Built-in Functions

*class***bool**([*x*]) Return a Boolean value, i.e. one of True or False. *x* is converted using the standard [truth testing procedure](https://docs.python.org/3/library/stdtypes.html#truth).

*class***bytearray**([*src*[, *enc*[, *err*]]]) Return a new array of bytes, which is an mutable sequence of ints in range 0 <= x < 256.

*class***bytes**([*src*[, *enc*[, *err*]]]) Return a new “bytes” object, which is an immutable sequence of ints in range 0 <= x < 256.

*class***complex**([*real*[, *imag*]]) Return a complex number with the value *real* + *imag*\*1j or convert a string/number to complex number.

*class***float**([*x*]) Return a floating point number constructed from a number or string *x*.

*class***frozenset**([*iterable*]) Return a new [frozenset](https://docs.python.org/3/library/stdtypes.html#frozenset) object, optionally with elements taken from *iterable*. frozenset is a built-in class.

*class***int**(*x*, *base=10*) Return an integer object constructed from a number or string *x*, or return 0 if no arguments are given.

*class***list**([*iterable*]) Rather than being a function, [list](https://docs.python.org/3/library/stdtypes.html#list) is actually a mutable sequence type.

*class***object** Return featureless object. [object](https://docs.python.org/3/library/functions.html#object) is a base for all classes. It has methods common to all instances of Python classes.

*class*set([*iterable*]) Return a new [set](https://docs.python.org/3/library/stdtypes.html#set) object, optionally with elements taken from *iterable*. set is a built-in class. *class***str**(*object=''*)

*class***str**(*object=b''*, *enc*, *err*) Return a [str](https://docs.python.org/3/library/stdtypes.html#str) version of *object*.

*class***tuple**([*iterable*]) Rather than being a function, [tuple](https://docs.python.org/3/library/stdtypes.html#tuple) is actually an immutable sequence type

*class***type**(*name*, *b*, *dict*, *\*\*kwds*) With one argument, return type of an *object*. Return value is a type object and generally = [object.\_\_class\_\_](https://docs.python.org/3/library/stdtypes.html#instance.__class__).

Iterables:

**all**(*iterable*) Return True if all elements of the *iterable* are true (or if the iterable is empty).

**any**(*iterable*) Return True if any element of the *iterable* is true. If the iterable is empty, return False.

**iter**(*object*[, *sentinel*]) Return an [iterator](https://docs.python.org/3/glossary.html#term-iterator) object. The first argument is interpreted very differently depending on presence of 2nd argument. Without a 2nd argument, *object* must be a collection ([\_\_iter\_\_()](https://docs.python.org/3/reference/datamodel.html#object.__iter__) ), or sequence ([\_\_getitem\_\_()](https://docs.python.org/3/reference/datamodel.html#object.__getitem__)).

**next**(*iter*[, *def*]) Retrieve the next item from *iterator* by calling its [\_\_next\_\_()](https://docs.python.org/3/library/stdtypes.html#iterator.__next__) method. If *default* given, it is returned if the iterator is exhausted.

**zip**(*\*iterables*) Make an iterator that aggregates elements from each of the iterables. Returns an iterator of tuples. Iterator stops when the shortest input iterable is exhausted. With a single iterable argument, it returns an iterator of 1-tuples.

**sorted**(*iterable*, *\**, *key*, *reverse*) Return a new sorted list from the items in *iterable*.

**reversed**(*seq*) Return a reverse [iterator](https://docs.python.org/3/glossary.html#term-iterator).

**map**(*func*, *iterable*, *...*) return an iterator that applies *func* to every item of *iterable*. If multiple iterables - stops when shortest is exhausted.

**filter**(*func*, *iterable*) construct an iterator from those elements of *iterable* for which *function* returns true.

**reduce**(*func, iterable[, i*]) reduce applies a function of two arguments cumulatively to the elements of an *iterable* through all elements

Objects/Classes:

*class***slice**(*start*, *stop*[, *step*]) Return a [slice](https://docs.python.org/3/glossary.html#term-slice) object representing the set of indices specified by range(start, stop, step).

