Computational Thinking and Programming - 2

CSV Files

Syllabus

CSV files:

- import csv module
- open / close csv file
- write into a csv file using csv.writerow() and
- read from a csv file using csv.reader( )

Introduction - CSV Files

CSV (Comma Separated Values) file format is the most common format for tabulated data to be used in any spreadsheet tool (Calc, Google Sheets, Excel) and any database. A CSV file stores tabular data (numbers and text) in plain text. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The name CSV comes from the use of commas as a field separator for this file format.

Introduction - CSV Files

Reading / writing data in text files is a common and simple way to share information between the programs. However, CSV is the most popular format for exchanging data nowadays as it can be directly used in text editors, spreadsheets and databases.

A CSV file (Comma Separated Values) is a type of plain text file that uses specific structuring to arrange tabular data. Because it is a plain text file, it can contain only actual text data - in other words, printable ASCII or Unicode characters.

Sample CSV File Format - without header

The following are some of the examples of CSV files without header:

CSV File content with 3 Fields/Columns

```
1,Simran,80
2,Rishabh,99
```

CSV File content with 4 Fields/Columns with comma as separator

```
12,"Simran",3400,"34,ABC Colony, DL"
14,"Rishabh",4300,"R-90,XYZ Nagar, DL"
```

CSV File Format - with header

The following is an example of CSV file with header:
CSV File content with 3 Fields/Columns

Rno,Name,Percent
1,Harish Gupta,100
2,Nidhi Goel,90
3,Sangeeta Sareen,70

CSV File using Spreadsheet Tool

	A	B	C
1	Rollno	Name	Marks
2	1	Harish Gupta	100
3	2	Nidhi Goel	90
4	3	Sangeeta Sareen	70

CSV File using Text Editor

```
Rollno,Name,Marks
1,Harish Gupta,100
2,Nidhi Goel,90
3,Sangeeta Sareen,70
```

CSV Files

CSV Files use comma as the separator between the specific data values, but if comma itself is a part of the content, it can be enclosed inside quotation marks.

PEOPLE.CSV

```
Name,      Age,     Profession
Jasmeet,   23,      Doctor
Manju,     22,      Engineer
Rahul,     67,      Professor
Simi,      43,      Singer
```

STUDENT.CSV

```
RNo,       Name,    Address
A001       Kartik   "H1, Hauz Khas"
A002       Simi      "C1 South Ex"
A003       Arsh      "70, Block A, Jang Pura"
A004       Pooja     "CBlock, Defence Colony"
```

File modes:

FILE MODES	OPERATION
r	**Read mode :** Read contents from an existing CSV file, raises an exception, FileNotFoundError, if the file does not exist.
r+	**Update mode:** Opens the file both for **reading** and **writing** the CSV file. The file pointer/cursor is placed at the beginning of the CSV file.
w	**Write mode :** Opens a CSV file in write-only mode. In case it is existing, the contents are overwritten.
a	**Append mode :** It opens the file to write the content at the end of an existing CSV file. The file cursor is at the end of the file in case the file exists. In case the file is not existing, a new file gets created.

Writing contents to a CSV File

Writing data to a CSV file involves the following steps:

1. import csv module
2. Opening the csv file in write mode using open() or with open….as
3. Creating the writer object
4. Writing the data onto the file using the writer object
5. Closing the file using close(), if open function is used.

Creating writer object using **csv.writer()** method - returns a writer object that is used to write data in the csv file.

Opening and Writing contents to a CSV File

Opening of a CSV file is the same as that for a text file.

For writing into a CSV file, the following method is used:

csv.writer() - Method required for writing the contents in a CSV file.It will work only when the CSV file is opened as a text file in "w" or "a" mode.

```
import csv
with open(<FileName>,<Mode>,newline="") as <CSVFileObject>:
    <WriterObj> = csv.writer(<CSVFileObject>,delimiter = ',')
    <WriterObj>.writerow(<List Content>)
```

Filename is name of the file with csv extension.

Creating writer Object

Creating writer Object

The csv.writer() function is used to return a writer object that is used to write data into csv file.

```
<WriterObj> = csv.writer(<CSVFileObject>,delimiter = ',')
```

<CSVFileObject> is the object used to open the csv file.

The delimiter character string is optional, default value is comma (,)

Writing To a CSV File

Once the writer object returned by csv.writer() converts the user's data into a delimited string, the row can be written to the csv file using

- writerow() and
- writerows() method.

writerow() -

- The writerow() method writes one row of data (given as list, tuple or string) onto the csv file using the writer object.
- The row can include list/tuple having multiple fields where the fields can be strings or numeric or both.
- while using writerow(), it is not required to add a new line character\n or other EOL indicator to indicate the end of the line; writerow() does it automatically as necessary.

writerows() method

writerows() - This method is used to write multiple rows at a time.

The writerows() method takes a sequence (nested list/nested tuple) as parameter and writes each item as a comma separated line of items in the file.

