

pandas.DataFrame

```
class pandas.DataFrame(data=None, index=None, columns=None, dtype=None,
copy=None)
```

[\[source\]](#)

Two-dimensional, size-mutable, potentially heterogeneous tabular data.

Data structure also contains labeled axes (rows and columns). Arithmetic operations align on both row and column labels. Can be thought of as a dict-like container for Series objects. The primary pandas data structure.

Parameters:

data : *ndarray (structured or homogeneous), Iterable, dict, or DataFrame*

Dict can contain Series, arrays, constants, dataclass or list-like objects. If data is a dict, column order follows insertion-order. If a dict contains Series which have an index defined, it is aligned by its index. This alignment also occurs if data is a Series or a DataFrame itself. Alignment is done on Series/DataFrame inputs.

If data is a list of dicts, column order follows insertion-order.

index : *Index or array-like*

Index to use for resulting frame. Will default to RangeIndex if no indexing information part of input data and no index provided.

columns : *Index or array-like*

Column labels to use for resulting frame when data does not have them, defaulting to RangeIndex(0, 1, 2, ..., n). If data contains column labels, will perform column selection instead.

dtype : *dtype, default None*

Data type to force. Only a single dtype is allowed. If None, infer.

copy : *bool or None, default None*

Copy data from inputs. For dict data, the default of None behaves like `copy=True`. For

[Skip to main content](#)

is a dict containing one or more Series (possibly of different dtypes), `copy=False` will ensure that these inputs are not copied.

! *Changed in version 1.3.0.*

See also

[`DataFrame.from_records`](#)

Constructor from tuples, also record arrays.

[`DataFrame.from_dict`](#)

From dicts of Series, arrays, or dicts.

[`read_csv`](#)

Read a comma-separated values (csv) file into DataFrame.

[`read_table`](#)

Read general delimited file into DataFrame.

[`read_clipboard`](#)

Read text from clipboard into DataFrame.

Notes

Please reference the [User Guide](#) for more information.

Examples

Constructing DataFrame from a dictionary.

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

Notice that the inferred dtype is int64.

```
>>> df.dtypes
col1    int64
```

[Skip to main content](#)

```
col2    int64
dtype: object
```

To enforce a single dtype:

```
>>> df = pd.DataFrame(data=d, dtype=np.int8)
>>> df.dtypes
col1    int8
col2    int8
dtype: object
```

Constructing DataFrame from a dictionary including Series:

```
>>> d = {'col1': [0, 1, 2, 3], 'col2': pd.Series([2, 3], index=[2, 3])}
>>> pd.DataFrame(data=d, index=[0, 1, 2, 3])
   col1  col2
0      0   NaN
1      1   NaN
2      2   2.0
3      3   3.0
```

Constructing DataFrame from numpy ndarray:

```
>>> df2 = pd.DataFrame(np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]]),
...                     columns=['a', 'b', 'c'])
>>> df2
   a  b  c
0  1  2  3
1  4  5  6
2  7  8  9
```

Constructing DataFrame from a numpy ndarray that has labeled columns:

```
>>> data = np.array([(1, 2, 3), (4, 5, 6), (7, 8, 9)],
...                  dtype=[("a", "i4"), ("b", "i4"), ("c", "i4")])
>>> df3 = pd.DataFrame(data, columns=['c', 'a'])
>>> df3
   c  a
0  3  1
1  6  4
2  9  7
```

```
>>> from dataclasses import make_dataclass
>>> Point = make_dataclass("Point", [("x", int), ("y", int)])
>>> pd.DataFrame([Point(0, 0), Point(0, 3), Point(2, 3)])
   x  y
0  0  0
1  0  3
2  2  3
```

Constructing DataFrame from Series/DataFrame:

```
>>> ser = pd.Series([1, 2, 3], index=["a", "b", "c"])
>>> df = pd.DataFrame(data=ser, index=["a", "c"])
>>> df
   0
a  1
c  3
```

```
>>> df1 = pd.DataFrame([1, 2, 3], index=["a", "b", "c"], columns=["x"])
>>> df2 = pd.DataFrame(data=df1, index=["a", "c"])
>>> df2
   x
a  1
c  3
```

Attributes

<code>T</code>	The transpose of the DataFrame.
<code>at</code>	Access a single value for a row/column label pair.
<code>attrs</code>	Dictionary of global attributes of this dataset.
<code>axes</code>	Return a list representing the axes of the DataFrame.
<code>columns</code>	The column labels of the DataFrame.
<code>dtypes</code>	Return the dtypes in the DataFrame.
<code>empty</code>	Indicator whether Series/DataFrame is empty.
<code>flags</code>	Get the properties associated with this pandas object.

