42 – CSV Files

Lesson Notes

from the

Teach Python Programming With Confidence Masterclass

By Nichola Wilkin Ltd

What are CSV files?

CSV stands for "Comma Separated Values" and are able to store 2D data in a table. The most common way that most teachers read a CSV file is through a spreadsheet such as Excel.

In an earlier section we had a look at the following 2D list in Python

```python
stars = [["Timothy",12,"Gemini"],
         ["Isabella",17,"Leo"],
         ["Zane",16,"Sagittarius"],
         ["Leah",18,"Gemini"],
         ["Keith",10,"Aquarius"]]
```

If we were going to save similar data in a text file it could look as follows:

```
Timothy,12,Gemini
Isabella,17,Leo
Zane,16,Sagittarius
Leah,18,Gemini
Keith,10,Aquarius
```

As you can see each record is on a separate row and the data is split up with commas, hence Comma Separated Values! This is how the data is stored in a CSV file but is it not how most programmers will view it. As spreadsheets can read CSV files and the columns make it easier to view the data in a neat table, programmers usually use a spreadsheet to view the data. Excel would display the data as follows:

	A	B	C
1	Timothy	12	Gemini
2	Isabella	17	Leo
3	Zane	16	Sagittarius
4	Leah	18	Gemini
5	Keith	10	Aquarius

It is exactly the same data, just shown in three different formats.

Writing a new CSV file

In order to use a CSV file you need to import the CSV module into Python. Then it is very similar to how you create a new text file.

```python
import csv

file = open("stars.csv","w")
name = "Timothy"
age = 12
starSign = "Gemini"
newRecord = name+","+str(age)+","+starSign+"\n"
file.write(newRecord)
file.close()
```

You will see it is opening the "stars.csv" file in write mode (shown with the "w") and then we are inputting the data. Name and starSign are both strings but age is a number as it doesn't have the speech marks around it.

Look at the newRecord line and you will see the age is being converted into a string so it can be joined with the commas. However, when I run this program and look at the data in a spreadsheet you will notice something rather odd.

	A	B	C
1	Timothy	12	Gemini
2			

Timothy and Gemini both appear on the left of the cells and the age is appearing on the right. It understands that the age is an integer and not a string and is treating it as a number.

Please note: When you are running the Python code you cannot have the CSV already open as it will not be able to run the code and you will get an error message similar to that shown below, so make sure you close the CSV file before you try to run the code.

```
Traceback (most recent call last):
  File "C:/Users/admin/Desktop/My Programs/test.py", line 3, in <module>
    file = open("stars.csv","w")
PermissionError: [Errno 13] Permission denied: 'stars.csv'
```

Have a look at the following example:

```python
import csv

file = open("stars.csv","w")
name = input("Enter name: ")
age = input("Enter age: ")
starSign = input("Enter star sign: ")
newRecord = name+","+age+","+starSign+"\n"
file.write(newRecord)
file.close()
```

We are not indicating that age is an integer anywhere and we treat it in exactly the same way as the name and star sign but when I run it, if I enter a number for the age it will automatically select the integer as the most suitable data type.

	A	B	C	
1	Bob	17	Leo	

We have been using the write mode but there is another option you can use when creating a new CSV file.

Instead of the "w" mode you can use "x" mode. This will only create a new blank file if the file does not already exist. If there is already a file created with the same name in that location it will create this error message and stop the program from running.

> The "x" will only create a new program if there is not one already created otherwise it will crash the program.

```python
import csv

file = open("stars.csv","x")
name = input("Enter name: ")
age = input("Enter age: ")
starSign = input("Enter star sign: ")
newRecord = name+","+age+","+starSign+"\n"
file.write(newRecord)
file.close()
```

This is the error message you will get if the file already exists:

```
Traceback (most recent call last):
  File "C:/Users/admin/Desktop/My Programs/test.py", line 3, in <module>
    file = open("stars.csv","x")
FileExistsError: [Errno 17] File exists: 'stars.csv'
```

Adding data to an existing CSV file

If you want to add data to the end of an existing CSV file use the append mode ("a").

```
import csv

file = open("stars.csv","a")
for count in range(0,3):
    name = input("Enter name: ")
    age = input("Enter age: ")
    starSign = input("Enter star sign: ")
    newRecord = name+","+age+","+starSign+"\n"
    file.write(newRecord)
file.close()
```

Reading a CSV file

You can also read the file using the read mode ("r")

```
import csv

file = open("stars.csv","r")
print(file.read())
file.close()
```

This will display the data with commas separating the values.

```
Bob,17,Leo
Sandra,48,Virgo
Keith,10,Leo
Zoe,25,Scorpio
```

It is possible to manipulate the data further using 2D lists and CSV files but I feel if you want to interrogate the data further and find specifics it is easier to do that using SQLight which is what we are going to be doing in our next lesson.