CSV Files
(comma separated values)

CSV Files
(comma separated values)

- CSV is a comma separated values.
- It is a file format used to store tabular data.
- The tabular data stores in csv file as plain text.

Features of CSV

- file extension **.csv**.
- They are plain text files having ASCII/Unicode Characters.
- Used to handle large amount of data.
- Language which support text files will also support csv files.
- Data is stored in the form of rows and column i.e in tabular form.
- Thus easy to export from databases or spreadsheets.
- csv files can be easily imported to other programs.

Why CSV files

- In practical application we may have to work only data rows only without any heading.
- Each records of csv file consist of field values separated by commas.
- CSV are generally but not necessarily created using excel.
- CSV file can be open in excel, word, notepad or any text editor or any programming language which supports file processing .
- The data from this file can also be imported in a table in a database.

CSV Advantages

- • CSV is faster to handle
- • CSV is smaller in size
- • CSV is easy to generate
- • CSV is human readable and easy to edit manually
- • CSV is simple to implement and parse
- • CSV is processed by almost all existing applications

CSV Disadvantages

- • No standard way to represent binary data
- • There is no distinction between text and numeric values
- • Poor support of special characters and control characters
- • CSV allows to move most basic data only. Complex configurations cannot be imported and exported this way
- • Problems with importing CSV into SQL (no distinction between NULL and quotes)

How CSV files different from TEXT files

- Text files contain text which can be opened by any text editor and there is **plain text with no format**

- CSV also contain text data but in a **format** where each line is considered as **row/record which has many fields(columns)**.
  - fields are the values separated by a delimiter like , " ' ,"*","/" ,"\n"etc.
  - the first record is the title of each field.

To Read a CSV file

- **csv module** is used to read/write in a csv file.
- **csv.reader()** -is used to read the records from the csv file.

  the records read are in the form of list.

(in other words . It loads the data from CSV file into an iterable after parsing delimited data)

- records can also be read in to a dictionary using **csv.DictReader()** function
- **Pandas library** can also be used to read data. They provide high quality data analysis.

Syntax to read data from CSV file

- **Import csv module**
- **Open a csv file in read mode**
- **Create the reader object**
- **Fetch data through loop i.e. row by row**
- **Close the file**

Example -To read a csv file

```
import csv
f=open("book1.csv","r")  # open an existing csv file
r=csv.reader(f)
for i in r:
    print(i)
f.close()
```

Import csv module

Open a CSV file in read mode

Create the reader object

Fetch data through loop i.e. row by row

Close the file

Example -To read a csv file

- import csv
- f=open("book1.csv","r", newline='\r\n') # open an existing csv file
- r=csv.reader(f)
- for i in r:
- print(i)
- f.close()

Import csv module

Open a CSV file in read mode

Create the reader object

Fetch data through loop i.e. row by row

Close the file

newline parameter is used to remove the blank line in a file

End of line(EOL) characters used in different operating systems

CR[\r]	**carriage return**	**macintosh**
LF[\n]	**line feed**	**unix**
CR/LF[\r\n]	**carriage return/line feed**	**ms-dos,windows**
NULL[\0]	**null character**	**all operating system**

newline=“ “ (null string no space in between) with file open will ensure that no translation of EOL character takes place

Example -To display a csv file in tabular format

```
import csv
f=open(r"C:\Users\user\Desktop\Book1.csv","r")
ro=csv.reader(f)
for i in ro:
    print(i[0],"\t",i[1],"\t",i[2],"\t",i[3])
```

Example -To display a csv file in tabular format or create table from csv

- **#pip install prettytable**
- from prettytable import from_csv
- f=open(r"C:\Users\user\Desktop\Book1.csv","r")
- j=from_csv(f)
- print(j)

Example -To display a list in tabular format or create table from list

- **#pip install prettytable**
- from prettytable import PrettyTable
- l=['a','b','c']
- l1=['1','2','3']
- t=PrettyTable(['list 1','list 2'])
- for x in range(0,3):
- t.add_row([l[x],l1[x]])
- print(t)

Example -To display a csv file in tabular format

```
f1=open("marks.csv","r")
ro=csv.reader(f1)
t=[]
for i in ro:
    t.append(i)
t=t[1:]
fc="%3s %14s %18s %12s"
print("="*60)
print(fc%('s.no', 'name', 'marks', 'age'))
print("="*60)
rf="%3s %14s %18s %12s"
for j in t:
    print(rf%(j[0],j[1],j[2],j[3]))
    print("-"*60)
f1.close()
```

To write in to CSV files

- **csv module** is used to read/write in a csv file.
- **csv.writer()** -to convert file object into writer object for writing the records
- **csv.writerow()** -is used to insert the records in the csv files
- records can be written as a **nested list or dictionary**.
- **DictWriter()** is used to write dictionary from csv file
- **pandas library** can also be used to write data into csv files.

Example -To write a csv file

- import csv #Creation of csv file
- h=['s.no', 'name', 'marks', 'age']
- re=[[1, 'abc', 98, 14],[2, 'def', 87, 16],[3, 'ghi', 95, 15],[4, 'hhh', 92, 16]]
- f=open("marks.csv","w")
- r=csv.writer(f,lineterminator='\n')
- r.writerow(h)
- r.writerows(re)
- f.close()

Example -To write n no of records into csv file

```
import csv
def writecsv():
    f = open("data.csv","w")
    rec=csv.writer(f,lineterminator="\n")
    rec.writerow(['roll','name','age'])
    while True:
        r=int(input("enter rollno"))
        n=input("enter name")
        a=int(input("enter age"))
        v=[r,n.upper(),a]
        rec.writerow(v)
        ch=input("more records")
        if(ch=='n'):
            break
```

