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.

Parameters:

data : *ndarray (structured or homogeneous), Iterable, dict, or DataFrame*
Dict can contain Series, arrays, constants, or list-like objects
Changed in version 0.23.0: If data is a dict, argument order is maintained for Python 3.6 and later.

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. Will default to RangeIndex (0, 1, 2, …, n) if no column labels are provided

dtype : *dtype, default None*
Data type to force. Only a single dtype is allowed. If None, 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

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
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 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
```

Attributes

T	Transpose index and columns.
at	Access a single value for a row/column label pair.
axes	Return a list representing the axes of the DataFrame.
blocks	(DEPRECATED) Internal property, property synonym for as_blocks().
columns	The column labels of the DataFrame.
dtypes	Return the dtypes in the DataFrame.
empty	Indicator whether DataFrame is empty.
ftypes	Return the ftypes (indication of sparse/dense and dtype) in DataFrame.
iat	Access a single value for a row/column pair by integer position.
iloc	Purely integer-location based indexing for selection by position.
index	The index (row labels) of the DataFrame.
is_copy	Return the copy.
ix	A primarily label-location based indexer, with integer position fallback.
loc	Access a group of rows and columns by label(s) or a boolean array.
ndim	Return an int representing the number of axes / array dimensions.
shape	Return a tuple representing the dimensionality of the DataFrame.
size	Return an int representing the number of elements in this object.
style	Property returning a Styler object containing methods for building a styled HTML representation fo the DataFrame.
values	Return a Numpy representation of the DataFrame.

timetuple

Methods

abs()	Return a Series/DataFrame with absolute numeric value of each element.
add(other[, axis, level, fill_value])	Addition of dataframe and other, element-wise (binary operator *add*).
add_prefix(prefix)	Prefix labels with string *prefix*.
add_suffix(suffix)	Suffix labels with string *suffix*.
agg(func[, axis])	Aggregate using one or more operations over the specified axis.
aggregate(func[, axis])	Aggregate using one or more operations over the specified axis.
align(other[, join, axis, level, copy, …])	Align two objects on their axes with the specified join method for each axis Index.
all([axis, bool_only, skipna, level])	Return whether all elements are True, potentially over an axis.
any([axis, bool_only, skipna, level])	Return whether any element is True, potentially over an axis.
append(other[, ignore_index, …])	Append rows of *other* to the end of caller, returning a new object.
apply(func[, axis, broadcast, raw, reduce, …])	Apply a function along an axis of the DataFrame.
applymap(func)	Apply a function to a Dataframe elementwise.
as_blocks([copy])	(DEPRECATED) Convert the frame to a dict of dtype -> Constructor Types that each has a homogeneous dtype.
as_matrix([columns])	(DEPRECATED) Convert the frame to its Numpy-array representation.
asfreq(freq[, method, how, normalize, …])	Convert TimeSeries to specified frequency.
asof(where[, subset])	Return the last row(s) without any NaNs before *where*.
assign(**kwargs)	Assign new columns to a DataFrame.
astype(dtype[, copy, errors])	Cast a pandas object to a specified dtype `dtype`.
at_time(time[, asof, axis])	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 **DataFrame.fillna()** with `method='bfill'`.
bool()	Return the bool of a single element PandasObject.
boxplot([column, by, ax, fontsize, rot, …])	Make a box plot from DataFrame columns.
clip([lower, upper, axis, inplace])	Trim values at input threshold(s).
clip_lower(threshold[, axis, inplace])	(DEPRECATED) Trim values below a given threshold.
clip_upper(threshold[, axis, inplace])	(DEPRECATED) Trim values above a given threshold.
combine(other, func[, fill_value, overwrite])	Perform column-wise combine with another DataFrame based on a passed function.
combine_first(other)	Update null elements with value in the same location in *other*.
compound([axis, skipna, level])	Return the compound percentage of the values for the requested axis.
convert_objects([convert_dates, …])	(DEPRECATED) Attempt to infer better dtype for object columns.

copy([deep])	Make a copy of this object's indices and data.
corr([method, min_periods])	Compute pairwise correlation of columns, excluding NA/null values.
corrwith(other[, axis, drop, method])	Compute pairwise correlation between rows or columns of DataFrame with rows or columns of Series or DataFrame.
count([axis, level, numeric_only])	Count non-NA cells for each column or row.
cov([min_periods])	Compute pairwise covariance of columns, excluding NA/null values.
cummax([axis, skipna])	Return cumulative maximum over a DataFrame or Series axis.
cummin([axis, skipna])	Return cumulative minimum over a DataFrame or Series axis.
cumprod([axis, skipna])	Return cumulative product over a DataFrame or Series axis.
cumsum([axis, skipna])	Return cumulative sum over a DataFrame or Series axis.
