

# pandas.DataFrame.to\_csv

```
DataFrame.to_csv(self, 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, escapechar=None, decimal='.')
```

Write object to a comma-separated values (csv) file.

[\[source\]](#)

*Changed in version 0.24.0:* The order of arguments for Series was changed.

**path\_or\_buf** : *str or file handle, default None*

File path or object, if None is provided the result is returned as a string. If a file object is passed it should be opened with *newline=""*, disabling universal newlines.

*Changed in version 0.24.0:* Was previously named “path” for Series.

**sep** : *str, default ‘,’*

String of length 1. Field delimiter for the output file.

**na\_rep** : *str, default ‘’*

Missing data representation.

**float\_format** : *str, default None*

Format string for floating point numbers.

## Parameters:

**columns** : *sequence, optional*

Columns to write.

**header** : *bool or list of str, default True*

Write out the column names. If a list of strings is given it is assumed to be aliases for the column names.

*Changed in version 0.24.0:* Previously defaulted to False for Series.

**index** : *bool, default True*

Write row names (index).

**index\_label** : *str or sequence, or False, default None*

Column label for index column(s) if desired. If None is given, and *header* and *index* are True, then the index names are used. A sequence should be given if the object uses MultiIndex. If False do not print fields for index names. Use `index_label=False` for easier importing in R.

**mode** : *str*

Python write mode, default 'w'.

**encoding** : *str, optional*

A string representing the encoding to use in the output file, defaults to 'utf-8'.

**compression** : *str, default 'infer'*

Compression mode among the following possible values: {'infer', 'gzip', 'bz2', 'zip', 'xz', None}. If 'infer' and *path\_or\_buf* is path-like, then detect compression from the following extensions: '.gz', '.bz2', '.zip' or '.xz'. (otherwise no compression).

*Changed in version 0.24.0:* 'infer' option added and set to default.

**quoting** : *optional constant from csv module*

Defaults to `csv.QUOTE_MINIMAL`. If you have set a *float\_format* then floats are converted to strings and thus `csv.QUOTE_NONNUMERIC` will treat them as non-numeric.

**quotechar** : *str, default '"'*

String of length 1. Character used to quote fields.

**line\_terminator** : *str, optional*

The newline character or character sequence to use in the output file. Defaults to `os.linesep`, which depends on the OS in which this method is called ('n' for linux, 'rn' for Windows, i.e.).

*Changed in version 0.24.0.*

**chunksize** : *int or None*

Rows to write at a time.

**date\_format** : *str, default None*

Format string for datetime objects.

**doublequote** : *bool, default True*

Control quoting of *quotechar* inside a field.

**escapechar** : *str, default None*

String of length 1. Character used to escape *sep* and *quotechar* when appropriate.

**decimal** : *str, default '.'*

Character recognized as decimal separator. E.g. use ',' for European data.

None or str

**Returns:** If *path\_or\_buf* is None, returns the resulting csv format as a string. Otherwise returns None.

#### See also:

[read\\_csv](#)

Load a CSV file into a DataFrame.

[to\\_excel](#)

Write DataFrame to an Excel file.

## Examples

```
>>> df = pd.DataFrame({'name': ['Raphael', 'Donatello'],
...                    'mask': ['red', 'purple'],
...                    'weapon': ['sai', 'bo staff']})
>>> df.to_csv(index=False)
'name,mask,weapon\nRaphael,red,sai\nDonatello,purple,bo staff\n'
```