**getattr**(*object*, *name*[, *default*]) Return the value of the named attribute of *object*. *name* must be a string.

**setattr**(*object*, *name*, *value*) The arguments are an object, a string and value. The string may name an existing attribute or a new attribute.

**delattr**(*object*, *name*) This is a relative of [setattr()](https://docs.python.org/3/library/functions.html#setattr). The function deletes named attribute, if object allows it.

**hasattr**(*obj*, *name*) The arguments are an object and a string. Result is True if string is the name of one of object’s attributes, False if not.

**isinstance**(*object*, *classinfo*) Return True if the *object* argument is an instance of the *classinfo* , or of a (direct, indirect or [virtual](https://docs.python.org/3/glossary.html#term-abstract-base-class)) subclass thereof.

**issubclass**(*class*, *classinfo*) Return True if *class* is a subclass (direct, indirect or [virtual](https://docs.python.org/3/glossary.html#term-abstract-base-class)) of *classinfo*.

**vars**([*object*]) Return the [\_\_dict\_\_](https://docs.python.org/3/library/stdtypes.html#object.__dict__) attribute for a module, class, instance, or any other object with a [\_\_dict\_\_](https://docs.python.org/3/library/stdtypes.html#object.__dict__) attribute.

@**classmethod** Transform method into class method. Class method receives class as implicit first arg, just as inst method receives inst.

@**staticmethod** Transform a method into a static method. A static method does not receive an implicit first argument.

*class***memoryview**(*object*) Return a “memory view” object created from the given argument.

*class***property**(*fget=None*, *fset=None*, *fdel=None*, *doc=None*) Return a property attribute.

Mathematical:

**abs**(*x*) Return the absolute value of a number. The argument may be an integer, a float, or object implementing [\_\_abs\_\_()](https://docs.python.org/3/reference/datamodel.html#object.__abs__).

**divmod**(*a*, *b*) Take 2 numbers as arguments and return a pair of nums consisting of quotient and remainder when using int division.

**max**(*iterable*, *\**[, *key*, *default*]) Return the largest item in an iterable or the largest of two or more arguments.

**min**(*iterable*, *\**[, *key*, *default*])

**pow**(*base*, *exp*[, *mod*]) Return *base* to the power *exp*; if *mod* is present, return *base* to the power *exp*. Equivalent to base\*\*exp.

**round**(*number*[, *ndigits*]) Return *number* rounded to *ndigits* precision after the decimal point.

**sum**(*iterable*, */*, *start=0*) Sums *start* and the items of an *iterable* from left to right and returns the total.

Other:

**chr**(*i*) Return the string representing a character whose Unicode code point is the integer *i*. For example, chr(97) returns the string 'a'.

**ord**(*c*) Given a string of one Unicode character, return an integer representing the Unicode code of that character. This is the inverse of [chr()](https://docs.python.org/3/library/functions.html#chr).

**len**(*s*) Return the length of an object. The arg may be a sequence (string, bytes, tuple, list, range) or a collection (dictionary, set, or frozen set).

**ascii**(*object*) As [repr()](https://docs.python.org/3/library/functions.html#repr), return a string containing a printable representation of an object, but escape the non-ASCII characters in the

**repr**(*object*) Return string containing printable representation of obj. Return a string that yields an object with same value when passed to [eval()](https://docs.python.org/3/library/functions.html#eval).

**bin**(*x*) Convert an integer number to a binary string prefixed with “0b”. The result is a valid Python expression.

**callable**(*object*) Return [True](https://docs.python.org/3/library/constants.html#True) if the *object* argument appears callable, [False](https://docs.python.org/3/library/constants.html#False) if not. If this returns True, it is still possible that a call fails.

**dir([*object*])** Without arguments, return list of names in the current local scope. With argument, attempt to return a list of valid attributes for object.