describe([percentiles, include, exclude])	Generate descriptive statistics that summarize the central tendency, dispersion and shape of a dataset's distribution, excluding `NaN` values.
diff([periods, axis])	First discrete difference of element.
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)	Compute the matrix mutiplication between the DataFrame and other.
drop([labels, axis, index, columns, level, …])	Drop specified labels from rows or columns.
drop_duplicates([subset, keep, inplace])	Return DataFrame with duplicate rows removed, optionally only considering certain columns.
droplevel(level[, axis])	Return DataFrame with requested index / column level(s) removed.
dropna([axis, how, thresh, subset, inplace])	Remove missing values.
duplicated([subset, keep])	Return boolean Series denoting duplicate rows, optionally only considering certain columns.
eq(other[, axis, level])	Equal to of dataframe and other, element-wise (binary operator *eq*).
equals(other)	Test whether two objects contain the same elements.
eval(expr[, inplace])	Evaluate a string describing operations on DataFrame columns.
ewm([com, span, halflife, alpha, …])	Provides exponential weighted functions.
expanding([min_periods, center, axis])	Provides expanding transformations.
ffill([axis, inplace, limit, downcast])	Synonym for **DataFrame.fillna()** with `method='ffill'`.
fillna([value, method, axis, inplace, …])	Fill NA/NaN values using the specified method.
filter([items, like, regex, axis])	Subset rows or columns of dataframe according to labels in the specified index.
first(offset)	Convenience method for subsetting initial periods of time series data based on a date offset.
first_valid_index()	Return index 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, …])	(DEPRECATED) Read CSV file.
from_dict(data[, orient, dtype, columns])	Construct DataFrame from dict of array-like or dicts.
from_items(items[, columns, orient])	(DEPRECATED) Construct a DataFrame from a list of tuples.
from_records(data[, index, exclude, …])	Convert structured or record ndarray to DataFrame.
ge(other[, axis, level])	Greater than or equal to of dataframe and other, element-wise (binary operator *ge*).
get(key[, default])	Get item from object for given key (DataFrame column, Panel slice, etc.).
get_dtype_counts()	Return counts of unique dtypes in this object.
get_ftype_counts()	(DEPRECATED) Return counts of unique ftypes in this object.
get_value(index, col[, takeable])	(DEPRECATED) Quickly retrieve single value at passed column and index.
get_values()	Return an ndarray after converting sparse values to dense.
groupby([by, axis, level, as_index, sort, …])	Group DataFrame or Series using a mapper or by a Series of columns.
gt(other[, axis, level])	Greater than of dataframe and other, element-wise (binary operator *gt*).
head([n])	Return the first *n* rows.
hist([column, by, grid, xlabelsize, xrot, …])	Make a histogram of the DataFrame's.
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.
infer_objects()	Attempt to infer better dtypes for object columns.
info([verbose, buf, max_cols, memory_usage, …])	Print a 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.
isin(values)	Whether each element in the DataFrame is contained in values.
isna()	Detect missing values.
isnull()	Detect missing values.
items()	Iterator over (column name, Series) pairs.
iteritems()	Iterator over (column name, Series) pairs.
iterrows()	Iterate over DataFrame rows as (index, Series) pairs.
itertuples([index, name])	Iterate over DataFrame rows as namedtuples.
join(other[, on, how, lsuffix, rsuffix, sort])	Join columns of another DataFrame.
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 index for last non-NA/null value.

le(other[, axis, level])	Less than or equal to of dataframe and other, element-wise (binary operator *le*).
lookup(row_labels, col_labels)	Label-based “fancy indexing” function for DataFrame.
lt(other[, axis, level])	Less than of dataframe and other, element-wise (binary operator *lt*).
mad([axis, skipna, level])	Return the mean absolute deviation of the values for the requested axis.
mask(cond[, other, inplace, axis, level, …])	Replace values where the condition is True.
max([axis, skipna, level, numeric_only])	Return the maximum of the values for the requested axis.
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.
melt([id_vars, value_vars, var_name, …])	Unpivots a DataFrame from wide format to long format, optionally leaving identifier variables set.
memory_usage([index, deep])	Return the memory usage of each column in bytes.
merge(right[, how, on, left_on, right_on, …])	Merge DataFrame or named Series objects with a database-style join.
min([axis, skipna, level, numeric_only])	Return the minimum of the values for the requested axis.
mod(other[, axis, level, fill_value])	Modulo of dataframe and other, element-wise (binary operator *mod*).
mode([axis, numeric_only, dropna])	Get the mode(s) of each element along the selected axis.
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])	Not equal to of dataframe and other, element-wise (binary operator *ne*).
nlargest(n, columns[, keep])	Return the first *n* rows ordered by *columns* in descending order.
notna()	Detect existing (non-missing) values.
notnull()	Detect existing (non-missing) values.
nsmallest(n, columns[, keep])	Return the first *n* rows ordered by *columns* in ascending order.
