

pandas.DataFrame

`class pandas.DataFrame(data=None, index=None, columns=None, dtype=None, copy=False)` [\[source\]](#)

Two-dimensional size-mutable, potentially heterogeneous tabular data structure with 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

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

Dict can contain Series, arrays, constants, or list-like objects

index : *Index or array-like*

Index to use for resulting frame. Will default to `np.arange(n)` if no indexing information part of input data and no index provided

Parameters:

columns : *Index or array-like*

Column labels to use for resulting frame. Will default to `np.arange(n)` if no column labels are provided

dtype : *dtype, default None*

Data type to force, otherwise infer

copy : *boolean, default False*

Copy data from inputs. Only affects DataFrame / 2d ndarray input

See also:

[DataFrame.from_records](#)

constructor from tuples, also record arrays

[DataFrame.from_dict](#)

from dicts of Series, arrays, or dicts

[DataFrame.from_items](#)

from sequence of (key, value) pairs

[pandas.read_csv](#), [pandas.read_table](#), [pandas.read_clipboard](#)

Examples

```
>>> d = {'col1': ts1, 'col2': ts2}
>>> df = DataFrame(data=d, index=index)
>>> df2 = DataFrame(np.random.randn(10, 5))
>>> df3 = DataFrame(np.random.randn(10, 5),
...                  columns=['a', 'b', 'c', 'd', 'e'])
```

Attributes

[Scroll To Top](#)

T	Transpose index and columns
----------	-----------------------------

at	Fast label-based scalar accessor
-----------	----------------------------------

axes	Return a list with the row axis labels and column axis labels as the only members.
blocks	Internal property, property synonym for <code>as_blocks()</code>
dtypes	Return the dtypes in this object.
empty	True if NDFrame is entirely empty [no items], meaning any of the axes are of length 0.
ftypes	Return the ftypes (indication of sparse/dense and dtype) in this object.
iat	Fast integer location scalar accessor.
iloc	Purely integer-location based indexing for selection by position.
is_copy	
ix	A primarily label-location based indexer, with integer position fallback.
loc	Purely label-location based indexer for selection by label.
ndim	Number of axes / array dimensions
shape	Return a tuple representing the dimensionality of the DataFrame.
size	number of elements in the NDFrame
style	Property returning a Styler object containing methods for building a styled HTML representation fo the DataFrame.
values	Numpy representation of NDFrame

Methods

abs()	Return an object with absolute value taken—only applicable to objects that are all numeric.
add(other[, axis, level, fill_value])	Addition of dataframe and other, element-wise (binary operator <i>add</i>).
add_prefix(prefix)	Concatenate prefix string with panel items names.
add_suffix(suffix)	Concatenate suffix string with panel items names.
agg(func[, axis])	Aggregate using callable, string, dict, or list of string/callables
aggregate(func[, axis])	Aggregate using callable, string, dict, or list of string/callables
align(other[, join, axis, level, copy, ...])	Align two object on their axes with the
all([axis, bool_only, skipna, level])	Return whether all elements are True over requested axis
any([axis, bool_only, skipna, level])	Return whether any element is True over requested axis
append(other[, ignore_index, verify_integrity])	Append rows of <i>other</i> to the end of this frame, returning a new object.
apply(func[, axis, broadcast, raw, reduce, args])	Applies function along input axis of DataFrame.
applymap(func)	Apply a function to a DataFrame that is intended to operate elementwise, i.e.
as_blocks([copy])	Convert the frame to a dict of dtype -> Constructor Types that each has a homogeneous dtype.
as_matrix([columns])	Convert the frame to its Numpy-array representation.
asfreq(freq[, method, how, normalize, ...])	Convert TimeSeries to specified frequency.
asof(when[, subset])	The last row without any NaN is taken (or the last row without
assign(**kwargs)	Assign new columns to a DataFrame, returning a new object (a copy) with all the original columns in addition to the new ones.
astype(dtype[, copy, errors])	Cast object to input numpy.dtype

[Scroll To Top](#)

