```
Lab. CSV file 다루기

1. csv module을 사용하지 않는 기본 python 코드
input_file = 'supplier_data.csv'
output_file = 'output.csv'

with open(input_file, 'r', newline='') as filereader:
   with open(output_file, 'w', newline='') as filewriter:
      header = filereader.readline()
      header = header.strip()
      header_list = header.split(',')
      print(header_list)
      filewriter.write(','.join(map(str,header_list))+'\n')
      for row in filereader:
         row = row.strip()
         row_list = row.split(',')
         print(row_list)
         filewriter.write(','.join(map(str,row_list))+'\n')


2. pandas를 이용한 CSV file 처리
   import pandas as pd

   input_file = 'supplier_data.csv'
   output_file = 'output1.csv'

   data_frame = pd.read_csv(input_file)
   print(data_frame)
   data_frame.to_csv(output_file, index=False)


3. csv module을 사용한 기본 python 코드
   import csv

   input_file = 'supplier_data.csv'
   output_file = 'output2.csv'

   with open(input_file, 'r', newline='') as csv_in_file:
       with open(output_file, 'w', newline='') as csv_out_file:
           filereader = csv.reader(csv_in_file, delimiter=',')
           filewriter = csv.writer(csv_out_file, delimiter=',')
           for row_list in filereader:
               print(row_list)
               filewriter.writerow(row_list)


4. 특정 조건을 만족하는 행의 filtering을 기본 python code로 구현
   import csv

   input_file = 'supplier_data.csv'
   output_file = 'output3.csv'
```

```
52
53    with open(input_file, 'r', newline='') as csv_in_file:
54        with open(output_file, 'w', newline='') as csv_out_file:
55            filereader = csv.reader(csv_in_file)
56            filewriter = csv.writer(csv_out_file)
57            header = next(filereader)
58            filewriter.writerow(header)
59            for row_list in filereader:
60                supplier = str(row_list[0]).strip()
61                cost = str(row_list[3]).strip('$').replace(',', '')
62                if supplier == 'Supplier Z' or float(cost) > 600.0:
63                    #cost가 $600.00 이상인 행만 필터링
64                    print(row_list)
65                    filewriter.writerow(row_list)
66
67
68  5. 특정 조건을 만족하는 행의 filtering을 pandas의 loc()로 구현
69    import pandas as pd
70
71    input_file = 'supplier_data.csv'
72    output_file = 'output4.csv'
73
74    data_frame = pd.read_csv(input_file)
75
76    data_frame['Cost'] = data_frame['Cost'].str.strip('$').astype(float)
77    data_frame_value_meets_condition = data_frame.loc[(data_frame['Supplier Name']\
78    .str.contains('Z')) | (data_frame['Cost'] > 600.0), :]
79
80    data_frame_value_meets_condition.to_csv(output_file, index=False)
81
82
83  6. 정규표현식을 활용한 filtering을 기본 python code로 구현
84    import re
85    import csv
86
87    input_file = 'supplier_data.csv'
88    output_file = 'output5.csv'
89
90    pattern = re.compile(r'(?P<my_pattern_group>^001-.*)', re.I)
91
92    with open(input_file, 'r', newline='') as csv_in_file:
93       with open(output_file, 'w', newline='') as csv_out_file:
94          filereader = csv.reader(csv_in_file)
95          filewriter = csv.writer(csv_out_file)
96          header = next(filereader)
97          filewriter.writerow(header)
98          for row_list in filereader:
99             invoice_number = row_list[1]
100            if pattern.search(invoice_number):
101               filewriter.writerow(row_list)
102
```

103
104 7.
105 import pandas as pd
106
107 input_file = 'supplier_data.csv'
108 output_file = 'output6.csv'
109
110 data_frame = pd.read_csv(input_file)
111 data_frame_value_matches_pattern = data_frame.loc[data_frame['Invoice Number']\
112 .str.startswith("001-"), :]
113
114 data_frame_value_matches_pattern.to_csv(output_file, index=False)