**format**(*val*[, *spec*]) Convert a *value* to a “formatted” representation, as controlled by *format\_spec*.

**hash**(*object*) Return the hash value of the object (if it has one). Hash values are ints. They are used to quickly compare dict keys during a dict lookup.

**help**([*object*]) Invoke the built-in help system. If no argument is given, the interactive help system starts on the interpreter console.

**hex**(*x*) Convert int to a hexadecimal string prefixed with “0x”. If *x* is not [int](https://docs.python.org/3/library/functions.html#int) object, it has to define an [\_\_index\_\_()](https://docs.python.org/3/reference/datamodel.html#object.__index__) method that returns an int.

**id**(*object*) Return the “identity” of an object. This is an integer which is guaranteed to be unique and constant for this object during its lifetime.

**input**([*prompt*]) The function reads a line from input, converts it to a string (stripping a trailing newline), and returns that.

**globals**() / **locals** () Return a dictionary representing the current global/local symbol table. This is always dict of the current module.

**breakpoint**(*\*args*, *\*\*kws*) This function drops you into the debugger at the call site. Specifically, it calls [sys.breakpointhook()](https://docs.python.org/3/library/sys.html#sys.breakpointhook)

**eval**(*expr*[, *globs*[, *locs*]]) The return value is the result of the evaluated expression. Errors are reported as exceptions. Example:

**exec**(*object*[, *globals*[, *locals*]])

**compile**(*source*, *filename*, *mode*, *flags=0*, *dont\_inherit=False*, *optimize=-1*) Compile the *source* into a code or AST object.

**open**(*file*, *mode='r'*, *buffering=-1*, *encoding=None*, *errors=None*, *newline=None*, *closefd=True*, *opener=None*)

Open *file* ([path-like object](https://docs.python.org/3/glossary.html#term-path-like-object) - absolute or relative to the current working dir) and return a corresponding [file object](https://docs.python.org/3/glossary.html#term-file-object).

r-read, w-write, x-excl. creation, a-append, b-binary, t-text mode, + for updating (read&write); *errors* is opt that specifies how errors are to be handled

**print**(*\*objects*, *sep=' '*, *end='\n'*, *file=sys.stdout*, *flush=False*)

Print *objects* to the text stream *file*, separated by *sep* and followed by *end*. *sep*, *end*, *file* and *flush.*

**super**([*type*[, *object-or-type*]]) Return proxy obj that delegates method calls to parent/sibling class. Useful for accessing inherited overridden methods.

**\_\_import\_\_**(*name*, *globals=None*, *locals=None*, *fromlist=()*, *level=0*) . This function is invoked by the [import](https://docs.python.org/3/reference/simple_stmts.html#import) statement. Imports module *name*, using given *globals*/*locals* to determine how to interpret name in a package context. Use of [\_\_import\_\_()](https://docs.python.org/3/library/functions.html#__import__) is discouraged in favor of [importlib.import\_module()](https://docs.python.org/3/library/importlib.html#importlib.import_module)

