pandas.DataFrame

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

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

Parameters:

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

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

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 as_blocks()
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 *add*).
add_prefix(prefix)	Concatenate prefix string with panel items names.
add_suffix(suffix)	Concatenate suffix string with panel items names.
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 *other* 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 all TimeSeries inside to specified frequency using DateOffset objects.
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, raise_on_error])	Cast object to input numpy.dtype
at_time(time[, asof])	Select values at particular time of day (e.g.
between_time(start_time, end_time[, ...])	Select values between particular times of the day (e.g., 9:00-9:30 AM).
bfill([axis, inplace, limit, downcast])	Synonym for NDFrame.fillna(method='bfill')
bool()	Return the bool of a single element PandasObject.
boxplot([column, by, ax, fontsize, rot, ...])	Make a box plot from DataFrame column optionally grouped by some columns or

clip([lower, upper, axis])	Trim values at input threshold(s).
clip_lower(threshold[, axis])	Return copy of the input with values below given value(s) truncated.
clip_upper(threshold[, axis])	Return copy of input with values above given value(s) truncated.
combine(other, func[, fill_value, overwrite])	Add two DataFrame objects and do not propagate NaN values, so if for a
combineAdd(other)	DEPRECATED.
combineMult(other)	DEPRECATED.
combine_first(other)	Combine two DataFrame objects and default to non-null values in frame calling the method.
compound([axis, skipna, level])	Return the compound percentage of the values for the requested axis
consolidate([inplace])	Compute NDFrame with “consolidated” internals (data of each dtype grouped together in a single ndarray).
convert_objects([convert_dates, ...])	Deprecated.
copy([deep])	Make a copy of this objects data.
corr([method, min_periods])	Compute pairwise correlation of columns, excluding NA/null values
corrwith(other[, axis, drop])	Compute pairwise correlation between rows or columns of two DataFrame objects.
count([axis, level, numeric_only])	Return Series with number of non-NA/null observations over requested axis.
cov([min_periods])	Compute pairwise covariance of columns, excluding NA/null values
cummax([axis, dtype, out, skipna])	Return cumulative cummax over requested axis.
cummin([axis, dtype, out, skipna])	Return cumulative cummin over requested axis.
cumprod([axis, dtype, out, skipna])	Return cumulative cumprod over requested axis.
cumsum([axis, dtype, out, skipna])	Return cumulative cumsum over requested axis.
describe([percentiles, include, exclude])	Generate various summary statistics, excluding NaN values.

diff([periods, axis])	1st discrete difference of object
div(other[, axis, level, fill_value])	Floating division of dataframe and other, element-wise (binary operator *truediv*).
divide(other[, axis, level, fill_value])	Floating division of dataframe and other, element-wise (binary operator *truediv*).
dot(other)	Matrix multiplication with DataFrame or Series objects
drop(labels[, axis, level, inplace, errors])	Return new object with labels in requested axis removed.
drop_duplicates(*args, **kwargs)	Return DataFrame with duplicate rows removed, optionally only
dropna([axis, how, thresh, subset, inplace])	Return object with labels on given axis omitted where alternately any
duplicated(*args, **kwargs)	Return boolean Series denoting duplicate rows, optionally only
eq(other[, axis, level])	Wrapper for flexible comparison methods eq
equals(other)	Determines if two NDFrame objects contain the same elements.
eval(expr[, inplace])	Evaluate an expression in the context of the calling DataFrame instance.
ewm([com, span, halflife, alpha, ...])	Provides exponential weighted functions
expanding([min_periods, freq, center, axis])	Provides expanding transformations.
ffill([axis, inplace, limit, downcast])	Synonym for NDFrame.fillna(method='ffill')
fillna([value, method, axis, inplace, ...])	Fill NA/NaN values using the specified method
filter([items, like, regex, axis])	Restrict the info axis to set of items or wildcard
first(offset)	Convenience method for subsetting initial periods of time series data based on a date offset.
first_valid_index()	Return label for first non-NA/null value
floordiv(other[, axis, level, fill_value])	Integer division of dataframe and other, element-wise (binary operator *floordiv*).
from_csv(path[, header, sep, index_col, ...])	Read CSV file (DISCOURAGED, please use **pandas.read_csv()** instead).
from_dict(data[, orient, dtype])	Construct DataFrame from dict of array-like or dicts

