



# CSV Module

# CSV module



A CSV (Comma Separated Values) format is one of the most simple and common ways to store tabular data. To represent a CSV file, it must be saved with the `.csv` file extension.

## Working with CSV files in Python

While we could use the built-in `open()` function to work with CSV files in Python, there is a dedicated `csv` module that makes working with CSV files much easier.

Before we can use the methods to the `csv` module, we need to import the module first using:

```
import csv
```

## Reading CSV files Using `csv.reader()`

To read a CSV file in Python, we can use the `csv.reader()` function. Suppose we have a `csv` file named **people.csv** in the current directory with the following entries.

Name	Age	Profession
Jack	23	Doctor
Miller	22	Engineer

```
import csv
with open('people.csv', 'r') as file:
    reader = csv.reader(file)
    for row in reader:
        print(row)
```

### Output

```
['Name', 'Age', 'Profession']
['Jack', '23', 'Doctor']
['Miller', '22', 'Engineer']
```

## Writing CSV files Using `csv.writer()`

To write to a CSV file in Python, we can use the `csv.writer()` function.

The `csv.writer()` function returns a `writer` object that converts the user's data into a delimited string. This string can later be used to write into CSV files using the `writerow()` function. Let's take an example.

```
import csv
with open('protagonist.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(["SN", "Movie", "Protagonist"])
    writer.writerow([1, "Lord of the Rings", "Frodo Baggins"])
    writer.writerow([2, "Harry Potter", "Harry Potter"])
```

When we run the above program, a **protagonist.csv** file is created with the following content:

```
SN,Movie,Protagonist
1,Lord of the Rings,Frodo Baggins
2,Harry Potter,Harry Potter
```

## Writing multiple rows with writerows()

```
import csv

csv_rowlist = [{"SN": "Movie", "Protagonist": "Frodo Baggins"}, [{"SN": "1", "Movie": "Lord of the Rings", "Protagonist": "Frodo Baggins"}, {"SN": "2", "Movie": "Harry Potter", "Protagonist": "Harry Potter"}]]

with open('protagonist.csv', 'w') as file:
    writer = csv.writer(file)
    writer.writerows(csv_rowlist)
```

## Python csv.DictReader() Class

The objects of a `csv.DictReader()` class can be used to read a CSV file as a dictionary.

### Example 6: Python csv.DictReader()

Suppose we have the same file **people.csv** as in **Example 1**.

Name	Age	Profession
Jack	23	Doctor
Miller	22	Engineer

Let's see how `csv.DictReader()` can be used.

```
import csv
with open("people.csv", 'r') as file:
    csv_file = csv.DictReader(file)
    for row in csv_file:
        print(dict(row))
```

## Example 7: Python csv.DictWriter()

```
import csv

with open('players.csv', 'w', newline='') as file:
    fieldnames = ['player_name', 'fide_rating']
    writer = csv.DictWriter(file, fieldnames=fieldnames)

    writer.writeheader()
    writer.writerow({'player_name': 'Magnus Carlsen', 'fide_rating': 2870})
    writer.writerow({'player_name': 'Fabiano Caruana', 'fide_rating': 2822})
    writer.writerow({'player_name': 'Ding Liren', 'fide_rating': 2801})
```

The program creates a **players.csv** file with the following entries:

```
player_name,fide_rating
Magnus Carlsen,2870
Fabiano Caruana,2822
Ding Liren,2801
```