CSV FILE HANDLING

CSV files are delimited files that store tabular data (data stored in rows and columns as we see in spreadsheets or databases) where comma delimits values every value, i.e., they are separated with comma

Advantages of csv file:

- Easier to create,
- Easier to manipulate large amount of data
- Preferred export and import format for databases
- Preferred export and import format for spread sheets,
- Capable of storing large amounts of data.

CSV files are the text files, you can apply text file procedures on these and then split values using split() function but better procedure is using csv module of Python.

Python csv Module

import csv

The csv module of Python provides functionality to read and write tabular data in CSV format. It provides two specific types of objects

1. **reader objects**: objects read delimited sequence records from a CSV file.
2. **writer objects**: objects write delimited sequence records in a CSV file.

Opening/Closing CSV Files

A CSV file is opened as open any other text file but make sure to do the following two things

1. Specify the file extension as .csv
2. Open the file like other text files
   Dfile = open ("stu.csv", "w") # CSV file opened in write mode with file handle as Dfile
   File1 = open("stu.csv", "r") # CSV file opened in read mode with file handle as File1

   Dfile.close()

   Note:

   1. The file modes, "w", "w+", "a", "a +": The file will get created if it does not exist already.
   2. The modes "w" and "w +": will overwrite the existing file. (if file is exist)
   3. The file mode "a" or "a+" will retain the contents of existing the file. (if file is exist)

Role of Argument newline in Opening of csv Files

- The role of newline argument is to handle newline characters while working with csv files
- As csv files are text files, while storing them, some types of translations occur
- translation of end of line (EOL) character
- Additional optional argument as newline ="((null string; no space in between)

- With file open( ) will ensure that no translation of end of line (EOL) character takes place.

That is, open your csv file as :

```
Dfile = open("stu.csv", "w", newline = '')
```

null string ; no space in between

*CSV file opened in **write mode** with file handle as **Dfile** (no EOL translation)*

Or

```
File1 = open("stu.csv", "r", newline = '')
```

*CSV file opened in **read mode** with file handle as **File1** (no EOL translation)*

Writing in CSV Files:

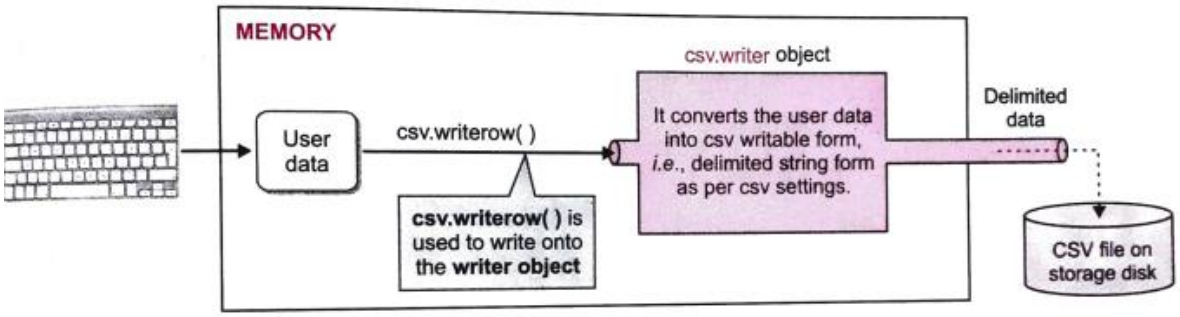
Writing into csv files involves the conversion of the user data into the writable delimited from and then storing it in the form of csv file.

Functions used with csv file for writing

1. **csv.writer ():** returns a writer object which writes data into CSV file
2. **csvwriterobject>.writerow() :** writes one row of data onto the writer object
3. **csvwriterobject>.writerows() :** writes multiple rows of data onto the writer object

Note: Before writing onto csv file, the data must be in csv-writable-delimited-form. This task is performed by the writer object.

The data row written to a writer object (written using writerow() or writerows() functions) gets converted to csv writable delimited form and then written on to the linked csv file on the disk.



Role of the csv writer object.