<code>at_time(time[, asof])</code>	Select values at particular time of day (e.g.
<code>between_time(start_time, end_time[, ...])</code>	Select values between particular times of the day (e.g., 9:00-9:30 AM).
<code>bfill([axis, inplace, limit, downcast])</code>	Synonym for <code>DataFrame.fillna(method='bfill')</code>
<code>bool()</code>	Return the bool of a single element PandasObject.
<code>boxplot([column, by, ax, fontsize, rot, ...])</code>	Make a box plot from DataFrame column optionally grouped by some columns or
<code>clip([lower, upper, axis])</code>	Trim values at input threshold(s).
<code>clip_lower(threshold[, axis])</code>	Return copy of the input with values below given value(s) truncated.
<code>clip_upper(threshold[, axis])</code>	Return copy of input with values above given value(s) truncated.
<code>combine(other, func[, fill_value, overwrite])</code>	Add two DataFrame objects and do not propagate NaN values, so if for a
<code>combine_first(other)</code>	Combine two DataFrame objects and default to non-null values in frame calling the method.
<code>compound([axis, skipna, level])</code>	Return the compound percentage of the values for the requested axis
<code>consolidate([inplace])</code>	DEPRECATED: consolidate will be an internal implementation only.
<code>convert_objects([convert_dates, ...])</code>	Deprecated.
<code>copy([deep])</code>	Make a copy of this objects data.
<code>corr([method, min_periods])</code>	Compute pairwise correlation of columns, excluding NA/null values
<code>corrwith(other[, axis, drop])</code>	Compute pairwise correlation between rows or columns of two DataFrame objects.
<code>count([axis, level, numeric_only])</code>	Return Series with number of non-NA/null observations over requested axis.
<code>cov([min_periods])</code>	Compute pairwise covariance of columns, excluding NA/null values
<code>cummax([axis, skipna])</code>	Return cumulative max over requested axis.
<code>cummin([axis, skipna])</code>	Return cumulative minimum over requested axis.
<code>cumprod([axis, skipna])</code>	Return cumulative product over requested axis.
<code>cumsum([axis, skipna])</code>	Return cumulative sum over requested axis.
<code>describe([percentiles, include, exclude])</code>	Generates descriptive statistics that summarize the central tendency, dispersion and shape of a dataset's distribution, excluding NaN values.
<code>diff([periods, axis])</code>	1st discrete difference of object
<code>div(other[, axis, level, fill_value])</code>	Floating division of dataframe and other, element-wise (binary operator <i>truediv</i>).
<code>divide(other[, axis, level, fill_value])</code>	Floating division of dataframe and other, element-wise (binary operator <i>truediv</i>).
<code>dot(other)</code>	Matrix multiplication with DataFrame or Series objects
<code>drop(labels[, axis, level, inplace, errors])</code>	Return new object with labels in requested axis removed.
<code>drop_duplicates([subset, keep, inplace])</code>	Return DataFrame with duplicate rows removed, optionally only
<code>dropna([axis, how, thresh, subset, inplace])</code>	Return object with labels on given axis omitted where alternately any Scroll To Top
<code>uplicated([subset, keep])</code>	Return boolean Series denoting duplicate rows, optionally only
<code>eq(other[, axis, level])</code>	Wrapper for flexible comparison methods eq

