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'

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)
        filewriter = csv.writer(csv_out_file)
        header = next(filereader)
        filewriter.writerow(header)
        for row_list in filereader:
            supplier = str(row_list[0]).strip()
            cost = str(row_list[3]).strip('$').replace(',', '')
            if supplier == 'Supplier Z' or float(cost) > 600.0:
                #cost가 $600.00 이상인 행만 필터링
                print(row_list)
                filewriter.writerow(row_list)
```

5. 특정 조건을 만족하는 행의 filtering을 pandas의 loc()로 구현

```
import pandas as pd

input_file = 'supplier_data.csv'
output_file = 'output4.csv'

data_frame = pd.read_csv(input_file)

data_frame['Cost'] = data_frame['Cost'].str.strip('$').astype(float)
data_frame_value_meets_condition = data_frame.loc[(data_frame['Supplier Name']\
.str.contains('Z')) | (data_frame['Cost'] > 600.0), :]

data_frame_value_meets_condition.to_csv(output_file, index=False)
```

6. 정규표현식을 활용한 filtering을 기본 python code로 구현

```
import re
```

```
    import csv

    input_file = 'supplier_data.csv'
    output_file = 'output5.csv'

    pattern = re.compile(r'(?P<my_pattern_group>^001-.*)', re.I)

    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)
            filewriter = csv.writer(csv_out_file)
            header = next(filereader)
            filewriter.writerow(header)
            for row_list in filereader:
                invoice_number = row_list[1]
                if pattern.search(invoice_number):
                    filewriter.writerow(row_list)


7. pandas로 구현하기
    import pandas as pd

    input_file = 'supplier_data.csv'
    output_file = 'output6.csv'

    data_frame = pd.read_csv(input_file)
    data_frame_value_matches_pattern = data_frame.loc[data_frame['Invoice Number']\
    .str.startswith("001-"), :]

    data_frame_value_matches_pattern.to_csv(output_file, index=False)
```