[Skip to main content](#)

<code>iloc</code>	(DEPRECATED) Purely integer-location based indexing for selection by position.
<code>index</code>	The index (row labels) of the DataFrame.
<code>loc</code>	Access a group of rows and columns by label(s) or a boolean array.
<code>ndim</code>	Return an int representing the number of axes / array dimensions.
<code>shape</code>	Return a tuple representing the dimensionality of the DataFrame.
<code>size</code>	Return an int representing the number of elements in this object.
<code>style</code>	Returns a Styler object.
<code>values</code>	Return a Numpy representation of the DataFrame.

Methods

<code>abs</code> ()	Return a Series/DataFrame with absolute numeric value of each element.
<code>add</code> (other[, axis, level, fill_value])	Get Addition of dataframe and other, element-wise (binary operator <i>add</i>).
<code>add_prefix</code> (prefix[, axis])	Prefix labels with string <i>prefix</i> .
<code>add_suffix</code> (suffix[, axis])	Suffix labels with string <i>suffix</i> .
<code>agg</code> ([func, axis])	Aggregate using one or more operations over the specified axis.
<code>aggregate</code> ([func, axis])	Aggregate using one or more operations over the specified axis.
<code>align</code> (other[, join, axis, level, copy, ...])	Align two objects on their axes with the specified join method.
<code>all</code> ([axis, bool_only, skipna])	Return whether all elements are True, potentially over an axis.

[Skip to main content](#)

<code>any</code> (*[, axis, bool_only, skipna])	Return whether any element is True, potentially over an axis.
<code>apply</code> (func[, axis, raw, result_type, args, ...])	Apply a function along an axis of the DataFrame.
<code>applymap</code> (func[, na_action])	(DEPRECATED) Apply a function to a Dataframe elementwise.
<code>asfreq</code> (freq[, method, how, normalize, ...])	Convert time series to specified frequency.
<code>asof</code> (where[, subset])	Return the last row(s) without any NaNs before <i>where</i> .
<code>assign</code> (**kwargs)	Assign new columns to a DataFrame.
<code>astype</code> (dtype[, copy, errors])	Cast a pandas object to a specified dtype <code>dtype</code> .
<code>at_time</code> (time[, asof, axis])	Select values at particular time of day (e.g., 9:30AM).
<code>backfill</code> (*[, axis, inplace, limit, downcast])	(DEPRECATED) Fill NA/NaN values by using the next valid observation to fill the gap.
<code>between_time</code> (start_time, end_time[, ...])	Select values between particular times of the day (e.g., 9:00-9:30 AM).
<code>bfill</code> (*[, axis, inplace, limit, limit_area, ...])	Fill NA/NaN values by using the next valid observation to fill the gap.
<code>bool</code> ()	(DEPRECATED) Return the bool of a single element Series or DataFrame.
<code>boxplot</code> ([column, by, ax, fontsize, rot, ...])	Make a box plot from DataFrame columns.
<code>clip</code> ([lower, upper, axis, inplace])	Trim values at input threshold(s).

[Skip to main content](#)

<code>combine</code> (other, func[, fill_value, overwrite])	Perform column-wise combine with another DataFrame.
<code>combine_first</code> (other)	Update null elements with value in the same location in <i>other</i> .
<code>compare</code> (other[, align_axis, keep_shape, ...])	Compare to another DataFrame and show the differences.
<code>convert_dtypes</code> ([infer_objects, ...])	Convert columns to the best possible dtypes using dtypes supporting <code>pd.NA</code> .
<code>copy</code> ([deep])	Make a copy of this object's indices and data.
<code>corr</code> ([method, min_periods, numeric_only])	Compute pairwise correlation of columns, excluding NA/null values.
<code>corrwith</code> (other[, axis, drop, method, ...])	Compute pairwise correlation.
<code>count</code> ([axis, numeric_only])	Count non-NA cells for each column or row.
<code>cov</code> ([min_periods, ddof, numeric_only])	Compute pairwise covariance of columns, excluding NA/null values.
<code>cummax</code> ([axis, skipna])	Return cumulative maximum over a DataFrame or Series axis.
<code>cummin</code> ([axis, skipna])	Return cumulative minimum over a DataFrame or Series axis.
<code>cumprod</code> ([axis, skipna])	Return cumulative product over a DataFrame or Series axis.
<code>cumsum</code> ([axis, skipna])	Return cumulative sum over a DataFrame or Series axis.
<code>describe</code> ([percentiles, include, exclude])	Generate descriptive statistics.