<code>equals(other)</code>	Determines if two NDFrame objects contain the same elements.
<code>eval(expr[, inplace])</code>	Evaluate an expression in the context of the calling DataFrame instance.
<code>ewm([com, span, halflife, alpha, ...])</code>	Provides exponential weighted functions
<code>expanding([min_periods, freq, center, axis])</code>	Provides expanding transformations.
<code>ffill([axis, inplace, limit, downcast])</code>	Synonym for <code>DataFrame.fillna(method='ffill')</code>
<code>fillna([value, method, axis, inplace, ...])</code>	Fill NA/NaN values using the specified method
<code>filter([items, like, regex, axis])</code>	Subset rows or columns of dataframe according to labels in the specified index.
<code>first(offset)</code>	Convenience method for subsetting initial periods of time series data based on a date offset.
<code>first_valid_index()</code>	Return label for first non-NA/null value
<code>floordiv(other[, axis, level, fill_value])</code>	Integer division of dataframe and other, element-wise (binary operator <i>floordiv</i>).
<code>from_csv(path[, header, sep, index_col, ...])</code>	Read CSV file (DISCOURAGED, please use <code>pandas.read_csv()</code> instead).
<code>from_dict(data[, orient, dtype])</code>	Construct DataFrame from dict of array-like or dicts
<code>from_items(items[, columns, orient])</code>	Convert (key, value) pairs to DataFrame.
<code>from_records(data[, index, exclude, ...])</code>	Convert structured or record ndarray to DataFrame
<code>ge(other[, axis, level])</code>	Wrapper for flexible comparison methods ge
<code>get(key[, default])</code>	Get item from object for given key (DataFrame column, Panel slice, etc.).
<code>get_dtype_counts()</code>	Return the counts of dtypes in this object.
<code>get_ftype_counts()</code>	Return the counts of ftypes in this object.
<code>get_value(index, col[, takeable])</code>	Quickly retrieve single value at passed column and index
<code>get_values()</code>	same as values (but handles sparseness conversions)
<code>groupby([by, axis, level, as_index, sort, ...])</code>	Group series using mapper (dict or key function, apply given function to group, return result as series) or by a series of columns.
<code>gt(other[, axis, level])</code>	Wrapper for flexible comparison methods gt
<code>head([n])</code>	Returns first n rows
<code>hist(data[, column, by, grid, xlabelsize, ...])</code>	Draw histogram of the DataFrame's series using matplotlib / pylab.
<code>idxmax([axis, skipna])</code>	Return index of first occurrence of maximum over requested axis.
<code>idxmin([axis, skipna])</code>	Return index of first occurrence of minimum over requested axis.
<code>info([verbose, buf, max_cols, memory_usage, ...])</code>	Concise summary of a DataFrame.
<code>insert(loc, column, value[, allow_duplicates])</code>	Insert column into DataFrame at specified location.
<code>interpolate([method, axis, limit, inplace, ...])</code>	Interpolate values according to different methods.
<code>isin(values)</code>	Return boolean DataFrame showing whether each element in the DataFrame is contained in values.
<code>isnull()</code>	Return a boolean same-sized object indicating if the values are null.
<code>items()</code>	Iterator over (column name, Series) pairs.
<code>iteritems()</code>	Iterator over (column name, Series) pairs.

[Scroll To Top](#)

