

Table Of Contents

- What's New in 0.25.2
- Installation
- Getting started
- User Guide
- Pandas ecosystem
- API reference
  - Input/output
  - General functions
  - Series
  - DataFrame
    - Constructor
    - Attributes and underlying data
    - Conversion
    - Indexing, iteration
    - Binary operator functions
    - Function application, GroupBy & window
    - Computations / descriptive stats
    - Reindexing / selection / label manipulation
    - Missing data handling
    - Reshaping, sorting, transposing
    - Combining / joining / merging
    - Time series-related
    - Plotting
    - Sparse accessor
    - Serialization / IO / conversion
    - Sparse
  - Pandas arrays
  - Panel
  - Index objects
  - Date offsets
  - Frequencies
  - Window
  - GroupBy
  - Resampling
  - Style
  - Plotting
  - General utility functions
  - Extensions
- Development
- Release Notes

Search

Go

Enter search terms or a module, class or function name.

# pandas.DataFrame.to\_string

`DataFrame.to_string(self, buf=None, columns=None, col_space=None, header=True, index=True, na_rep='NaN', formatters=None, float_format=None, sparsify=None, index_names=True, justify=None, max_rows=None, min_rows=None, max_cols=None, show_dimensions=False, decimal='.', line_width=None)` [\[source\]](#)

Render a DataFrame to a console-friendly tabular output.

Parameters:

**buf** : *StringIO-like, optional*  
Buffer to write to.

**columns** : *sequence, optional, default None*  
The subset of columns to write. Writes all columns by default.

**col\_space** : *int, optional*  
The minimum width of each column.

**header** : *bool, optional*  
Write out the column names. If a list of strings is given, it is assumed to be aliases for the column names.

**index** : *bool, optional, default True*  
Whether to print index (row) labels.

**na\_rep** : *str, optional, default 'NaN'*  
String representation of NaN to use.

**formatters** : *list or dict of one-param. functions, optional*  
Formatter functions to apply to columns' elements by position or name. The result of each function must be a unicode string. List must be of length equal to the number of columns.

**float\_format** : *one-parameter function, optional, default None*  
Formatter function to apply to columns' elements if they are floats. The result of this function must be a unicode string.

**sparsify** : *bool, optional, default True*  
Set to False for a DataFrame with a hierarchical index to print every multiindex key at each row.

**index\_names** : *bool, optional, default True*  
Prints the names of the indexes.

**justify** : *str, default None*  
How to justify the column labels. If None uses the option from the print configuration (controlled by `set_option`), 'right' out of the box. Valid values are

- left
- right
- center
- justify
- justify-all
- start
- end
- inherit
- match-parent
- initial
- unset.

**max\_rows** : *int, optional*  
Maximum number of rows to display in the console.

**min\_rows** : *int, optional*  
The number of rows to display in the console in a truncated repr (when number of rows is above *max\_rows*).

**max\_cols** : *int, optional*  
Maximum number of columns to display in the console.

**show\_dimensions** : *bool, default False*  
Display DataFrame dimensions (number of rows by number of columns).

**decimal** : *str, default '.'*  
Character recognized as decimal separator, e.g. ',' in Europe.  
*New in version 0.18.0.*

**line\_width** : *int, optional*  
Width to wrap a line in characters.

**Returns:** str (or unicode, depending on data and options)

String representation of the dataframe.

See also:

[to\\_html](#)  
Convert DataFrame to HTML.

Examples

```
>>> d = {'col1': [1, 2, 3], 'col2': [4, 5, 6]}
>>> df = pd.DataFrame(d)
>>> print(df.to_string())
  col1  col2
0     1     4
1     2     5
2     3     6
```