000000-104-2007-00064'	'20070907'	'"	'01'	'영업/정상'	'01'	'영업'	'110102' '서울특별시 종로구 평동 85-15번'
'28'	'휴게음식점'	'07_24_05_P'	'3000000'	'3000000-104-2007-00069'	'20071002'	'"	'01'	'영업/정상'	'01'	'영업'	'110051' '서울특별시 종로구 도림동 83번'
'29'	'휴게음식점'	'07_24_05_P'	'3000000'	'3000000-104-2017-00054'	'20170316'	'"	'01'	'영업/정상'	'01'	'영업'	'110522' '서울특별시 종로구 명륜2가 180-
'30'	'휴게음식점'	'07_24_05_P'	'3000000'	'3000000-104-2017-00055'	'20170320'	'"	'01'	'영업/정상'	'01'	'영업'	'110102' '서울특별시 종로구 평동 43번지'
'32'	'휴게음식점'	'07_24_05_P'	'3000000'	'3000000-104-2003-00012'	'20030214'	'"	'01'	'영업/정상'	'01'	'영업'	'110756' '서울특별시 종로구 적선동 80번'
'37'	'휴게음식점'	'07_24_05_P'	'3000000'	'3000000-104-2017-00074'	'20170323'	'"	'01'	'영업/정상'	'01'	'영업'	'110111' '서울특별시 종로구 관철동 16-5'
'40'	'휴게음식점'	'07_24_05_P'	'3000000'	'3000000-104-2017-00038'	'20170223'	'"	'01'	'영업/정상'	'01'	'영업'	'110160' '서울특별시 종로구 공평동 144번'
'41'	'휴게음식점'	'07_24_05_P'	'3000000'	'3000000-104-2017-00039'	'20170223'	'"	'01'	'영업/정상'	'01'	'영업'	'110160' '서울특별시 종로구 공평동 144번'
'42'	'휴게음식점'	'07_24_05_P'	'3000000'	'3000000-104-2017-00040'	'20170223'	'"	'01'	'영업/정상'	'01'	'영업'	'110040' '서울특별시 종로구 통의동 108번'
'44'	'휴게음식점'	'07_24_05_P'	'3000000'	'3000000-104-2018-00094'	'20180601'	'"	'01'	'영업/정상'	'01'	'영업'	'110796' '서울특별시 종로구 신문로1가 24'
'46'	'휴게음식점'	'07_24_05_P'	'3000000'	'3000000-104-2018-00096'	'20180605'	'"	'01'	'영업/정상'	'01'	'영업'	'110833' '서울특별시 종로구 예지동 25-2'
'48'	'휴게음식점'	'07_24_05_P'	'3000000'	'3000000-104-2018-00100'	'20180611'	'"	'01'	'영업/정상'	'01'	'영업'	'110809' '서울특별시 종로구 동숭동 1-93'
'52'	'휴게음식점'	'07_24_05_P'	'3000000'	'3000000-104-2017-00047'	'20170228'	'"	'01'	'영업/정상'	'01'	'영업'	'110524' '서울특별시 종로구 명륜4가 83-1'
'53'	'휴게음식점'	'07_24_05_P'	'3000000'	'3000000-104-2018-00103'	'20180613'	'"	'01'	'영업/정상'	'01'	'영업'	'110835' '서울특별시 종로구 용인동 153'

Ln 1, Col 1 100% Macintosh (CR) UTF-8

Log 관리

- 로깅의 개념

- 이동이나 클릭 등 프로그램을 사용할 때 하는 모든 기본적인 이벤트(event)를 저장하는 것을 로그(log)정보를 저장한다고 한다.
- 프로그램이 실행되는 동안 일어나는 정보를 파일이나 기록으로 남기는 일을 로깅(logging)이라고 한다.
- 로그를 기록하는 가장 일반적인 방법은 파일을 생성하여 로그 정보를 남기는 것이다. 처음 프로그램을 실행할 때 로그 파일 하나를 생성하고, 그 후에 발생하는 이벤트를 로그 파일에 저장하는 방식이다.

