CSV files (csv module)

CSV stands for Comma Separated Values it is data interchange format.
CSV file is text file having extension .csv

The so-called CSV (Comma Separated Values) format is the most common import and export format for spreadsheets and databases.

“csv” module provides the classes or data types to work with csv files. It is a default module which comes with python software.

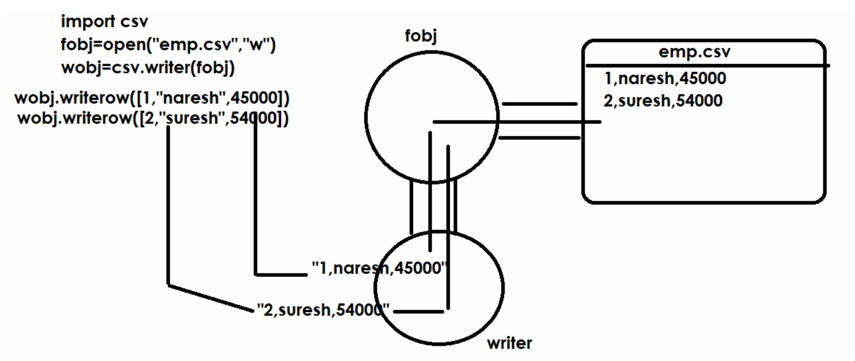
Default separator used to write data is comma (comma separated values string).

csv module provides the following classes

1. writer
2. reader
3. DictWriter
4. DictReader

writer class

csv.writer(csvfile)
Return a writer object responsible for converting the user’s data into delimited strings on the given file-like object. csvfile can be any object with a write() method. If csvfile is a file object, it should be opened with newline=''



```
import csv
fobj=open("emp.csv","w")
wobj=csv.writer(fobj)
wobj.writerow([1,"naresh",45000])
wobj.writerow([2,"suresh",54000])
```

Writer object provides a method writerow, this method converts a list into comma separated values string.

Example:

```
# Create emp.csv file for storing employee details
import csv
fobj=open("emp.csv","a",newline='')
wobj=csv.writer(fobj)
while True:
    empno=int(input("EmployeeNo :"))
    ename=input("EmployeeName :")
    sal=float(input("Salary :"))
    wobj.writerow([empno,ename,sal])
    ans=input("Add another employee?")
    if ans=="no":
        break

fobj.close()
```

Output

```
EmployeeNo :1
EmployeeName :naresh
Salary :45000
Add another employee?yes
EmployeeNo :2
EmployeeName :suresh
Salary :65000
Add another employee?yes
EmployeeNo :3
EmployeeName :kishore
Salary :35000
Add another employee?yes
EmployeeNo :4
EmployeeName :rajesh
Salary :90000
Add another employee?no
```

reader object

csv.reader(csvfile)
Return a reader object that will process lines from the given csvfile. A csvfile must be an iterable of strings, each in the reader's defined csv format.

Example:

```
# reading data from emp.csv file

import csv

fobj=open("emp.csv","r")
robj=csv.reader(fobj)
for emp in robj:
    print(emp)

fobj.close()
```

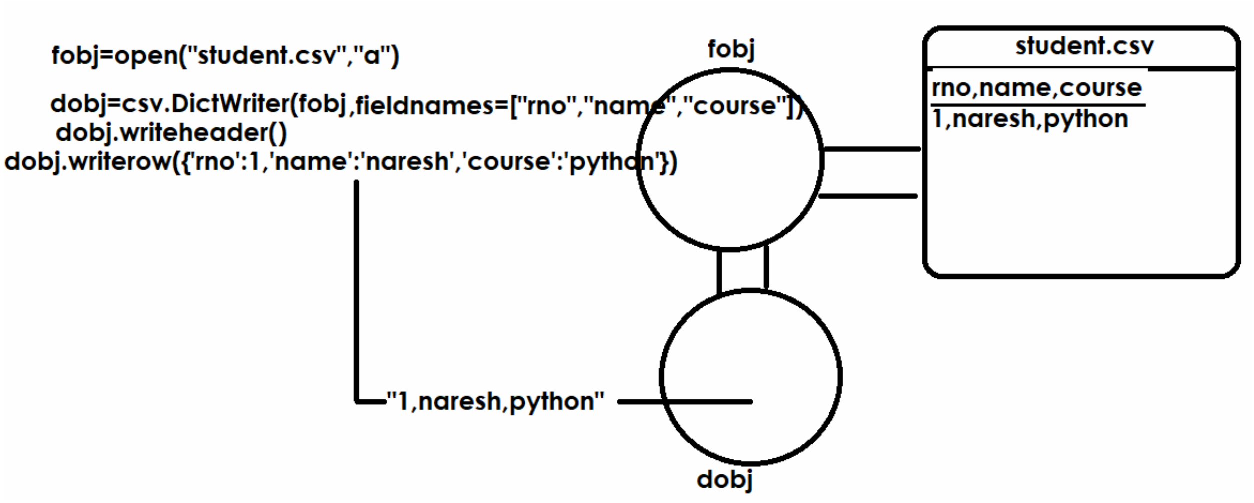
Output

```
['1', 'naresh', '45000.0']
['2', 'suresh', '65000.0']
['3', 'kishore', '35000.0']
['4', 'rajesh', '90000.0']
```

Writer and reader objects write and read data in list.