from_items(items[, columns, orient])	Convert (key, value) pairs to DataFrame.
from_records(data[, index, exclude, ...])	Convert structured or record ndarray to DataFrame
ge(other[, axis, level])	Wrapper for flexible comparison methods ge
get(key[, default])	Get item from object for given key (DataFrame column, Panel slice, etc.).
get_dtype_counts()	Return the counts of dtypes in this object.
get_ftype_counts()	Return the counts of ftypes in this object.
get_value(index, col[, takeable])	Quickly retrieve single value at passed column and index
get_values()	same as values (but handles sparseness conversions)
groupby([by, axis, level, as_index, sort, ...])	Group series using mapper (dict or key function, apply given function to group, return result as series) or by a series of columns.
gt(other[, axis, level])	Wrapper for flexible comparison methods gt
head([n])	Returns first n rows
hist(data[, column, by, grid, xlabelsize, ...])	Draw histogram of the DataFrame's series using matplotlib / pylab.
icol(i)	DEPRECATED.
idxmax([axis, skipna])	Return index of first occurrence of maximum over requested axis.
idxmin([axis, skipna])	Return index of first occurrence of minimum over requested axis.
iget_value(i, j)	DEPRECATED.
info([verbose, buf, max_cols, memory_usage, ...])	Concise summary of a DataFrame.
insert(loc, column, value[, allow_duplicates])	Insert column into DataFrame at specified location.
interpolate([method, axis, limit, inplace, ...])	Interpolate values according to different methods.
irow(i[, copy])	DEPRECATED.
isin(values)	Return boolean DataFrame showing whether each element in the DataFrame is contained in values.
isnull()	Return a boolean same-sized object indicating if the values are null.

iteritems()	Iterator over (column name, Series) pairs.
iterkv(*args, **kwargs)	iteritems alias used to get around 2to3. Deprecated
iterrows()	Iterate over DataFrame rows as (index, Series) pairs.
itertuples([index, name])	Iterate over DataFrame rows as namedtuples, with index value as first element of the tuple.
join(other[, on, how, lsuffix, rsuffix, sort])	Join columns with other DataFrame either on index or on a key column.
keys()	Get the ‘info axis’ (see Indexing for more)
kurt([axis, skipna, level, numeric_only])	Return unbiased kurtosis over requested axis using Fisher’s definition of kurtosis (kurtosis of normal == 0.0).
kurtosis([axis, skipna, level, numeric_only])	Return unbiased kurtosis over requested axis using Fisher’s definition of kurtosis (kurtosis of normal == 0.0).
last(offset)	Convenience method for subsetting final periods of time series data based on a date offset.
last_valid_index()	Return label for last non-NA/null value
le(other[, axis, level])	Wrapper for flexible comparison methods le
lookup(row_labels, col_labels)	Label-based “fancy indexing” function for DataFrame.
lt(other[, axis, level])	Wrapper for flexible comparison methods lt
mad([axis, skipna, level])	Return the mean absolute deviation of the values for the requested axis
mask(cond[, other, inplace, axis, level, ...])	Return an object of same shape as self and whose corresponding entries are from self where cond is False and otherwise are from other.
max([axis, skipna, level, numeric_only])	This method returns the maximum of the values in the object.
mean([axis, skipna, level, numeric_only])	Return the mean of the values for the requested axis
median([axis, skipna, level, numeric_only])	Return the median of the values for the requested axis
memory_usage([index, deep])	Memory usage of DataFrame columns.
merge(right[, how, on, left_on, right_on, ...])	Merge DataFrame objects by performing a database-style join operation by columns or indexes.