<code>iterrows()</code>	Iterate over DataFrame rows as (index, Series) pairs.
<code>itertuples([index, name])</code>	Iterate over DataFrame rows as namedtuples, with index value as first element of the tuple.
<code>join(other[, on, how, lsuffix, rsuffix, sort])</code>	Join columns with other DataFrame either on index or on a key column.
<code>keys()</code>	Get the 'info axis' (see Indexing for more)
<code>kurt([axis, skipna, level, numeric_only])</code>	Return unbiased kurtosis over requested axis using Fisher's definition of kurtosis (kurtosis of normal == 0.0).
<code>kurtosis([axis, skipna, level, numeric_only])</code>	Return unbiased kurtosis over requested axis using Fisher's definition of kurtosis (kurtosis of normal == 0.0).
<code>last(offset)</code>	Convenience method for subsetting final periods of time series data based on a date offset.
<code>last_valid_index()</code>	Return label for last non-NA/null value
<code>le(other[, axis, level])</code>	Wrapper for flexible comparison methods le
<code>lookup(row_labels, col_labels)</code>	Label-based "fancy indexing" function for DataFrame.
<code>lt(other[, axis, level])</code>	Wrapper for flexible comparison methods lt
<code>mad([axis, skipna, level])</code>	Return the mean absolute deviation of the values for the requested axis
<code>mask(cond[, other, inplace, axis, level, ...])</code>	Return an object of same shape as self and whose corresponding entries are from self where cond is False and otherwise are from other.
<code>max([axis, skipna, level, numeric_only])</code>	This method returns the maximum of the values in the object.
<code>mean([axis, skipna, level, numeric_only])</code>	Return the mean of the values for the requested axis
<code>median([axis, skipna, level, numeric_only])</code>	Return the median of the values for the requested axis
<code>melt([id_vars, value_vars, var_name, ...])</code>	"Unpivots" a DataFrame from wide format to long format, optionally
<code>memory_usage([index, deep])</code>	Memory usage of DataFrame columns.
<code>merge(right[, how, on, left_on, right_on, ...])</code>	Merge DataFrame objects by performing a database-style join operation by columns or indexes.
<code>min([axis, skipna, level, numeric_only])</code>	This method returns the minimum of the values in the object.
<code>mod(other[, axis, level, fill_value])</code>	Modulo of dataframe and other, element-wise (binary operator <i>mod</i>).
<code>mode([axis, numeric_only])</code>	Gets the mode(s) of each element along the axis selected.
<code>mul(other[, axis, level, fill_value])</code>	Multiplication of dataframe and other, element-wise (binary operator <i>mul</i>).
<code>multiply(other[, axis, level, fill_value])</code>	Multiplication of dataframe and other, element-wise (binary operator <i>mul</i>).
<code>ne(other[, axis, level])</code>	Wrapper for flexible comparison methods ne
<code>nlargest(n, columns[, keep])</code>	Get the rows of a DataFrame sorted by the <i>n</i> largest values of <i>columns</i> .
<code>notnull()</code>	Return a boolean same-sized object indicating if the values are not null.
<code>nsmallest(n, columns[, keep])</code>	Get the rows of a DataFrame sorted by the <i>n</i> smallest values of <i>columns</i> .

[Scroll To Top](#)

<code>nunique([axis, dropna])</code>	Return Series with number of distinct observations over requested axis.
<code>pct_change([periods, fill_method, limit, freq])</code>	Percent change over given number of periods.
<code>pipe(func, *args, **kwargs)</code>	Apply func(self, *args, **kwargs)
<code>pivot([index, columns, values])</code>	Reshape data (produce a “pivot” table) based on column values.
<code>pivot_table(data[, values, index, columns, ...])</code>	Create a spreadsheet-style pivot table as a DataFrame.
<code>plot</code>	alias Of <code>FramePlotMethods</code>
<code>pop(item)</code>	Return item and drop from frame.
<code>pow(other[, axis, level, fill_value])</code>	Exponential power of dataframe and other, element-wise (binary operator <i>pow</i>).
<code>prod([axis, skipna, level, numeric_only])</code>	Return the product of the values for the requested axis
<code>product([axis, skipna, level, numeric_only])</code>	Return the product of the values for the requested axis
<code>quantile([q, axis, numeric_only, interpolation])</code>	Return values at the given quantile over requested axis, a la <code>numpy.percentile</code> .
<code>query(expr[, inplace])</code>	Query the columns of a frame with a boolean expression.
<code>radd(other[, axis, level, fill_value])</code>	Addition of dataframe and other, element-wise (binary operator <i>radd</i>).
<code>rank([axis, method, numeric_only, ...])</code>	Compute numerical data ranks (1 through n) along axis.
<code>rdiv(other[, axis, level, fill_value])</code>	Floating division of dataframe and other, element-wise (binary operator <i>rtruediv</i>).
<code>reindex([index, columns])</code>	Conform DataFrame to new index with optional filling logic, placing NA/NaN in locations having no value in the previous index.
<code>reindex_axis(labels[, axis, method, level, ...])</code>	Conform input object to new index with optional filling logic, placing NA/NaN in locations having no value in the previous index.
<code>reindex_like(other[, method, copy, limit, ...])</code>	Return an object with matching indices to myself.
<code>rename([index, columns])</code>	Alter axes input function or functions.
<code>rename_axis(mapper[, axis, copy, inplace])</code>	Alter index and / or columns using input function or functions.
<code>reorder_levels(order[, axis])</code>	Rearrange index levels using input order.
<code>replace([to_replace, value, inplace, limit, ...])</code>	Replace values given in ‘to_replace’ with ‘value’.
<code>resample(rule[, how, axis, fill_method, ...])</code>	Convenience method for frequency conversion and resampling of time series.
<code>reset_index([level, drop, inplace, ...])</code>	For DataFrame with multi-level index, return new DataFrame with labeling information in the columns under the index names, defaulting to ‘level_0’, ‘level_1’, etc.
<code>rfloordiv(other[, axis, level, fill_value])</code>	Integer division of dataframe and other, element-wise (binary operator <i>rfloordiv</i>).
<code>rmod(other[, axis, level, fill_value])</code>	Modulo of dataframe and other, element-wise (binary operator <i>rmod</i>).
<code>rmul(other[, axis, level, fill_value])</code>	Multiplication of dataframe and other, element-wise (binary operator <i>rmul</i>).
<code>rolling(window[, min_periods, freq, center, ...])</code>	Provides rolling window calculations.
<code>round([decimals])</code>	Round a DataFrame to a variable number of decimal places.

