

# Unit 6 Lesson 2: Writing CSV Files

## Quick Reference Guide

---

### Import

```
import csv
```

---

### Create New CSV File (Write Mode)

```
data = [  
    {'name': 'Alice', 'score': 100},  
    {'name': 'Bob', 'score': 85}  
]  
  
with open("output.csv", "w", newline="") as file:  
    fieldnames = ['name', 'score']  
    writer = csv.DictWriter(file, fieldnames=fieldnames)  
  
    writer.writeheader()    # Write column names  
    writer.writerows(data)  # Write all data
```

#### Result file:

```
name,score  
Alice,100  
Bob,85
```

---

## + Append to Existing File

```
new_record = {'name': 'Charlie', 'score': 92}

with open("output.csv", "a", newline="") as file:
    fieldnames = ['name', 'score']
    writer = csv.DictWriter(file, fieldnames=fieldnames)

    writer.writerow(new_record) # NO header!
```

## Key Rules

Rule	Why It Matters
<b>Always use</b> <code>newline=""</code>	Prevents extra blank lines on Windows
<b>"w" mode overwrites</b>	Existing data is deleted!
<b>"a" mode appends</b>	Adds to end, keeps existing data
<b>No header when appending</b>	Header is already in the file

## writerow vs writerows

```
# writerow() - One record at a time
writer.writerow({'name': 'Alice', 'score': 100})

# writerows() - Multiple records at once
writer.writerows(list_of_dicts)
```

# Read-Modify-Write Pattern

```
import csv

# 1. READ - Load existing data
with open("data.csv", "r") as file:
    data = list(csv.DictReader(file))

# 2. MODIFY - Change the data
for item in data:
    item['value'] = int(item['value']) + 100

# 3. WRITE - Save changes back
with open("data.csv", "w", newline="") as file:
    fieldnames = ['col1', 'col2', 'value']
    writer = csv.DictWriter(file, fieldnames=fieldnames)
    writer.writeheader()
    writer.writerows(data)
```

## File Modes Summary

Mode	Creates?	Existing Data	Use For
"w"	Yes	Deleted	New files, complete rewrites
"a"	Yes	Kept	Adding records
"r"	No	Read only	Loading data

## fieldnames

The `fieldnames` list controls:

1. **Which keys** are written from each dictionary
2. **Column order** in the CSV file

```
fieldnames = ['username', 'level', 'score']  
# Columns will appear in this exact order
```

## ✖ Common Mistakes

### Forgetting newline=""

```
# ✖ Wrong  
with open("data.csv", "w") as file:  
  
# ✔ Correct  
with open("data.csv", "w", newline="") as file:
```

### Writing header when appending

```
# ✖ Wrong - header appears multiple times!  
with open("data.csv", "a", newline="") as file:  
    writer.writeheader() # DON'T do this!  
    writer.writerow(record)  
  
# ✔ Correct  
with open("data.csv", "a", newline="") as file:  
    writer.writerow(record) # Just the data
```

### Using "w" when you meant "a"

```
# ✖ DANGER - deletes all existing data!  
with open("scores.csv", "w", newline="") as file:  
  
# ✔ Safe - adds to existing data  
with open("scores.csv", "a", newline="") as file:
```