min([axis, skipna, level, numeric_only])	This method returns the minimum of the values in the object.
mod(other[, axis, level, fill_value])	Modulo of dataframe and other, element-wise (binary operator *mod*).
mode([axis, numeric_only])	Gets the mode(s) of each element along the axis selected.
mul(other[, axis, level, fill_value])	Multiplication of dataframe and other, element-wise (binary operator *mul*).
multiply(other[, axis, level, fill_value])	Multiplication of dataframe and other, element-wise (binary operator *mul*).
ne(other[, axis, level])	Wrapper for flexible comparison methods ne
nlargest(n, columns[, keep])	Get the rows of a DataFrame sorted by the *n* largest values of *columns*.
notnull()	Return a boolean same-sized object indicating if the values are not null.
nsmallest(n, columns[, keep])	Get the rows of a DataFrame sorted by the *n* smallest values of *columns*.
pct_change([periods, fill_method, limit, freq])	Percent change over given number of periods.
pipe(func, *args, **kwargs)	Apply func(self, *args, **kwargs)
pivot([index, columns, values])	Reshape data (produce a “pivot” table) based on column values.
pivot_table(data[, values, index, columns, ...])	Create a spreadsheet-style pivot table as a DataFrame.
plot	alias of **FramePlotMethods**
pop(item)	Return item and drop from frame.
pow(other[, axis, level, fill_value])	Exponential power of dataframe and other, element-wise (binary operator *pow*).
prod([axis, skipna, level, numeric_only])	Return the product of the values for the requested axis
product([axis, skipna, level, numeric_only])	Return the product of the values for the requested axis
quantile([q, axis, numeric_only, interpolation])	Return values at the given quantile over requested axis, a la numpy.percentile.
query(expr[, inplace])	Query the columns of a frame with a boolean expression.

radd(other[, axis, level, fill_value])	Addition of dataframe and other, element-wise (binary operator *radd*).
rank([axis, method, numeric_only, ...])	Compute numerical data ranks (1 through n) along axis.
rdiv(other[, axis, level, fill_value])	Floating division of dataframe and other, element-wise (binary operator *rtruediv*).
reindex([index, columns])	Conform DataFrame to new index with optional filling logic, placing NA/NaN in locations having no value in the previous index.
reindex_axis(labels[, axis, method, level, ...])	Conform input object to new index with optional filling logic, placing NA/NaN in locations having no value in the previous index.
reindex_like(other[, method, copy, limit, ...])	Return an object with matching indices to myself.
rename([index, columns])	Alter axes input function or functions.
rename_axis(mapper[, axis, copy, inplace])	Alter index and / or columns using input function or functions.
reorder_levels(order[, axis])	Rearrange index levels using input order.
replace([to_replace, value, inplace, limit, ...])	Replace values given in ‘to_replace’ with ‘value’.
resample(rule[, how, axis, fill_method, ...])	Convenience method for frequency conversion and resampling of regular time-series data.
reset_index([level, drop, inplace, ...])	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.
rfloordiv(other[, axis, level, fill_value])	Integer division of dataframe and other, element-wise (binary operator *rfloordiv*).
rmod(other[, axis, level, fill_value])	Modulo of dataframe and other, element-wise (binary operator *rmod*).
rmul(other[, axis, level, fill_value])	Multiplication of dataframe and other, element-wise (binary operator *rmul*).
rolling(window[, min_periods, freq, center, ...])	Provides rolling transformations.
round([decimals])	Round a DataFrame to a variable number of decimal