**Python basic data structure methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lists/Arrays** |  | **Dictionaries** |  |
| [**append()**](https://www.w3schools.com/python/ref_list_append.asp) | Adds an element at the end of the list | [**clear()**](https://www.w3schools.com/python/ref_dictionary_clear.asp) | Removes all the elements from the dictionary |
| [**clear()**](https://www.w3schools.com/python/ref_list_clear.asp) | Removes all the elements from the list | [**copy()**](https://www.w3schools.com/python/ref_dictionary_copy.asp) | Returns a copy of the dictionary |
| [**copy()**](https://www.w3schools.com/python/ref_list_copy.asp) | Returns a copy of the list | [**fromkeys()**](https://www.w3schools.com/python/ref_dictionary_fromkeys.asp) | Returns a dictionary with the specified keys and value |
| [**count()**](https://www.w3schools.com/python/ref_list_count.asp) | Returns the num of elements with the spec value | [**get()**](https://www.w3schools.com/python/ref_dictionary_get.asp) | Returns the value of the specified key |
| [**extend()**](https://www.w3schools.com/python/ref_list_extend.asp) | Add elements of a list /iterable to the end of the list | [**items()**](https://www.w3schools.com/python/ref_dictionary_items.asp) | Returns a list containing a tuple for each key value pair |
| [**index()**](https://www.w3schools.com/python/ref_list_index.asp) | Returns the index of the first element with value | [**keys()**](https://www.w3schools.com/python/ref_dictionary_keys.asp) | Returns a list containing the dictionary's keys |
| [**insert()**](https://www.w3schools.com/python/ref_list_insert.asp) | Adds an element at the specified position | [**pop()**](https://www.w3schools.com/python/ref_dictionary_pop.asp) | Removes the element with the specified key |
| [**pop()**](https://www.w3schools.com/python/ref_list_pop.asp) | Removes the element at the specified position | [**popitem()**](https://www.w3schools.com/python/ref_dictionary_popitem.asp) | Removes the last inserted key-value pair |
| [**remove()**](https://www.w3schools.com/python/ref_list_remove.asp) | Removes the first item with the specified value | [**setdefault()**](https://www.w3schools.com/python/ref_dictionary_setdefault.asp) | Returns value of the spec key; If key does not exist: insert key:value |
| [**reverse()**](https://www.w3schools.com/python/ref_list_reverse.asp) | Reverses the order of the list | [**update()**](https://www.w3schools.com/python/ref_dictionary_update.asp) | Updates the dictionary with the specified key-value pairs |
| [**sort()**](https://www.w3schools.com/python/ref_list_sort.asp) | Sorts the list | [**values()**](https://www.w3schools.com/python/ref_dictionary_values.asp) | Returns a list of all the values in the dictionary |
|  |  |  |  |