로그 관리

- 기본 로그 관리 모듈: logging
 - 파이썬의 기본 로그 관리 모듈은 logging 모듈

```
import logging

logging.debug("틀렸습니다!")
logging.info("확인해주세요!")
logging.warning("조심해야합니다!")
logging.error("에러 났네요!!!")
logging.critical("크...허...망...")
```

로그 관리

- 기본 로그 관리 모듈: logging

- 파이썬 로깅을 실행할 때 로깅 레벨(logging level)을 지정할 수 있다. 이 로깅 레벨은 DEBUG, INFO, WARNING, ERROR, CRITICAL 등 총 5단계로 나눈다.

단계	개요	예시
DEBUG	개발 시 처리를 기록하는 로그 정보를 남김	<ul style="list-style-type: none">다음 함수로 A를 호출함변수 A를 ○○으로 변경함
INFO	처리가 진행되는 동안의 정보를 알림	<ul style="list-style-type: none">서버가 시작되었음서버가 종료됨사용자 A가 프로그램에 접속함
WARNING	사용자가 잘못 입력한 정보나 처리는 가능하지만 의도치 않은 정보가 들어 왔을 때 알림	<ul style="list-style-type: none">문자열 입력을 기대했으나, 정수형이 입력됨, 문자열 casting으로 처리함함수에 인수로 이차원 리스트를 기대했으나, 일차원 리스트가 들어옴, 이차원으로 변환 후 처리
ERROR	잘못된 처리로 에러가 발생하였지만, 프로그램은 동작할 수 있음을 알림	<ul style="list-style-type: none">파일에 기록해야 하는데 파일이 없음, 예외 처리 후 사용자에게 알림외부 서비스와 연결 불가
CRITICAL	잘못된 처리로 데이터가 손실되었거나 프로그램이 더는 동작할 수 없음을 알림	<ul style="list-style-type: none">잘못된 접근으로 해당 파일이 삭제됨사용자에 의한 강제 종료

로그 관리

- 기본 로그 관리 모듈: logging
 - 파이썬에서 로깅을 사용하기 위해서는 Logger 객체를 활용해야 한다.

```
import logging

# Logger 선언 : Logger 객체 획득
logger = logging.getLogger("main")
# Logger의 출력 방법 선언 : 파일, 화면, 다른 네트워크 등을 선택
stream_handler = logging.StreamHandler()
# Logger의 출력 등록 : 핸들러를 로거에 등록
logger.addHandler(stream_handler)

#출력 레벨 설정
logger.setLevel(logging.DEBUG)

logger.debug("틀렸습니다!")
logger.info("확인해주세요!")
logger.warning("조심해야합니다!")
logger.error("에러 났네요!!!")
logger.critical("크...허...망...")
```



Logging Program

로깅 프로그램

- logging formatter
 - formatter는 로그의 결과 값을 일정 형식을 지정하여 출력하는 기능이다

```
import logging

# Logger 생성
logger = logging.getLogger('myapp')
# FileHandler 생성
myapp_handler = logging.FileHandler('myapp.log')

# Logging Formatter 생성: 시간, 로깅 레벨, 프로세스ID, 메시지
formatter = logging.Formatter('%(asctime)s %(levelname)s %(process)d%(message)s')

# FileHandler에 formatter 등록
myapp_handler.setFormatter(formatter)
# Logger에 'FileHandler' 등록
logger.addHandler(myapp_handler)
# 로깅 레벨 설정
logger.setLevel(logging.INFO)

# 로깅 정보 출력
logger.error('ERROR occurred')
logger.info('HERE WE ARE')
logger.info('TEST finished')
```

loggingformatter.py

myapp.log ×

myapp.log

```
1 2020-09-02 21:47:22,738 ERROR 345828ERROR occurred
2 2020-09-02 21:47:22,739 INFO 345828HERE WE ARE
3 2020-09-02 21:47:22,739 INFO 345828TEST finished
4
```