[Skip to main content](#)

<code>div</code> (other[, axis, level, fill_value])	Get Floating division of dataframe and other, element-wise (binary operator <i>truediv</i>).
<code>divide</code> (other[, axis, level, fill_value])	Get Floating division of dataframe and other, element-wise (binary operator <i>truediv</i>).
<code>dot</code> (other)	Compute the matrix multiplication between the DataFrame and other.
<code>drop</code> ([labels, axis, index, columns, level, ...])	Drop specified labels from rows or columns.
<code>drop_duplicates</code> ([subset, keep, inplace, ...])	Return DataFrame with duplicate rows removed.
<code>droplevel</code> (level[, axis])	Return Series/DataFrame with requested index / column level(s) removed.
<code>dropna</code> (*[, axis, how, thresh, subset, ...])	Remove missing values.
<code>uplicated</code> ([subset, keep])	Return boolean Series denoting duplicate rows.
<code>eq</code> (other[, axis, level])	Get Equal to of dataframe and other, element-wise (binary operator <i>eq</i>).
<code>equals</code> (other)	Test whether two objects contain the same elements.
<code>eval</code> (expr, *, inplace)	Evaluate a string describing operations on DataFrame columns.
<code>ewm</code> ([com, span, halflife, alpha, ...])	Provide exponentially weighted (EW) calculations.
<code>expanding</code> ([min_periods, axis, method])	Provide expanding window calculations.
<code>explode</code> (column[, ignore_index])	Transform each element of a list-like to a

[Skip to main content](#)

<code>ffill</code> (*[, axis, inplace, limit, limit_area, ...])	Fill NA/NaN values by propagating the last valid observation to next valid.
<code>fillna</code> ([value, method, axis, inplace, ...])	Fill NA/NaN values using the specified method.
<code>filter</code> ([items, like, regex, axis])	Subset the dataframe rows or columns according to the specified index labels.
<code>first</code> (offset)	(DEPRECATED) Select initial periods of time series data based on a date offset.
<code>first_valid_index</code> ()	Return index for first non-NA value or None, if no non-NA value is found.
<code>floordiv</code> (other[, axis, level, fill_value])	Get Integer division of dataframe and other, element-wise (binary operator <i>floordiv</i>).
<code>from_dict</code> (data[, orient, dtype, columns])	Construct DataFrame from dict of array-like or dicts.
<code>from_records</code> (data[, index, exclude, ...])	Convert structured or record ndarray to DataFrame.
<code>ge</code> (other[, axis, level])	Get Greater than or equal to of dataframe and other, element-wise (binary operator <i>ge</i>).
<code>get</code> (key[, default])	Get item from object for given key (ex: DataFrame column).
<code>groupby</code> ([by, axis, level, as_index, sort, ...])	Group DataFrame using a mapper or by a Series of columns.
<code>gt</code> (other[, axis, level])	Get Greater than of dataframe and other, element-wise (binary operator <i>gt</i>).
<code>head</code> ([n])	Return the first <i>n</i> rows.

[Skip to main content](#)