[Scroll To Top](#)

<code>rpow</code> (other[, axis, level, fill_value])	Exponential power of dataframe and other, element-wise (binary operator <i>rpow</i>).
<code>rsub</code> (other[, axis, level, fill_value])	Subtraction of dataframe and other, element-wise (binary operator <i>rsub</i>).
<code>rtruediv</code> (other[, axis, level, fill_value])	Floating division of dataframe and other, element-wise (binary operator <i>rtruediv</i>).
<code>sample</code> ([n, frac, replace, weights, ...])	Returns a random sample of items from an axis of object.
<code>select</code> (crit[, axis])	Return data corresponding to axis labels matching criteria
<code>select_dtypes</code> ([include, exclude])	Return a subset of a DataFrame including/excluding columns based on their dtype.
<code>sem</code> ([axis, skipna, level, ddof, numeric_only])	Return unbiased standard error of the mean over requested axis.
<code>set_axis</code> (axis, labels)	public version of axis assignment
<code>set_index</code> (keys[, drop, append, inplace, ...])	Set the DataFrame index (row labels) using one or more existing columns.
<code>set_value</code> (index, col, value[, takeable])	Put single value at passed column and index
<code>shift</code> ([periods, freq, axis])	Shift index by desired number of periods with an optional time freq
<code>skew</code> ([axis, skipna, level, numeric_only])	Return unbiased skew over requested axis
<code>slice_shift</code> ([periods, axis])	Equivalent to <i>shift</i> without copying data.
<code>sort_index</code> ([axis, level, ascending, ...])	Sort object by labels (along an axis)
<code>sort_values</code> (by[, axis, ascending, inplace, ...])	Sort by the values along either axis
<code>sortlevel</code> ([level, axis, ascending, inplace, ...])	DEPRECATED: use <code>DataFrame.sort_index()</code>
<code>squeeze</code> ([axis])	Squeeze length 1 dimensions.
<code>stack</code> ([level, dropna])	Pivot a level of the (possibly hierarchical) column labels, returning a DataFrame (or Series in the case of an object with a single level of column labels) having a hierarchical index with a new inner-most level of row labels.
<code>std</code> ([axis, skipna, level, ddof, numeric_only])	Return sample standard deviation over requested axis.
<code>sub</code> (other[, axis, level, fill_value])	Subtraction of dataframe and other, element-wise (binary operator <i>sub</i>).
<code>subtract</code> (other[, axis, level, fill_value])	Subtraction of dataframe and other, element-wise (binary operator <i>sub</i>).
<code>sum</code> ([axis, skipna, level, numeric_only])	Return the sum of the values for the requested axis
<code>swapaxes</code> (axis1, axis2[, copy])	Interchange axes and swap values axes appropriately
<code>swaplevel</code> ([i, j, axis])	Swap levels i and j in a MultiIndex on a particular axis
<code>tail</code> ([n])	Returns last n rows
<code>take</code> (indices[, axis, convert, is_copy])	Analogous to ndarray.take
<code>to_clipboard</code> ([excel, sep])	Attempt to write text representation of object to the system clipboard This can be pasted into Excel, for example.
<code>to_csv</code> ([path_or_buf, sep, na_rep, ...])	Write DataFrame to a comma-separated values (csv) file
<code>to_dense</code> ()	Return dense representation of Scroll To Top DataFrame (opposed to sparse)
<code>to_dict</code> ([orient])	Convert DataFrame to dictionary.
<code>to_excel</code> (excel_writer[, sheet_name, na_rep, ...])	Write DataFrame to an excel sheet