The writerow() Function:

Writes one rows of data onto the writer object

1. import csv module
2. Open csv file in a file-handle (just as you open other text files)
   fh =open("student.csv", "w", newline=' ')
3. Create the writer object by using the syntax as shown below:
   <name-of-writer-object= csv.writer(file handler, delimiter = 'delimiter character')
   Note: If delimiter argument skipped, comma will be used as the delimiter
   Stuwriter= csv.Writer(fh) # default delimiter is comma (" ,")
   stuwriter = csv.writer (fh, delimiter = "|") # delimiter character as pipe symbol ("|")
4. Obtain user data and form a Python sequence (list or tuple) out of it

   Sturec = (11, "Neelam', 79.8) # Sturec is Python sequence of data

5. Write the Python sequence containing user data onto the writer object using csv wniterow() functions
   stuwriter .writerow(['Rollno", 'Name', Marks'])
   stuwriter.writerow(Sturec)
6. Close the file

Write a program to create a CSV file to store student data (Rollno., Name, Marks). Obtain data from user and write 5 records into the file.

```
import csv
fh =open( "Student.csv", "w", newline=' ')          #open file, newline=" no blank space
stuwriter =csv.writer ( fh)                          # create writer object
stuwriter .writerow ([Rollno', 'Name", 'Marks' ])    # write header row in csv file.
for i in range (5) :
    print ("Student record", (i+1))
    rollno= int (input ("Enter rollno: "))
    name= input( "Enter name: ")
    marks= float (input ( "Enter marks: "))
    sturec= [rollno, name , marks]      # create Python sequence of user data
    stuwriter.writerow( sturec)         # Python sequence written on the file
fh.close()                              #close file
```

The writerows() Function:

- writes multiple rows of data onto the writer object
- The writerows( ) method writes all given rows to the CSV file
  Sturec = [[11, Nistha, 79.0], [12, Rudy, 89.0], [13,Rustom, 75.0]]
  writerobject>.writerows (Sturec)

Program: write a program to create a csv file (compresult.csv) and write the above data into it.

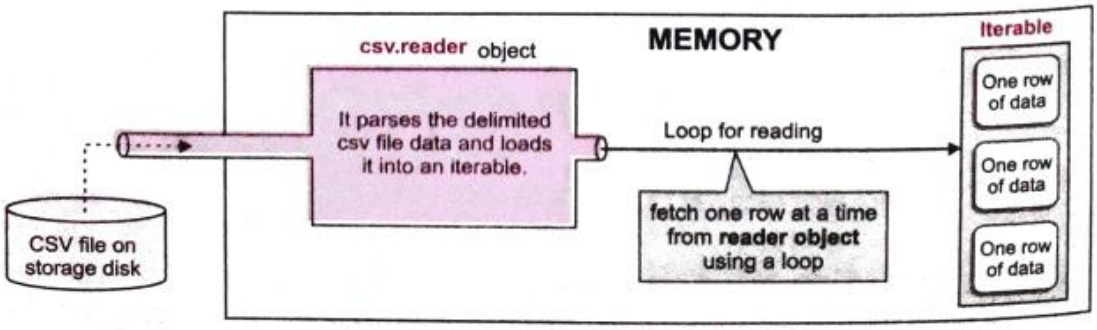
```
import csv
fh= open ("compresult.csv", "w", newline=' ')
Cwriter = csv.writer (fh)
Compdata= [
    [Name', 'Points', 'Rank'],
    ['Shradha', 4500, 23],
    ['Nishchay', 4800, 31],
    ['Ali', 4500, 25],
    [Adi', 5100, 14] ]
Cwriter. writerows (compdata)
fh.close()
```

Reading in CSV Files

- Reading from a csv file involves loading of a csv file's data in a Python iterable (Python sequence- List , tuple etc)
- Parsing data (Mean to remove its delimitation)
- Then reading from this iterable (Python sequence- List , tuple etc)

csv.reader():

- The csv.reader object loads data from the csv file,
- parses it, i.e., removes the delimiters
- Returns a reader object which loads data from CSV file into an Python iterable
- Python can fetch one row of data at a time from iterable.
- The csv.reader object does the opposite of csv.writer object.
- The csv.reader object loads dad from the csv file,



Role of csv.reader object.

1. import csv module
2. Open csv file in a file-handle in read mode (File must already exist otherwise exception will be raised)
3. fh= open("student.csv", "r")
4. Create the reader object
   stureader = csv.reader(fh) # Default delimiter character is comma
   stureader = cSV.reader(fh, delimiter = '| ') # Delimiter character is '| '
5. The reader object stores the parsed data in the form of iterable
6. Python can fetch one row of data at a time from iterable by using any loop.
   for rec in stureader:
   print (rec)
7. Once done, close the file.

Program: Write a program to read and display the contents of Employee.csv created in the previous program

```
import csv

with open("Employee.csv", "r") as fh :

    ereader = csv.reader (fh)

    print("File Employee.csv contains:")

    for rec in ereader:

        print(rec)
```