

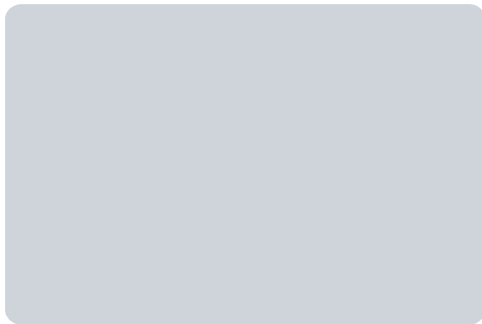
[Java  
Tutorial](#)[Index  
Posts](#)[Interview  
Questions](#)[Resources](#)

# Pandas to\_csv() – Convert DataFrame to CSV

Filed Under: [Pandas](#)

1 Comments

[Home](#) » [Python](#) » Pandas to\_csv() – Convert DataFrame to CSV



Pandas DataFrame to\_csv() function converts DataFrame into CSV data. We can pass a file object to write the CSV data into a file. Otherwise, the CSV data is returned in the string format.

## Pandas DataFrame to\_csv() Syntax

The syntax of DataFrame to\_csv() function is:

[< Previous](#)[Next >](#)

```
path_or_buf=None,  
    sep="," ,  
    na_rep="",  
  
    float_format=None,  
    columns=None,  
    header=True,  
    index=True,  
  
    index_label=None,  
    mode="w",  
    encoding=None,  
  
    compression="infer"  
,  
    quoting=None,  
    quotechar='"',  
  
    line_terminator=None,  
    chunksize=None,  
  
    date_format=None,  
  
    doublequote=True,
```

Some of the important parameters are:

**path\_or\_buf:** the file object to write the CSV data. If this argument is not provided, the CSV data is returned as a string.

**sep:** the delimiter for the CSV data. It should be a string of length 1, the default is a comma.

**na\_rep:** string representing null or missing values, default is empty string.

**header:** the allowed values are boolean or a list of string, default is True. If False, the column names are not written in the output. If a list of string, it's used to write the column names. The length of the list of string should be the same as the number of columns being written in the CSV file.

**index:** if True, index is included in the CSV data. If False, the index value is not written in the CSV output.

**index\_label:** used to specify the column name for index.

## Pandas DataFrame to CSV Examples

Let's look at some common examples of using to\_csv() function to convert DataFrame to CSV data.

### 1. Converting DataFrame to CSV String

```
import pandas as pd

d1 = {'Name':
      ['Pankaj',
       'Meghna'], 'ID': [1,
```

```
df =  
pd.DataFrame(d1)  
  
print('DataFrame:\n'  
      , df)  
  
# default CSV  
csv_data =  
df.to_csv()  
print('\nCSV  
String:\n',  
      csv_data)
```

Output:

```
DataFrame:  
      Name  ID Role  
0  Pankaj   1  CEO  
1  Meghna   2  CTO  
  
CSV String:  
      ,Name,ID,Role  
0,Pankaj,1,CEO  
1,Meghna,2,CTO
```

## 2. Specifying Delimiter for the CSV Output

```
csv_data =  
df.to_csv(sep='|')  
print(csv_data)
```

Output:



```
0|Pankaj|1|CEO  
1|Meghna|2|CTO
```

If the specified delimiter length is not 1, **TypeError: “delimiter” must be a 1-character string** is raised.

### 3. Selecting only few columns for CSV Output

```
csv_data =  
df.to_csv(columns=  
['Name', 'ID'])  
print(csv_data)
```

Output:

```
,Name,ID  
0,Pankaj,1  
1,Meghna,2
```

Notice that the index is not considered to be a valid column.

### 4. Ignoring Header Row in the CSV Output

```
csv_data =
```

Output:

```
0,Pankaj,1,CEO  
1,Meghna,2,CTO
```

## 5. Setting Custom Column Names in the CSV

```
csv_data =  
df.to_csv(header=  
['NAME', 'ID',  
'ROLE'])  
print(csv_data)
```

Output:

```
,NAME,ID,ROLE  
0,Pankaj,1,CEO  
1,Meghna,2,CTO
```

Again the index is not considered as the column of DataFrame object.

