

What is a CSV File?

A CSV file is a simple file format used to store tabular data, such as spreadsheets or databases. It stores data (numbers and text) in plain text, with each line representing a data record. Fields within each record are separated by commas, which is why it's called "Comma-Separated Values."

1. Reading CSV Files

To read data from a CSV file, you can use the `csv.reader` object. For example:

```
1. import csv
2.
3. # Open the CSV file in read mode
4. with open('data.csv', 'r') as file:
5.     # Create a CSV reader object
6.     csv_reader = csv.reader(file)
7.
8.     # Iterate over each row in the CSV file
9.     for row in csv_reader:
10.         print(row)
```

2. Writing CSV Files

To write data to a CSV file, you can use the `csv.writer` object. For example:

1. To write multiple rows at once: use `writerows()`

```
2. import csv
3.
4. # Data to be written to the CSV file
5. data = [
6.     ['Name', 'Age', 'Country'],
7.     ['John', 30, 'USA'],
8.     ['Emily', 25, 'UK'],
9.     ['Max', 35, 'Canada']
10.]
11.
12.# Open the CSV file in write mode
13.with open('output.csv', 'w', newline='') as file:
14.    # Create a CSV writer object
15.    csv_writer = csv.writer(file)
16.
17.    # Write data to the CSV file
18.    csv_writer.writerows(data)
```

2. To write single row at a time, use `writerow()`

```
1. import csv
2.
3. # Open the CSV file in write mode
4. with open('output.csv', 'w', newline='') as file:
5.     # Create a CSV writer object
6.     csv_writer = csv.writer(file)
7.
8.     # Write a single row to the CSV file
9.     csv_writer.writerow(['Name', 'Age', 'Country'])
10.    csv_writer.writerow(['John', 30, 'USA'])
```

3. use of next() method in CSV file handling

The **next()** method in Python is used to retrieve the next item from an iterator. In the context of CSV processing with Python's **csv** module, the **next()** method is commonly used with CSV reader objects to skip the header row. Here's an example:

```
1. import csv
2.
3. # Open the CSV file in read mode
4. with open('data.csv', 'r') as file:
5.     # Create a CSV reader object
6.     csv_reader = csv.reader(file)
7.
8.     # Skip the header row
9.     header = next(csv_reader)
10.
11. # Iterate over each row in the CSV file
12. for row in csv_reader:
13.     print(row)
```

4. line_num attribute in csv

In the context of the **csv.reader** object in Python's **csv** module, the **line_num** attribute is an integer that represents the current line number being read from the CSV file.

Example 1:

```
1. import csv
2.
3. filename = "students.csv"
4.
5. with open(filename, 'r') as csvfile:
6.     csvreader = csv.reader(csvfile)
7.     print(csvreader.line_num)
8.
9.     next(csvreader)
10.    print(csvreader.line_num)
11.
12.    next(csvreader)
13.    print(csvreader.line_num)
```

Example 2:

```
1. import csv
2.
3. # Open the CSV file in read mode
4. with open('data.csv', 'r') as file:
5.     # Create a CSV reader object
6.     csv_reader = csv.reader(file)
7.
8.     # Iterate over each row in the CSV file
9.     for row in csv_reader:
10.        print(f"Line {csv_reader.line_num}: {row}")
```

Example to read a csv file and print the first 5 rows

```
1. import csv
2.
3. # Specify the CSV file name
4. filename = "aapl.csv"
5.
6. # Initialize lists for fields and rows
7. fields = []
8. rows = []
9.
10.# Read the CSV file
11.with open(filename, 'r') as csvfile:
12.    csvreader = csv.reader(csvfile)
13.    fields = next(csvreader) # Extract field names from the first row
14.    for row in csvreader:
15.        rows.append(row)
16.
17.# Print total number of rows
18.print(f"Total no. of rows: {csvreader.line_num}")
19.
20.# Print field names
21.print("Field names are:", ', '.join(fields))
22.
23.# Print the first 5 rows
24.print("\nFirst 5 rows are:")
25.for row in rows[:5]:
26.    for col in row:
27.        print(f"{col:10}", end=" ")
28.    print()
```