Example -To write n no of records into csv file

```
import csv
def writecsv():
        f = open("data.csv","w“, newline=“ “)
        rec=csv.writer(f)
        rec.writerow(['roll','name','age'])
        while True:
            r=int(input("enter rollno"))
            n=input("enter name")
            a=int(input("enter age"))
            v=[r,n.upper(),a]
            rec.writerow(v)
            ch=input("more records")
            if(ch=='n'):
                break
```

newline=“ “ (null string no space in between) with file open will ensure that no translation of EOL character takes place

To write, read and search a record

- import csv

To Write:

```
def writecsv():
        f = ("data.csv","w")
        rec=csv.writer(f,lineterminator="\n")
        rec.writerow(['roll','name','age'])
        while True:
            r=int(input("enter rollno"))
            n=input("enter name")
            a=int(input("enter age"))
            v=[r,n.upper(),a]
            rec.writerow(v)
            ch=input("more records")
            if(ch=='n'):
                break
```

To Read

```
def readcsv():
    with open("data.csv","r") as f:
            rec=csv.reader(f)
            for i in rec:
                    print(i)
```

To Search

```
def searchcsv():
    with open("data.csv","r") as f:
        rec=csv.reader(f)
        next(rec)
        for i in rec:
            if (i[2]>='45'):
                print(i)
```

Calling of all functions

```
writecsv()
readcsv()
searchcsv()
```

next()- #this will read the first record and then pointer will move to the beginning of the second record

Questions on CSV

A csv file **counties.csv** contains data in the following order:
country,capital,code

sample of counties.csv is given below:

india,newdelhi,ii
us,washington,uu
malaysia,ualaumpur,mm
france,paris,ff

write a **python function** to read the file **counties.csv** and display the names of all those countries **whose no of characters in the capital are more than 6.**

- import csv

```
def writecsv():
    f=open("counties.csv","w")
    r=csv.writer(f,lineterminator='\n')
    r.writerow(['country','capital','code'])
    r.writerow(['india','newdelhi','ii'])
    r.writerow(['us','washington','uu'])
    r.writerow(['malysia','kualaumpur','mm'])
    r.writerow(['france','paris','ff'])
```

```
def searchcsv():
    f=open("counties.csv","r")
    r=csv.reader(f)
    f=0
    for i in r:
        if (len(i[1])>6):
            print(i[0])
            f+=1

    if(f==0):
        print("record not found")
```

Calling of functions

writecsv()

searchcsv()

write a python function to search and display the record of that product from the file **PRODUCT.CSV** which has maximum cost.

sample of product .csv is given below:

```
pid,pname,cost,quantity
p1, brush,    50,    200
p2, soap,     120,   150
p3, comb,     40,    300
p4, sheets,   100,   500
p5, pen,      10,    250
```

- import csv

```
def writecsv():
    f=open("product.csv","w")
    r=csv.writer(f,lineterminator='\n')
    r.writerow(['pid','pname','cost','qty'])
    r.writerow(['p1','brush','50','200'])
    r.writerow(['p2','toothbrush','120','150'])
    r.writerow(['p3','comb','40','300'])
    r.writerow(['p5','pen','10','250'])
```

Calling of functions

writecsv()

searchcsv()

```
def searchcsv():
        f=open("product.csv","r")
        r=csv.reader(f)
        next(r)
        m=-1
        for i in r:
                if (int(i[2])>m):
                m=int(i[2])
                d=i
        print(d)
```

next()- #this will read the first record and then pointer will move to the beginning of the second record

write a python function to search and display the total cost of all products from the file PRODUCT.CSV.

sample of product .csv is given below:

```
pid,pname,cost,quantity
p1, brush,    50,    200
p2, soap,     120,   150
p3, comb,     40,    300
p4, sheets,   100,   500
p5, pen,      10,    250
```

- import csv

```
def writecsv():
    f=open("product.csv","w")
    r=csv.writer(f,lineterminator='\n')
    r.writerow(['pid','pname','cost','qty'])
    r.writerow(['p1','brush','50','200'])
    r.writerow(['p2','toothbrush','120','150'])
    r.writerow(['p3','comb','40','300'])
    r.writerow(['p5','pen','10','250'])
```

Calling of functions

writecsv()

searchcsv()

```
def searchcsv():
        f=open("product.csv","r")
        r=csv.reader(f)
        next(r)
        s=0
        for i in r:
                s=s+int(i[2])
        print("total cost is",s)
```

next()- #this will read the first record and then pointer will move to the beginning of the second record

write a python function to find transfer
only those records from the file product.csv to another file "pro1.csv" whose quantity is more than 150. also include the first row with headings.

sample of product .csv is given below:

```
pid,pname,   cost,quantity
p1, brush,   50,    200
p2, soap,    120,   150
p3, comb,    40,    300
p4, sheets,  100,   500
p5, pen,     10,    250
```

- import csv

```
def writecsv():
    f=open("product.csv","w")
    r=csv.writer(f,lineterminator='\n')
    r.writerow(['pid','pname','cost','qty'])
    r.writerow(['p1','brush','50','200'])
    r.writerow(['p2','toothbrush','120','150'])
    r.writerow(['p3','comb','40','300'])
    r.writerow(['p5','pen','10','250'])
```

```
def readcsv():
    f=open("pro1.csv","r")
    r=csv.reader(f)
    for i in r:
        print(i)
```

```
def searchcsv():
        f=open("product.csv","r")
        f1=open("pro1.csv","w")
        r=csv.reader(f)
        w=csv.writer(f1,lineterminator='\n')
        g=next(r)
        w.writerow(g)
        for i in r:
                if i[3]>'150':
                w.writerow(i)
```

Calling of functions

writecsv()
searchcsv()
readcsv()

next()- #this will read the first record and then pointer will move to the beginning of the second record