로깅 프로그램

- logging formatter
 - logging.conf file은 파일에 로깅과 관련된 정보를 저장하여, 실제 사용할 때 파일을 생성하는 것만으로 로깅을 설정하는 방법으로 긴 코드를 쓰지 않고 매우 간단하게 설정

```
[loggers]
keys=root

[handlers]
keys=consoleHandler

[formatters]
keys=simpleFormatter

[logger_root]
level=DEBUG
handlers=consoleHandler

[handler_consoleHandler]
class=StreamHandler
level=DEBUG
formatter=simpleFormatter
args=(sys.stdout,)

[formatter_simpleFormatter]
format=%(asctime)s - %(name)s - %(levelname)s - %(message)s
datefmt=
```

로깅 프로그램

- logging formatter

➡ 실제 설정 파일을 사용하기 위해서는 다음과 같이 설정 파일을 호출한다.

```
logging.config.fileConfig('logging.conf')  
logger = logging.getLogger()
```


로깅 프로그램

- 실습 : 앞서 작성해본 예제를 기반으로 로깅 처리

```
import logging          # logging
import logging.config    # logging.config
import csv               # csv 객체 호출

# 모듈 호출 : logging
# Logger 생성
logging.config.fileConfig('logging.conf')
logger = logging.getLogger()

seoul_location_data = [ ]          # 기본 변수명 선언
header = [ ]
rownum = 0

# 변수 선언 등 생략
logger.info('Open file {0}'.format("TEST",))
```

로깅 프로그램

- 실습 : 앞서 작성해본 예제를 기반으로 로깅 처리

```
try:
    with open("new_shop_coffe.csv", "r", encoding = "utf-8") as p_file:
        csv_data = csv.reader(p_file)
        for row in csv_data:
            if rownum == 0:
                header = row
                location = row[12]
                if location.find(u"서울특별시")!=-1:
                    # 행 추가 시 로깅
                    logger.info('ID {0} added'.format(row[0],))
                    seoul_location_data.append(row)
                rownum +=1
except FileNotFoundError as ex:
    # 파일 open 시의 예외 발생 시 로깅
    logger.error('File NOT found {0}'.format(e,))

# 파일 생성 시작을 확인 하는 로깅
logger.info('Write new_shop_coffe_seoul only data at {0}'.format("new_shop_coffe_seoul_floating_data.csv",))

with open("new_shop_coffe_seoul_floating_data.csv", "w", encoding="utf8") as s_p_file:
    writer = csv.writer(s_p_file, delimiter='\t', quotechar="'", quoting=csv.QUOTE_ALL)
    writer.writerow(header)
    for row in seoul_location_data:
        writer.writerow(row)
    print('쓰기완료')
    logger.info('Write finished')
logger.info('Program finished')

# 파일 쓰기 완료 로깅
# 프로그램 종료 로깅
```

로깅 프로그램

- 실습

2020-09-02 22:04:33,593 - root - INFO - Open file TEST

2020-09-02 22:04:33,594 - root - INFO - ID 25 added

2020-09-02 22:04:33,594 - root - INFO - ID 26 added

...

중간 생략

...

2020-09-02 22:01:10,303 - root - INFO - ID 106801 added

2020-09-02 22:01:10,303 - root - INFO - ID 106828 added

2020-09-02 22:01:10,304 - root - INFO - ID 106840 added

2020-09-02 22:01:10,304 - root - INFO - ID 106855 added

2020-09-02 22:01:10,304 - root - INFO - ID 106873 added

2020-09-02 22:01:10,655 - root - INFO - Write new_shop_coffe_seoul only data at new_shop_coffe_seoul_floating_data.csv

쓰기완료

2020-09-02 22:01:10,840 - root - INFO - Write finished

2020-09-02 22:01:10,840 - root - INFO - Program finished

정리합시다

- CSV
- 로그관리
- 설정 저장
- logging Program