Working with csv files in Python Programming

The CSV file or comma separated values file are one of the most widely used flat files to store and hare data across platforms. The columns are separated by comma and there is optional header row also which will indicate the name of each column. Python can read the CSV files using many modules. In this article we will see how the CSV library in python can be used to read and write a CSV file. We can also see the pandas library onl to read the CSV file.

Reading CSV file using csv module

We can get the CSV file from (https://www.guru99.com/python-csv.html)

Example

```
import csv
with open('C:\\iris.csv','rt')as file:
    csv_rows = csv.reader(file)
    for row in csv_rows:
        print(row)
```

Output

Running the above code gives us the following result −

```
['sepal.length', 'sepal.width', 'petal.length', 'petal.width', 'variety']
['5.1', '3.5', '1.4', '.2', 'Setosa']
['4.9', '3', '1.4', '.2', 'Setosa']
['4.7', '3.2', '1.3', '.2', 'Setosa']
['4.6', '3.1', '1.5', '.2', 'Setosa']
['5', '3.6', '1.4', '.2', 'Setosa']
['5.4', '3.9', '1.7', '.4', 'Setosa']
['4.6', '3.4', '1.4', '.3', 'Setosa']
……
……
```

Reading CSV file using Pandas

Pandas library can also be used to read the csv files. It has the method for reading csv which can be directly applied bypassing the path and the file name. Once the file is read it becomes a data set and then we can print different parts of the data set as required.

Example

```
import pandas as pd
datainput = pd.read_csv('C:\\iris.csv')
print("Given dataset values : \n", datainput)
#size of the dataset
print("\nThe size of the dataset is :\n",datainput.shape)
```

```
#printing few rows from the dataset
print("\n printing few rows from the dataset :\n",datainput[0:6])
```

Output

Running the above code gives us the following resul −

```
Given dataset values :
    sepal.length    sepal.width    petal.length    petal.width    variety
0            5.1            3.5             1.4            0.2     Setosa
1            4.9            3.0             1.4            0.2     Setosa
2            4.7            3.2             1.3            0.2     Setosa
3            4.6            3.1             1.5            0.2     Setosa
4            5.0            3.6             1.4            0.2     Setosa
..           ...            ...             ...            ...        ...
145          6.7            3.0             5.2            2.3  Virginica
146          6.3            2.5             5.0            1.9  Virginica
147          6.5            3.0             5.2            2.0  Virginica
148          6.2            3.4             5.4            2.3  Virginica
149          5.9            3.0             5.1            1.8  Virginica
[150 rows x 5 columns]
The size of the dataset is :
(150, 5)
printing few rows from the dataset :
     sepal.length      sepal.width    petal.length    petal.width    variety
0             5.1             3.5             1.4            0.2     Setosa
1             4.9             3.0             1.4            0.2     Setosa
2             4.7             3.2             1.3            0.2     Setosa
3             4.6             3.1             1.5            0.2     Setosa
4             5.0             3.6             1.4            0.2     Setosa
5             5.4             3.9             1.7            0.4     Setosa
```

Writing CSV file using csv module

To create a csv file we use Python lists we declare a data set containing each row as a list and all the rows are sublist in a big singer list. We also create another data set which just represents the header row. Then we use various methods like writerow() and csv.writer to finally write the file into the local system.

Example

```
import csv
data = ["Month", "1958", "1959", "1960"]
x = [
["JAN", 340, 360, 417],
["FEB", 318, 342, 391],
["MAR", 362, 406, 419],
["APR", 348, 396, 461],
["MAY", 363, 420, 472],
```

```
["JUN", 435, 472, 535],
["JUL", 491, 548, 622],
["AUG", 505, 559, 606],
["SEP", 404, 463, 508],
["OCT", 359, 407, 461],
["NOV", 310, 362, 390],
["DEC", 337, 405, 432],
]
y = "C:\\years.csv"
with open(y, 'w') as work:
    z = csv.writer(work)
    z.writerow(data)
    z.writerows(x)
```

Output

Running the above code gives us the following result −

```
Month,1958,1959,1960
JAN,340,360,417
FEB,318,342,391
MAR,362,406,419
APR,348,396,461
MAY,363,420,472
JUN,435,472,535
JUL,491,548,622
AUG,505,559,606
SEP,404,463,508
OCT,359,407,461
NOV,310,362,390
DEC,337,405,432
```

Writing CSV file using pandas

Uisng pandas is we create a data frame which country is the rows as well as the headers of the rows. Then we use the to_csv method which text the filename and path as parameters and drives the data to csv file.

Example

```
from pandas import DataFrame
C = {'Month': ['JAN','FEB', 'MAR'],
    '1958': ['345', '435', '545'],
    '1959': ['377', '135', '985'],
    '1960': ['498', '354', '765'],
}
df = DataFrame(C, columns= ["Month", "1958", "1959", "1960"])
export_csv = df.to_csv (r'C:\\years_p.csv', index = None, header=True) # here you
print (df)
```

Output

Running the above code gives us the following result −

```
   Month 1958 1959 1960
0  JAN   345  377  498
1  FEB   435  135  354
2  MAR   545  985  765
```