Reading and Writing Delimited Files Using Python

There are various ways to read and write a delimited file using Python. The Python Pandas library provides methods to read and write a delimited file easily.

- read_csv() method is used to read a delimited file. It imports data in the form of a data frame. Refer to the official documentation of read_csv().
- to_csv() method is used to write a Pandas data frame to a delimited file. Refer to the official documentation to_csv().

A data frame is a two-dimensional data structure in which the data is organized in a tabular fashion having rows and columns.
Consider a CSV file *users.csv* with the following contents:

```
id,name,age
1,John,35
2,San,30
3,Mary,31
```

Use the following Python code example to read the *users.csv* file as a data frame and display its contents.

Table 2-18

Python code to read CSV file

```
import pandas

# Read the CSV file
user_df = pandas.read_csv('users.csv')

# Print the contents of the CSV file
print(user_df)
```

Output data frame

	Id	name	age
0	1	John	35
1	2	San	30
2	3	Mary	31

Use the following Python code example to write the data frame to a file *output_users.csv*.

Table 2-19

Python code to write a CSV file

```
import pandas

# Create a data frame.
users = {'id': [1, 2, 3], 'name': ['John', 'San', 'Mary'], 'age': [35, 30, 31]}
users_df = pandas.DataFrame(users )
```

```
# Write to a CSV file

users_df.to_csv('output_users.csv', index=False)
```

Both Pandas read_csv() and to_csv() methods consider comma ',' to be the delimiter by default. However, you can specify any delimiter explicitly using the sep parameter. The following example uses the pipe '|' as a delimiter.

```
pandas.read_csv('users.txt', sep='|')
```

```
df.to_csv('out_users.csv', sep='|', index=False)
```