<code>to_feather(fname)</code>	write out the binary feather-format for DataFrames
<code>to_gbq(destination_table, project_id[, ...])</code>	Write a DataFrame to a Google BigQuery table.
<code>to_hdf(path_or_buf, key, **kwargs)</code>	Write the contained data to an HDF5 file using HDFStore.
<code>to_html([buf, columns, col_space, header, ...])</code>	Render a DataFrame as an HTML table.
<code>to_json([path_or_buf, orient, date_format, ...])</code>	Convert the object to a JSON string.
<code>to_latex([buf, columns, col_space, header, ...])</code>	Render an object to a tabular environment table.
<code>to_msgpack([path_or_buf, encoding])</code>	msgpack (serialize) object to input file path
<code>to_panel()</code>	Transform long (stacked) format (DataFrame) into wide (3D, Panel) format.
<code>to_period([freq, axis, copy])</code>	Convert DataFrame from DatetimeIndex to PeriodIndex with desired
<code>to_pickle(path[, compression])</code>	Pickle (serialize) object to input file path.
<code>to_records([index, convert_datetime64])</code>	Convert DataFrame to record array.
<code>to_sparse([fill_value, kind])</code>	Convert to SparseDataFrame
<code>to_sql(name, con[, flavor, schema, ...])</code>	Write records stored in a DataFrame to a SQL database.
<code>to_stata(fname[, convert_dates, ...])</code>	A class for writing Stata binary dta files from array-like objects
<code>to_string([buf, columns, col_space, header, ...])</code>	Render a DataFrame to a console-friendly tabular output.
<code>to_timestamp([freq, how, axis, copy])</code>	Cast to DatetimeIndex of timestamps, at <i>beginning</i> of period
<code>to_xarray()</code>	Return an xarray object from the pandas object.
<code>transform(func, *args, **kwargs)</code>	Call function producing a like-indexed NDFrame
<code>transpose(*args, **kwargs)</code>	Transpose index and columns
<code>truediv(other[, axis, level, fill_value])</code>	Floating division of dataframe and other, element-wise (binary operator <i>truediv</i>).
<code>truncate([before, after, axis, copy])</code>	Truncates a sorted NDFrame before and/or after some particular index value.
<code>tshift([periods, freq, axis])</code>	Shift the time index, using the index's frequency if available.
<code>tz_convert(tz[, axis, level, copy])</code>	Convert tz-aware axis to target time zone.
<code>tz_localize(tz[, axis, level, copy, ambiguous])</code>	Localize tz-naive TimeSeries to target time zone.
<code>unstack([level, fill_value])</code>	Pivot a level of the (necessarily hierarchical) index labels, returning a DataFrame having a new level of column labels whose inner-most level consists of the pivoted index labels.
<code>update(other[, join, overwrite, ...])</code>	Modify DataFrame in place using non-NA values from passed DataFrame.
<code>var([axis, skipna, level, ddof, numeric_only])</code>	Return unbiased variance over requested axis.
<code>where(cond[, other, inplace, axis, level, ...])</code>	Return an object of same shape as self and whose corresponding entries are from self where cond is True and otherwise are from other.
<code>xs(key[, axis, level, drop_level])</code>	Returns a cross-section (row(s) or column(s)) from the Series/DataFrame.

[Scroll To Top](#)