<code>hist</code> ([column, by, grid, xlabelsize, xrot, ...])	Make a histogram of the DataFrame's columns.
<code>idxmax</code> ([axis, skipna, numeric_only])	Return index of first occurrence of maximum over requested axis.
<code>idxmin</code> ([axis, skipna, numeric_only])	Return index of first occurrence of minimum over requested axis.
<code>infer_objects</code> ([copy])	Attempt to infer better dtypes for object columns.
<code>info</code> ([verbose, buf, max_cols, memory_usage, ...])	Print a concise summary of a DataFrame.
<code>insert</code> (loc, column, value[, allow_duplicates])	Insert column into DataFrame at specified location.
<code>interpolate</code> ([method, axis, limit, inplace, ...])	Fill NaN values using an interpolation method.
<code>isetitem</code> (loc, value)	Set the given value in the column with position <i>loc</i> .
<code>isin</code> (values)	Whether each element in the DataFrame is contained in values.
<code>isna</code> ()	Detect missing values.
<code>isnull</code> ()	DataFrame.isnull is an alias for DataFrame.isna.
<code>items</code> ()	Iterate over (column name, Series) pairs.
<code>iterrows</code> ()	Iterate over DataFrame rows as (index, Series) pairs.
<code>itertuples</code> ([index, name])	Iterate over DataFrame rows as namedtuples.
<code>join</code> (other[, on, how, lsuffix, rsuffix, ...])	Join columns of another DataFrame.

[Skip to main content](#)

<code>keys()</code>	Get the 'info axis' (see Indexing for more).
<code>kurt</code> ([axis, skipna, numeric_only])	Return unbiased kurtosis over requested axis.
<code>kurtosis</code> ([axis, skipna, numeric_only])	Return unbiased kurtosis over requested axis.
<code>last</code> (offset)	(DEPRECATED) Select final periods of time series data based on a date offset.
<code>last_valid_index</code> ()	Return index for last non-NA value or None, if no non-NA value is found.
<code>le</code> (other[, axis, level])	Get Less than or equal to of dataframe and other, element-wise (binary operator <i>le</i>).
<code>lt</code> (other[, axis, level])	Get Less than of dataframe and other, element-wise (binary operator <i>lt</i>).
<code>map</code> (func[, na_action])	Apply a function to a Dataframe elementwise.
<code>mask</code> (cond[, other, inplace, axis, level])	Replace values where the condition is True.
<code>max</code> ([axis, skipna, numeric_only])	Return the maximum of the values over the requested axis.
<code>mean</code> ([axis, skipna, numeric_only])	Return the mean of the values over the requested axis.
<code>median</code> ([axis, skipna, numeric_only])	Return the median of the values over the requested axis.
<code>melt</code> ([id_vars, value_vars, var_name, ...])	Unpivot a DataFrame from wide to long format, optionally leaving identifiers set.

[Skip to main content](#)

<code>memory_usage</code> ([index, deep])	Return the memory usage of each column in bytes.
<code>merge</code> (right[, how, on, left_on, right_on, ...])	Merge DataFrame or named Series objects with a database-style join.
<code>min</code> ([axis, skipna, numeric_only])	Return the minimum of the values over the requested axis.
<code>mod</code> (other[, axis, level, fill_value])	Get Modulo of dataframe and other, element-wise (binary operator <i>mod</i>).
<code>mode</code> ([axis, numeric_only, dropna])	Get the mode(s) of each element along the selected axis.
<code>mul</code> (other[, axis, level, fill_value])	Get Multiplication of dataframe and other, element-wise (binary operator <i>mul</i>).
<code>multiply</code> (other[, axis, level, fill_value])	Get Multiplication of dataframe and other, element-wise (binary operator <i>mul</i>).
<code>ne</code> (other[, axis, level])	Get Not equal to of dataframe and other, element-wise (binary operator <i>ne</i>).
<code>nlargest</code> (n, columns[, keep])	Return the first <i>n</i> rows ordered by <i>columns</i> in descending order.
<code>notna</code> ()	Detect existing (non-missing) values.
<code>notnull</code> ()	DataFrame.notnull is an alias for DataFrame.notna.
<code>nsmallest</code> (n, columns[, keep])	Return the first <i>n</i> rows ordered by <i>columns</i> in ascending order.
<code>nunique</code> ([axis, dropna])	Count number of distinct elements in specified axis.

[Skip to main content](#)