## 6. Skipping Index Column in CSV Output



```
csv_data =  
df.to_csv(index=False)  
print(csv_data)
```

Output:

```
Name,ID,Role  
Pankaj,1,CEO  
Meghna,2,CTO
```

## 7. Setting Index Column Name in the CSV

```
csv_data =  
df.to_csv(index_label='Sl No. ')  
print(csv_data)
```

Output:

```
Sl No.,Name,ID,Role  
0,Pankaj,1,CEO  
1,Meghna,2,CTO
```

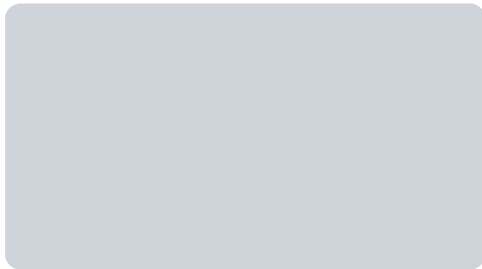
## 8. Converting DataFrame to CSV

```
with
open('csv_data.txt',
'w') as csv_file:

df.to_csv(path_or_bu
f=csv_file)
```

We are using [with statement](#) to open the file, it takes care of closing the file when the with statement block execution is finished.

This code snippet will create a CSV file with the following data.



## 9. Null, NA, or Missing Data Representation in the CSV Output

```
import pandas as pd

d1 = {'Name':
['Pankaj',
'Meghna'], 'ID': [1,
pd.NaT], 'Role':
```





```
print('DataFrame:\n', df)

csv_data = df.to_csv()
print('\nCSV String:\n', csv_data)

csv_data = df.to_csv(na_rep="None")
print('CSV String with Null Data Representation:\n', csv_data)
```

Output:

DataFrame:

	Name	ID	Role
0	Pankaj	1	NaT
1	Meghna	NaT	CTO

CSV String:

```
,Name,ID,Role
0,Pankaj,1,
1,Meghna,,CTO
```

CSV String with Null Data Representation:

```
,Name,ID,Role
0,Pankaj,1,None
1,Meghna,None,CTO
```

## References

[Pandas read\\_csv\(\) – Reading CSV File to DataFrame](#)

## DataFrame to\_csv() API Doc

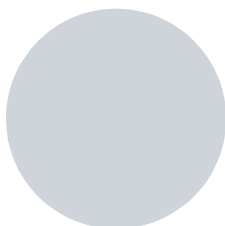
Pandas dropna() - Drop Null/NA Values from DataFrame

PREV

Pandas merge() - Merging Two DataFrame Objects

NEXT

Pankaj



I love Open Source technologies and writing about my experience about them is my passion.

Follow Author



## Comments



Kerem says:

August 17, 2020 at 2:30 pm

Dear Pankaj, I'm trying to write my data to csv file according to headers. To make it

< Previous    Next >



different values  
belong to position  
sensors of a robot.  
Could you help me to  
solve this issue? I  
would be more than  
happy.

Comments are closed.



JournalDev is one of the most popular websites for Java, Python, Android, and related technical articles. Our tutorials are regularly updated, error-free, and complete. Every month millions of developers like you visit JournalDev to read our tutorials.

JournalDev was founded by Pankaj Kumar in 2010 to share his experience and learnings with the whole world. He loves Open source technologies and writing on JournalDev



Most Popular

- Java / Java EE Tutorials
- Core Java Tutorial
- Python Tutorials
- Java Interview Questions
- Core Java Interview Questions
- Java Design Patterns
- Spring Tutorial

Favorite Sites

- AskPython
- CodeForGeek
- GoLangDocs
- LinuxForDevices
- Mkyong
- VM-Help
- WP-Design