	places.
rpow(other[, axis, level, fill_value])	Exponential power of dataframe and other, element-wise (binary operator *rpow*).
rsub(other[, axis, level, fill_value])	Subtraction of dataframe and other, element-wise (binary operator *rsub*).
rtruediv(other[, axis, level, fill_value])	Floating division of dataframe and other, element-wise (binary operator *rtruediv*).
sample([n, frac, replace, weights, ...])	Returns a random sample of items from an axis of object.
select(crit[, axis])	Return data corresponding to axis labels matching criteria
select_dtypes([include, exclude])	Return a subset of a DataFrame including/excluding columns based on their `dtype`.
sem([axis, skipna, level, ddof, numeric_only])	Return unbiased standard error of the mean over requested axis.
set_axis(axis, labels)	public verson of axis assignment
set_index(keys[, drop, append, inplace, ...])	Set the DataFrame index (row labels) using one or more existing columns.
set_value(index, col, value[, takeable])	Put single value at passed column and index
shift([periods, freq, axis])	Shift index by desired number of periods with an optional time freq
skew([axis, skipna, level, numeric_only])	Return unbiased skew over requested axis
slice_shift([periods, axis])	Equivalent to *shift* without copying data.
sort([columns, axis, ascending, inplace, ...])	DEPRECATED: use **`DataFrame.sort_values()`**
sort_index([axis, level, ascending, ...])	Sort object by labels (along an axis)
sort_values(by[, axis, ascending, inplace, ...])	Sort by the values along either axis
sortlevel([level, axis, ascending, inplace, ...])	Sort multilevel index by chosen axis and primary level.
squeeze(**kwargs)	Squeeze length 1 dimensions.
stack([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.
std([axis, skipna, level, ddof, numeric_only])	Return sample standard deviation over requested axis.
sub(other[, axis, level, fill_value])	Subtraction of dataframe and other, element-wise (binary operator *sub*).
subtract(other[, axis, level, fill_value])	Subtraction of dataframe and other, element-wise (binary operator *sub*).
sum([axis, skipna, level, numeric_only])	Return the sum of the values for the requested axis
swapaxes(axis1, axis2[, copy])	Interchange axes and swap values axes appropriately
swaplevel([i, j, axis])	Swap levels i and j in a MultiIndex on a particular axis
tail([n])	Returns last n rows
take(indices[, axis, convert, is_copy])	Analogous to ndarray.take
to_clipboard([excel, sep])	Attempt to write text representation of object to the system clipboard This can be pasted into Excel, for example.
to_csv([path_or_buf, sep, na_rep, ...])	Write DataFrame to a comma-separated values (csv) file
to_dense()	Return dense representation of NDFrame (as opposed to sparse)
to_dict(*args, **kwargs)	Convert DataFrame to dictionary.
to_excel(excel_writer[, sheet_name, na_rep, ...])	Write DataFrame to a excel sheet
to_gbq(destination_table, project_id[, ...])	Write a DataFrame to a Google BigQuery table.
to_hdf(path_or_buf, key, **kwargs)	Activate the HDFStore.
to_html([buf, columns, col_space, colSpace, ...])	Render a DataFrame as an HTML table.
to_json([path_or_buf, orient, date_format, ...])	Convert the object to a JSON string.
to_latex([buf, columns, col_space, ...])	Render a DataFrame to a tabular environment table.
to_msgpack([path_or_buf, encoding])	msgpack (serialize) object to input file path
to_panel()	Transform long (stacked) format (DataFrame) into wide (3D, Panel) format.
to_period([freq, axis, copy])	Convert DataFrame from DatetimeIndex to PeriodIndex with desired
to_pickle(path)	Pickle (serialize) object to input file path.

to_records([index, convert_datetime64])	Convert DataFrame to record array.
to_sparse([fill_value, kind])	Convert to SparseDataFrame
to_sql(name, con[, flavor, schema, ...])	Write records stored in a DataFrame to a SQL database.
to_stata(fname[, convert_dates, ...])	A class for writing Stata binary dta files from array-like objects
to_string([buf, columns, col_space, header, ...])	Render a DataFrame to a console-friendly tabular output.
to_timestamp([freq, how, axis, copy])	Cast to DatetimeIndex of timestamps, at *beginning* of period
to_wide(*args, **kwargs)	
to_xarray()	Return an xarray object from the pandas object.
transpose(*args, **kwargs)	Transpose index and columns
truediv(other[, axis, level, fill_value])	Floating division of dataframe and other, element-wise (binary operator *truediv*).
truncate([before, after, axis, copy])	Truncates a sorted NDFrame before and/or after some particular dates.
tshift([periods, freq, axis])	Shift the time index, using the index's frequency if available.
tz_convert(tz[, axis, level, copy])	Convert tz-aware axis to target time zone.
tz_localize(*args, **kwargs)	Localize tz-naive TimeSeries to target time zone.
unstack([level, fill_value])	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.
update(other[, join, overwrite, ...])	Modify DataFrame in place using non-NA values from passed DataFrame.
var([axis, skipna, level, ddof, numeric_only])	Return unbiased variance over requested axis.
where(cond[, other, inplace, axis, level, ...])	Return an object of same shape as self and whose corresponding entries are from self where cond is True and otherwise are from other.
xs(key[, axis, level, copy, drop_level])	Returns a cross-section (row(s) or column(s)) from the

Series/DataFrame.