DictWriter object

csv.**DictWriter**(*f*, *fieldnames)*
Create an object which operates like a regular writer but maps dictionaries onto output rows. The *fieldnames* parameter is a sequence of keys that identify the order in which values in the dictionary passed to the writerow() method are written to file *f*.


fobj=open("student.csv","a")
dobj=csv.DictWriter(fobj,fieldnames=["rno","name","course"])
dobj.writeheader()
dobj.writerow({'rno':1,'name':'naresh','course':'python'})
fobj
student.csv
rno,name,course
1,naresh,python
"1,naresh,python"
dobj

Example:

```
# Program to create student.csv file

import csv

fobj=open("students.csv","a",newline='')
dwobj=csv.DictWriter(fobj,fieldnames=['rno','name','course'])
dwobj.writeheader()
while True:
    rno=int(input("Rollno :"))
    name=input("Name :")
    course=input("Course :")
    stud={'rno':rno,'name':name,'course':course}
    dwobj.writerow(stud)
    ans=input("Add another student?")
    if ans=="no":
        break

fobj.close()
```

Output

```
Rollno :1
Name :naresh
Course :python
Add another student?yes
Rollno :2
Name :suresh
```

```
Course :java
Add another student?yes
Rollno :3
Name :kishore
Course :c
Add another student?yes
Rollno :4
Name :ramesh
Course :oracle
Add another student?no
```

csv.DictReader(f)
Create an object that operates like a regular reader but maps the information in each row to a dict whose keys are given by the optional fieldnames parameter.

Example:

```
# Program to read data from student.csv file

import csv

fobj=open("students.csv","r")
drobj=csv.DictReader(fobj,fieldnames=["rno","name","course"])

for stud in drobj:
    print(stud['rno'],stud['name'],stud['course'])

fobj.close()
```

Output

```
rno name course
1 naresh python
2 suresh java
3 kishore c
4 ramesh oracle
```

JSON files (json module)

JSON stands for Java Script Object Notation. It is a standard or format used in web application development to interchange data between two applications written in same language or different language.

JSON, which stands for JavaScript Object Notation, is a lightweight, text-based format for storing and exchanging data. It's widely used in web applications and APIs for transferring data between a server and a client.

Json file is having extension .json

In order to work with json files python provides json module. It is a default module which comes with python software.

Json allows writing data in key and value format.

Json provides the following functions

1. dumps() → Writing inside file
2. dump() → Converting without writing inside file
3. loads() → Reading from file
4. load() → converting json string into python object

Example:

```
# Without writing inside file

import json

product={'prodid':[1,2,3],
        'prodname':['aa','bb','cc'],
        'price':[100.0,200.0,300.0]}
print(product,type(product))
json_product=json.dumps(product)
print(json_product,type(json_product))
prod_dict=json.loads(json_product)
print(prod_dict,type(prod_dict))
```

Output

```
{'prodid': [1, 2, 3], 'prodname': ['aa', 'bb', 'cc'], 'price': [100.0, 200.0, 300.0]}
<class 'dict'>
{"prodid": [1, 2, 3], "prodname": ["aa", "bb", "cc"], "price": [100.0, 200.0,
300.0]} <class 'str'>
{'prodid': [1, 2, 3], 'prodname': ['aa', 'bb', 'cc'], 'price': [100.0, 200.0, 300.0]}
<class 'dict'>
```

Example:

```
# Creating json file
import json

fobj=open("employee.json","a")
while True:
    empno=int(input("EmployeeNo :"))
    ename=input("EmployeeName :")
    sal=float(input("EmployeeSal :"))
    emp={'empno':empno,'ename':ename,'sal':sal}
    json.dump(emp,fobj)
    ans=input("Add another employee?")
    if ans=="no":
        break

fobj.close()
```

Output

```
EmployeeNo :1
EmployeeName :naresh
EmployeeSal :45000
Add another employee?yes
EmployeeNo :2
EmployeeName :kishore
EmployeeSal :54000
Add another employee?yes
EmployeeNo :3
EmployeeName :ramesh
EmployeeSal :89000
Add another employee?no
```

Example:

```
# Reading from json file

import json
fobj=open("employee.json","r")
emp=json.load(fobj)
print(emp)
print(type(emp))
for e in emp:
    print(e)

fobj.close()
```

Output

```
[{'empno': 1, 'ename': 'naresh', 'sal': 45000.0}, {'empno': 2, 'ename':
'kishore', 'sal': 54000.0}, {'empno': 3, 'ename': 'ramesh', 'sal': 89000.0}]
<class 'list'>
{'empno': 1, 'ename': 'naresh', 'sal': 45000.0}
{'empno': 2, 'ename': 'kishore', 'sal': 54000.0}
{'empno': 3, 'ename': 'ramesh', 'sal': 89000.0}
```

Binary file