**Topic 72: CSV Files: Loading Information into Them (Part 3)**

**What**  
You can write rows of data into a CSV file using Python’s csv module, which helps manage the process of adding structured data to CSV files.

**Why**  
To save and manage structured data (like spreadsheets) in a CSV format, making it easy to store, share, and import/export between programs like Excel or databases.

**How**

1. **Open the file for writing**:  
   First, open (or create) the CSV file for writing:

python

Copy code

with open("whatever.csv", "w", newline="") as f:

This opens the file in write mode. The newline="" ensures proper handling of newlines in CSV files.

1. **Create a CSV writer object**:  
   Use csv.writer() to handle writing to the file:

python

Copy code

data\_handler = csv.writer(f, delimiter=",")

This sets up the writer and specifies the delimiter (comma , in this case).

1. **Write rows of data**:  
   Use the writerow() method to write each row of data as a list:

python

Copy code

data\_handler.writerow(["Year", "Event", "Winner"])

data\_handler.writerow(["1995", "Best-Kept Lawn", "None"])

data\_handler.writerow(["1999", "Gobstones", "Welch National"])

Each call to writerow() adds a new row to the CSV file.

**Things to Remember**

* **File creation and overwriting**: If the specified file does not exist, Python will create it. If the file already exists, its contents will be overwritten by the new data.
* **Delimiter specification**: The delimiter (usually a comma) is required, but it can be set to other characters like tabs, semicolons, or pipes depending on the file's format.
* **newline=""**: This is important for ensuring the file format is handled correctly across different platforms (e.g., Windows vs. Unix-based systems).
* **Writing rows**: Use writerow() to add each row of data as a list to the CSV file. Each call to writerow() adds one line of data.