Like writerow(), it is not required to add a new line character\n or other EOL indicator to indicate the end of the line; It does it automatically as necessary.

```
import csv
def CreateItem():
    Recs = [['ItemNo','Name','Price'], [1,'Pen',10], [2,'Pencil',9],
[3,'Stappler',70]]
    with open("F.csv",'w',newline="") as f:
            cv=csv.writer(f,delimiter = ',')
            cv.writerows(Recs)
```

	A	B	C
1	ItemNo	Name	Price
2	1	Pen	10
3	2	Pencil	9
4	3	Stappler	70
5			

writer() and writerow() method

```
import csv
def Create():
    with open("file.csv",'w',newline="") as f:
        cv = csv.writer(f,delimiter =',')
        cv.writerow(['Rollno','Name','Marks'])
        cv.writerow([1,'Harish Gupta',100])
        cv.writerow([2,'Nidhi Goel',90])
        cv.writerow([3,'Sangeeta Sareen',70])
```

In case the newline is not specified, then it will insert an empty row after every record. By default the newline character is \n\r which will result in an empty row after every record.

	A	B	C
1	Rollno	Name	Marks
2	1	Harish Gupta	100
3	2	Nidhi Goel	90
4	3	Sangeeta Sareen	70

```
Rollno,Name,Marks
1,Harish Gupta,100
2,Nidhi Goel,90
3,Sangeeta Sareen,70
```

writer() method of the CSV module

```
import csv
def CreateItem():
    Recs = [['ItemNo','Name','Price'], [1,'Pen',10], [2,'Pencil',9], [3,'Stappler',70]]
    with open("F.csv",'w',newline="") as f:
            cv=csv.writer(f,delimiter = ',')
            for r in Recs:
                cv.writerow(r)
```

Reading from a CSV File

For reading the contents from a CSV file, the following method is used:

csv.reader() - Method required for reading the contents from a CSV file.It will work only when the CSV file is opened as a text file in "r" mode.

```
import csv

with open(<FileName>,<Mode>, newline = "") as <CSVFileObject>:

    <ReaderObj> = csv.reader(<CSVFileObject>,delimiter = ',')
```

Reading from a CSV File

To read the data present in a CSV file -:

i. Import csv module

ii. Open the CSV file as a text file in reading mode (“r”) with Python’s built-in open() function, which returns a file object.

Syntax - Fileobject = open(<FileName>,<Mode>, newline = "") or with open….as

Example - f = open("file.csv","r")

iii. Creating reader object and manipulating the data using reader object

iv. Closing the file using close() Example - f.close()

reader() method

The reader() method is used to create a reader object to read the csv file.

<ReaderObj> = csv.reader(<CSVFileObject>, delimiter= string)

```
import csv
f  = open("file.csv","r")          #file opened in read mode
cv = csv.reader(f)                 #reader object created
for rec in cv:
    print(rec)
csv_fobj.close()
```

The reader() takes the file handle as argument and returns the reader object rdrobj.

The reader object loads data from the csv file, removes the delimiters and returns the data in the form of an iterable sequence like list, tuple etc.

Using a for loop the data is fetched from the reader object record by record and displayed.

Reading CSV File using next()

As csvreader is an iterable object, the next() function is used to return the current row and advances the iterator to the next row.

The next() takes the file handle as argument and moves the record pointer to the next record after reading the current one.

Write a function countrec() to count the number of records present in the csv file.

```
def CountRec():
    f = open("items.csv","r",newline="")  #file opened in read mode
    cv = csv.reader(f)                    #reader object created
    c = next(f)                           #read the heading row
    ctr=0
    for c in cv:
        ctr+=1
    print(ctr)
    f.close()
```

Reading from a CSV File

```
def Read():
    with open("file.csv",'r',newline="") as f:
        cv=csv.reader(f,delimiter =',')
        for c in cv:
            print(c)
```

OUTPUT:

```
['Rollno', 'Name', 'Marks']
['1', 'Harish Gupta', '100']
['2', 'Nidhi Goel', '90']
['3', 'Sangeeta Sareen', '70']
```

Reading from a CSV File

```
def Read():
    with open("file.csv",'r',newline="") as f:
        cv=csv.reader(f,delimiter =',')
        print(next(cv))
        print(next(cv))
        print(next(cv))
        print(next(cv))
```

OUTPUT:

```
['Rollno', 'Name', 'Marks']
['1', 'Harish Gupta', '100']
['2', 'Nidhi Goel', '90']
['3', 'Sangeeta Sareen', '70']
```

Reading from a CSV File

```
def Read():
    try:
        with open("file.csv",'r', newline="") as f:
            cv=csv.reader(f,delimiter =',')
            for c in cv:
                print(c)
    except FileNotFoundError:
        print("File named file.csv is not found")
```

Creating and Displaying CSV File - input from User

```
import csv
def CreateItem():
    with open("Item.csv",'w',newline="") as f:
        cv = csv.writer(f,delimiter =',')
        cv.writerow(['ItemCode','ItemName','Price'])
        while True:
            ICode = input("Enter the Item Code :")
            IName = input("Enter the Item Name : ")
            Price = int(input("Enter the Price : "))
            cv.writerow([ICode,IName,Price])
            choice = input("Enter more? (y/Y/n/N) : ")
            if choice in "nN":
                break
def DisplayItem():
    try:
        with open("Item.csv") as f:
            cv = csv.reader(f,delimiter = ',')
            for c in cv:
                print(c)
    except:
        print("File not found")
```

