

1. What is CSV format?

The so-called CSV(Comma Separated Values) format is the most common import and export format for spreadsheets and databases. The csv module implements classes to read and write tabular data in CSV format. It allows programmers to say, write this data in the format preferred by Excel, or read data from this file which was generated by Excel, without knowing the precise details of the CSV format used by Excel.

2. How to use CSV module?

csv.reader(csvfile)

Return a reader object which will iterate over lines in the given csvfile. csvfile can be any object which supports the iterator protocol and returns a string each time its `__next__()` method is called. file object and list object are both suitable.

```
# Import library
import csv

with open('file.csv', 'r') as csv_file:
    csv_reader = csv.reader(csv_file)

    # This will skip first line of csv_reader
    ## next(csv_reader)

    rows = [row for row in csv_reader]
```

3. Write file with new delimiter

```
import csv

with open('file.csv', 'r') as csv_file:
    csv_reader = csv.reader(csv_file)

    # Create new file with name as new_file.csv
    with open('new_file.csv', 'w') as new_file:
        csv_writer = csv.writer(new_file, delimiter='\t')

        # Write each line of csv_reader
        for line in csv_reader:
            csv_writer.writerow(line)
```

4. Read and Write Files with dictionary reader

This way is more obvious what we are doing.

```
import csv

with open('file.csv', 'r') as csv_file:
    csv_reader = csv.DictReader(csv_file)

    # We don't have to find the exact index number of column
    for line in csv_reader:
        print(line['first_name'])

    with open('new_file.csv', 'w') as new_file:
        fieldnames = ['first_name', 'last_name', 'email']
        csv_writer = csv.DictWriter(new_file, fieldnames=fieldnames, delimiter='\t')

        csv_writer.writeheader()

        for line in csv_reader:
            csv_writer.writerow(line)
```