<code>pad</code> (*[, axis, inplace, limit, downcast])	(DEPRECATED) Fill NA/NaN values by propagating the last valid observation to next valid.
<code>pct_change</code> ([periods, fill_method, limit, freq])	Fractional change between the current and a prior element.
<code>pipe</code> (func, *args, **kwargs)	Apply chainable functions that expect Series or DataFrames.
<code>pivot</code> (*, columns[, index, values])	Return reshaped DataFrame organized by given index / column values.
<code>pivot_table</code> ([values, index, columns, ...])	Create a spreadsheet-style pivot table as a DataFrame.
<code>pop</code> (item)	Return item and drop from frame.
<code>pow</code> (other[, axis, level, fill_value])	Get Exponential power of dataframe and other, element-wise (binary operator <i>pow</i>).
<code>prod</code> ([axis, skipna, numeric_only, min_count])	Return the product of the values over the requested axis.
<code>product</code> ([axis, skipna, numeric_only, min_count])	Return the product of the values over the requested axis.
<code>quantile</code> ([q, axis, numeric_only, ...])	Return values at the given quantile over requested axis.
<code>query</code> (expr, *[, inplace])	Query the columns of a DataFrame with a boolean expression.
<code>radd</code> (other[, axis, level, fill_value])	Get Addition of dataframe and other, element-wise (binary operator <i>radd</i>).
<code>rank</code> ([axis, method, numeric_only, ...])	Compute numerical data ranks (1 through n) along axis.

[Skip to main content](#)

<code>rdiv</code> (other[, axis, level, fill_value])	Get Floating division of dataframe and other, element-wise (binary operator <i>rtruediv</i>).
<code>reindex</code> ([labels, index, columns, axis, ...])	Conform DataFrame to new index with optional filling logic.
<code>reindex_like</code> (other[, method, copy, limit, ...])	Return an object with matching indices as other object.
<code>rename</code> ([mapper, index, columns, axis, copy, ...])	Rename columns or index labels.
<code>rename_axis</code> ([mapper, index, columns, axis, ...])	Set the name of the axis for the index or columns.
<code>reorder_levels</code> (order[, axis])	Rearrange index levels using input order.
<code>replace</code> ([to_replace, value, inplace, limit, ...])	Replace values given in <i>to_replace</i> with <i>value</i> .
<code>resample</code> (rule[, axis, closed, label, ...])	Resample time-series data.
<code>reset_index</code> ([level, drop, inplace, ...])	Reset the index, or a level of it.
<code>rfloordiv</code> (other[, axis, level, fill_value])	Get Integer division of dataframe and other, element-wise (binary operator <i>rfloordiv</i>).
<code>rmod</code> (other[, axis, level, fill_value])	Get Modulo of dataframe and other, element-wise (binary operator <i>rmod</i>).
<code>rmul</code> (other[, axis, level, fill_value])	Get Multiplication of dataframe and other, element-wise (binary operator <i>rmul</i>).
<code>rolling</code> (window[, min_periods, center, ...])	Provide rolling window calculations.
<code>round</code> ([decimals])	Round a DataFrame to a variable number of decimal places.

[Skip to main content](#)

<code>rpow</code> (other[, axis, level, fill_value])	Get Exponential power of dataframe and other, element-wise (binary operator <i>rpow</i>).
<code>rsub</code> (other[, axis, level, fill_value])	Get Subtraction of dataframe and other, element-wise (binary operator <i>rsub</i>).
<code>rtruediv</code> (other[, axis, level, fill_value])	Get Floating division of dataframe and other, element-wise (binary operator <i>rtruediv</i>).
<code>sample</code> ([n, frac, replace, weights, ...])	Return a random sample of items from an axis of object.
<code>select_dtypes</code> ([include, exclude])	Return a subset of the DataFrame's columns based on the column dtypes.
<code>sem</code> ([axis, skipna, ddof, numeric_only])	Return unbiased standard error of the mean over requested axis.
<code>set_axis</code> (labels, *, axis, copy)	Assign desired index to given axis.
<code>set_flags</code> (*[, copy, allows_duplicate_labels])	Return a new object with updated flags.
<code>set_index</code> (keys, *, drop, append, inplace, ...)	Set the DataFrame index using existing columns.
<code>shift</code> ([periods, freq, axis, fill_value, suffix])	Shift index by desired number of periods with an optional time <i>freq</i> .
<code>skew</code> ([axis, skipna, numeric_only])	Return unbiased skew over requested axis.
<code>sort_index</code> (*[, axis, level, ascending, ...])	Sort object by labels (along an axis).
<code>sort_values</code> (by, *, axis, ascending, ...)	Sort by the values along either axis.
<code>squeeze</code> ([axis])	Squeeze 1 dimensional axis objects into scalars.