nunique([axis, dropna])	Count distinct observations over requested axis.
pct_change([periods, fill_method, limit, freq])	Percentage change between the current and a prior element.
pipe(func, *args, **kwargs)	Apply func(self, *args, **kwargs).
pivot([index, columns, values])	Return reshaped DataFrame organized by given index / column values.
pivot_table([values, index, columns, …])	Create a spreadsheet-style pivot table as a DataFrame.
plot	alias of **pandas.plotting._core.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 re-

	quested 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.
query(expr[, inplace])	Query the columns of a DataFrame 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([labels, index, columns, axis, …])	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, …])	(DEPRECATED) Conform input object to new index.
reindex_like(other[, method, copy, limit, …])	Return an object with matching indices as other object.
rename([mapper, index, columns, axis, copy, …])	Alter axes labels.
rename_axis([mapper, index, columns, axis, …])	Set the name of the axis for the index or columns.
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, …])	Resample time-series data.
reset_index([level, drop, inplace, …])	Reset the index, or a level of it.
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, center, …])	Provides rolling window calculations.
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, …])	Return a random sample of items from an axis of object.
select(crit[, axis])	(DEPRECATED) Return data corresponding to axis labels matching criteria.
select_dtypes([include, exclude])	Return a subset of the DataFrame's columns based on the column dtypes.
sem([axis, skipna, level, ddof, numeric_only])	Return unbiased standard error of the mean over requested axis.
set_axis(labels[, axis, inplace])	Assign desired index to given axis.
set_index(keys[, drop, append, inplace, …])	Set the DataFrame index using existing columns.
set_value(index, col, value[, takeable])	(DEPRECATED) Put single value at passed column and index.

shift([periods, freq, axis, fill_value])	Shift index by desired number of periods with an optional time *freq*.
skew([axis, skipna, level, numeric_only])	Return unbiased skew over requested axis Normalized by N-1.
slice_shift([periods, axis])	Equivalent to *shift* without copying data.
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
squeeze([axis])	Squeeze 1 dimensional axis objects into scalars.
stack([level, dropna])	Stack the prescribed level(s) from columns to index.
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])	Return the last *n* rows.
take(indices[, axis, convert, is_copy])	Return the elements in the given *positional* indices along an axis.
to_clipboard([excel, sep])	Copy object to the system clipboard.
to_csv([path_or_buf, sep, na_rep, …])	Write object to a comma-separated values (csv) file.
to_dense()	Return dense representation of NDFrame (as opposed to sparse).
to_dict([orient, into])	Convert the DataFrame to a dictionary.
to_excel(excel_writer[, sheet_name, na_rep, …])	Write object to an Excel sheet.
to_feather(fname)	Write out the binary feather-format for DataFrames.
to_gbq(destination_table[, project_id, …])	Write a DataFrame to a Google BigQuery table.
to_hdf(path_or_buf, key, **kwargs)	Write the contained data to an HDF5 file using HDFStore.
to_html([buf, columns, col_space, header, …])	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, header, …])	Render an object to a LaTeX tabular environment table.
to_msgpack([path_or_buf, encoding])	Serialize object to input file path using msgpack format.
to_numpy([dtype, copy])	Convert the DataFrame to a NumPy array.
to_panel()	(DEPRECATED) Transform long (stacked) format (DataFrame) into wide (3D, Panel) format.
to_parquet(fname[, engine, compression, …])	Write a DataFrame to the binary parquet format.
to_period([freq, axis, copy])	Convert DataFrame from DatetimeIndex to PeriodIndex with desired frequency (inferred from index if not passed).
to_pickle(path[, compression, protocol])	Pickle (serialize) object to file.
to_records([index, convert_datetime64, …])	Convert DataFrame to a NumPy record array.

to_sparse([fill_value, kind])	Convert to SparseDataFrame.
to_sql(name, con[, schema, if_exists, …])	Write records stored in a DataFrame to a SQL database.
to_stata(fname[, convert_dates, …])	Export DataFrame object to Stata dta format.
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_xarray()	Return an xarray object from the pandas object.
transform(func[, axis])	Call `func` on self producing a DataFrame with transformed values and that has the same axis length as self.
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])	Truncate a Series or DataFrame before and after some index value.
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(tz[, axis, level, copy, …])	Localize tz-naive index of a Series or DataFrame 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 in place using non-NA values from another DataFrame.
var([axis, skipna, level, ddof, numeric_only])	Return unbiased variance over requested axis.
where(cond[, other, inplace, axis, level, …])	Replace values where the condition is False.
xs(key[, axis, level, drop_level])	Return cross-section from the Series/DataFrame.