Adding Data in an existing CSV File

```
def AddItem():
    with open("Item.csv",'a',newline="") as f:
        cv = csv.writer(f,delimiter =',')
        while True:
            ICode = input("Enter the Item Code :")
            IName = input("Enter the Item Name : ")
            Price = int(input("Enter the Price : "))
            cv.writerow([ICode,IName,Price])
            choice = input("Enter more? (y/Y/n/N) : ")
            if choice in "nN":
                break
```

Search in a CSV File

```
def SearchItem():
    Code = input("Enter the Item Code :")
    try:
        with open("Item.csv",newline="") as f:
            cv = csv.reader(f,delimiter = ',')
            for c in cv:
                if c[0] == Code:
                    print(c)
                    break
                else:
                    print(Code, " not found in the file")
    except:
        print("File not found")
```

Creation using writerows() method

```
import csv
def CreateItem():
    with open("Item.csv",'w', newline="") as f:
        cv = csv.writer(f,delimiter =',')
        L = []
        L.append(['ItemCode','ItemName','Price'])
        while True:
            ICode = input("Enter the Item Code :")
            IName = input("Enter the Item Name : ")
            Price = int(input("Enter the Price : "))
            L.append([ICode,IName,Price])
            choice = input("Enter more? (y/Y/n/N) : ")
            if choice in "nN":
                break
        cv.writerows(L)
```

In CSV file, both the methods can be used for creating a file
writerow()
writerows()
No difference in reading

will write all rows in one go

Display contents of the CSV file

```
def DisplayItem():
    try:
        with open("Item.csv") as f:
            cv = csv.reader(f,delimiter = ',')
            for c in cv:
                print(c)
    except:
        print("File not found")
```

```
Enter the Item Code :A001
Enter the Item Name : School Bag
Enter the Price : 500
Enter more? (y/Y/n/N) : Y
Enter the Item Code :A002
Enter the Item Name : Water Bottle
Enter the Price : 300
Enter more? (y/Y/n/N) : n
```

Update in the file

```
def UpdateItem():
    try:
        with open("Items.csv",'r+') as f:
            Code = input("Enter the Item Code: ")
            cv = csv.reader(f,delimiter = ',')
            L=[]; changed = False
            for c in cv:
                if c[0]==Code:

                    print("Item Name:",c[1],"Price:",c[2])
                    c[2]=  int(input("Enter new Price:"))
                    changed = True
                L.append(c)
            if changed == False:
                print(Code, " not present in the file")
```

```
 else:
    f.truncate(0)
    f.seek(0)
    cw = csv.writer(f)
    cw.writerows(L)
except :
   print("File not found ")
```

Count the number of records in the file

```
def CountRec():
    f = open("items.csv","r",newline="")#file opened in read mode
    cv = csv.reader(f)                  #reader object created
    c = next(f)                         #read the heading row
    ctr=0
    for c in cv:
        ctr+=1
    print(ctr)
    f.close()
```

Navigating through the options:

```
while True:
    print("1. Create a file (W/w) ")
    print("2. Display the file (D/d) ")
    print("3. Append into a file (A/a) ")
    print("4. Search the file (S/s) ")
    print("5. Update in a file (U/u) ")
    print("6. Count the number of rows (C/c) ")
    print("7. Quit the program (Q/q) ")
    choice = input("Enter your choice : ")
```

```
    if choice in "Ww":
        CreateItem()
    elif choice in "Dd":
        DisplayItem()
    elif choice in "Aa":
        AppendItem()
    elif choice in "Ss":
        SearchItem()
    elif choice in "Uu":
        UpdateItem()
    elif choice in "Cc":
        CountRec()
    elif choice in "Qq":
        break
```

Recapitulation

A CSV file (Comma Separated Value) is a type of text file (ASCII or UNICODE characters) that uses specific structuring to arrange tabular data.

In a CSV file

- each line, is referred to as a **Record** has the same structure
- consists of a number of values called **Fields**
- separated by some specified character (or sequence of characters), typically a comma (hence the name "**comma separated values**").
- The character (or characters) that separates the fields is called a **Delimiter**

Advantages

- CSV is human readable and easy to edit manually.
- CSV files are smaller in size, faster to handle and easy to generate.
- CSV file can store large amount of data.
- CSV is simple to implement and parse.
- CSV file can be processed by almost all existing applications.
- CSV files provide convenient way for exchanging data as it can be directly used in text editor, spreadsheets, and databases

THANK YOU!

HAPPY LEARNING!!!

WRITE A PROGRAM TO read from FILE “ABX.CSV” AND DISPLAY RECORD OF SCHOOLS WHICH HAVE NO_STUDENTS MORE THAN 4000

Ans :

```
import csv
f=open(“abx.csv”,”r”,newline=””)
cv=csv.reader(f)
for i in cv:
        if  int(i[2])>4000:
                print(i)
f.close()
```