[Skip to main content](#)

<code>stack</code> ([level, dropna, sort, future_stack])	Stack the prescribed level(s) from columns to index.
<code>std</code> ([axis, skipna, ddof, numeric_only])	Return sample standard deviation over requested axis.
<code>sub</code> (other[, axis, level, fill_value])	Get Subtraction of dataframe and other, element-wise (binary operator <i>sub</i>).
<code>subtract</code> (other[, axis, level, fill_value])	Get Subtraction of dataframe and other, element-wise (binary operator <i>sub</i>).
<code>sum</code> ([axis, skipna, numeric_only, min_count])	Return the sum of the values over the requested axis.
<code>swapaxes</code> (axis1, axis2[, copy])	(DEPRECATED) Interchange axes and swap values axes appropriately.
<code>swaplevel</code> ([i, j, axis])	Swap levels i and j in a <code>MultiIndex</code> .
<code>tail</code> ([n])	Return the last <i>n</i> rows.
<code>take</code> (indices[, axis])	Return the elements in the given <i>positional</i> indices along an axis.
<code>to_clipboard</code> (*[, excel, sep])	Copy object to the system clipboard.
<code>to_csv</code> ([path_or_buf, sep, na_rep, ...])	Write object to a comma-separated values (csv) file.
<code>to_dict</code> ([orient, into, index])	Convert the DataFrame to a dictionary.
<code>to_excel</code> (excel_writer, *[, sheet_name, ...])	Write object to an Excel sheet.
<code>to_feather</code> (path, **kwargs)	Write a DataFrame to the binary Feather format.
<code>to_gbq</code> (destination_table, *[, project_id, ...])	(DEPRECATED) Write a DataFrame to a Google BigQuery table.

[Skip to main content](#)

<code>to_hdf</code> (path_or_buf, *, key[, mode, ...])	Write the contained data to an HDF5 file using HDFStore.
<code>to_html</code> ([buf, columns, col_space, header, ...])	Render a DataFrame as an HTML table.
<code>to_json</code> ([path_or_buf, orient, date_format, ...])	Convert the object to a JSON string.
<code>to_latex</code> ([buf, columns, header, index, ...])	Render object to a LaTeX tabular, longtable, or nested table.
<code>to_markdown</code> ([buf, mode, index, storage_options])	Print DataFrame in Markdown-friendly format.
<code>to_numpy</code> ([dtype, copy, na_value])	Convert the DataFrame to a NumPy array.
<code>to_orc</code> ([path, engine, index, engine_kwargs])	Write a DataFrame to the ORC format.
<code>to_parquet</code> ([path, engine, compression, ...])	Write a DataFrame to the binary parquet format.
<code>to_period</code> ([freq, axis, copy])	Convert DataFrame from DatetimeIndex to PeriodIndex.
<code>to_pickle</code> (path, *[, compression, protocol, ...])	Pickle (serialize) object to file.
<code>to_records</code> ([index, column_dtypes, index_dtypes])	Convert DataFrame to a NumPy record array.
<code>to_sql</code> (name, con, *[, schema, if_exists, ...])	Write records stored in a DataFrame to a SQL database.
<code>to_stata</code> (path, *[, convert_dates, ...])	Export DataFrame object to Stata dta format.
<code>to_string</code> ([buf, columns, col_space, header, ...])	Render a DataFrame to a console-friendly tabular output.
<code>to_timestamp</code> ([freq, how, axis, copy])	Cast to DatetimeIndex of timestamps, at <i>beginning</i> of period.

[Skip to main content](#)

`to_xarray` ()

Return an xarray object from the pandas object.

`to_xml` ([path_or_buffer, index, root_name, ...])

Render a DataFrame to an XML document.

`transform` (func[, axis])

Call `func` on self producing a DataFrame with the same axis shape as self.

`transpose` (*args[, copy])

Transpose index and columns.

`truediv` (other[, axis, level, fill_value])

Get Floating division of dataframe and other, element-wise (binary operator *truediv*).

`truncate` ([before, after, axis, copy])

Truncate a Series or DataFrame before and after some index value.

`tz_convert` (tz[, axis, level, copy])

Convert tz-aware axis to target time zone. [pandas.DataFrame.index](#)

[Previous](#)
[DataFrame](#)

© 2024, pandas via [NumFOCUS, Inc.](#) Hosted by [OVHcloud](#).

Created using [Sphinx](#) 8.0.2.

Built with the [PyData Sphinx Theme](#)
0.14.4.