**Strings**

**capitalize()** Converts the first character to upper case

**casefold()** Converts string into lower case

**center()**  Returns a centered string

**count()**  Returns the number of times a specified value occurs in a string

**encode()** Returns an encoded version of the string

**endswith()** Returns true if the string ends with the specified value

**expandtabs()**  Sets the tab size of the string

**find()** Searches the string for a specified value and returns the position of where it was found

**format()**  Formats specified values in a string

**format\_map()** Formats specified values in a string

**index()**  Searches the string for a specified value and returns the position of where it was found

**isalnum()** Returns True if all characters in the string are alphanumeric

**isalpha()** Returns True if all characters in the string are in the alphabet

**isdecimal()** Returns True if all characters in the string are decimals

**isdigit()**  Returns True if all characters in the string are digits

**isidentifier()** Returns True if the string is an identifier

**islower()** Returns True if all characters in the string are lower case

**isnumeric()** Returns True if all characters in the string are numeric

**isprintable()** Returns True if all characters in the string are printable

**isspace()** Returns True if all characters in the string are whitespaces

**istitle()** Returns True if the string follows the rules of a title

**isupper()** Returns True if all characters in the string are upper case

**join()** Joins the elements of an iterable to the end of the string

**ljust()**  Returns a left justified version of the string

**lower()**  Converts a string into lower case

**lstrip()** Returns a left trim version of the string

**maketrans()**  Returns a translation table to be used in translations

**partition()** Returns a tuple where the string is parted into three parts

**replace()** Returns a string where a specified value is replaced with a specified value

**rfind()**  Searches the string for a specified value and returns the last position of where it was found

**rindex()**  Searches the string for a specified value and returns the last position of where it was found

**rjust()**  Returns a right justified version of the string

**rpartition()** Returns a tuple where the string is parted into three parts

**rsplit()** Splits the string at the specified separator, and returns a list

**rstrip()**  Returns a right trim version of the string

**split()**  Splits the string at the specified separator, and returns a list

**splitlines()** Splits the string at line breaks and returns a list

**startswith()** Returns true if the string starts with the specified value

**strip()** Returns a trimmed version of the string

**swapcase()** Swaps cases, lower case becomes upper case and vice versa

**title()** Converts the first character of each word to upper case

**translate()** Returns a translated string

**upper()**  Converts a string into upper case

**zfill()** Fills the string with a specified number of 0 values at the beginning

**Sets** (Frozensets are like sets except that they cannot be changed (immutable))

**add()** Adds an element to the set

**clear()**  Removes all the elements from the set

**copy()**  Returns a copy of the set

**difference()** Returns a set containing the difference between two or more sets

**difference\_update()**  Removes the items in this set that are also included in another, specified set

**discard()** Remove the specified item

**intersection()**  Returns a set, that is the intersection of two or more sets

**intersection\_update()**  Removes the items in this set that are not present in other, specified set(s)

**isdisjoint()** Returns whether two sets have a intersection or not

**issubset()** Returns whether another set contains this set or not

**issuperset()**  Returns whether this set contains another set or not

**pop()**  Removes an element from the set

**remove()**  Removes the specified element

**symmetric\_difference()**  Returns a set with the symmetric differences of two sets

**symmetric\_difference\_update() I**nserts the symmetric differences from this set and another

**union()**  Return a set containing the union of sets

**update()**  Update the set with another set, or any other iterable

**Tuples**

**count()**  Returns the number of times a specified value occurs in a tuple

**index()**  Searches the tuple for a specified value and returns the position of where it was found

## Constructor

|  |  |
| --- | --- |
| [**DataFrame**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html#pandas.DataFrame)([data, index, columns, dtype, copy]) | Two-dimensional, size-mutable, potentially heterogeneous tabular data. |

## Attributes and underlying data

|  |  |
| --- | --- |
| [**DataFrame.index**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.index.html#pandas.DataFrame.index) | The index (row labels) of the DataFrame. |
| [**DataFrame.columns**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.columns.html#pandas.DataFrame.columns) | The column labels of the DataFrame. |
| [**DataFrame.dtypes**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.dtypes.html#pandas.DataFrame.dtypes) | Return the dtypes in the DataFrame. |
| [**DataFrame.info**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.info.html#pandas.DataFrame.info)([verbose, buf, max\_cols, …]) | Print a concise summary of a DataFrame. |
| [**DataFrame.select\_dtypes**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.select_dtypes.html#pandas.DataFrame.select_dtypes)([include, exclude]) | Return a subset of the DataFrame’s columns based on the column dtypes. |
| [**DataFrame.values**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.values.html#pandas.DataFrame.values) | Return a Numpy representation of the DataFrame. |
| [**DataFrame.axes**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.axes.html#pandas.DataFrame.axes) | Return a list representing the axes of the DataFrame. |
| [**DataFrame.ndim**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.ndim.html#pandas.DataFrame.ndim) | Return an int representing the number of axes / array dimensions. |
| [**DataFrame.size**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.size.html#pandas.DataFrame.size) | Return an int representing the number of elements in this object. |
| [**DataFrame.shape**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.shape.html#pandas.DataFrame.shape) | Return a tuple representing the dimensionality of the DataFrame. |
| [**DataFrame.memory\_usage**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.memory_usage.html#pandas.DataFrame.memory_usage)([index, deep]) | Return the memory usage of each column in bytes. |
| [**DataFrame.empty**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.empty.html#pandas.DataFrame.empty) | Indicator whether DataFrame is empty. |
| [**DataFrame.set\_flags**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.set_flags.html#pandas.DataFrame.set_flags)(\*[, copy, …]) | Return a new object with updated flags. |

## Conversion

|  |  |
| --- | --- |
| [**DataFrame.astype**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.astype.html#pandas.DataFrame.astype)(dtype[, copy, errors]) | Cast a pandas object to a specified dtype dtype. |
| [**DataFrame.convert\_dtypes**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.convert_dtypes.html#pandas.DataFrame.convert_dtypes)([infer\_objects, …]) | Convert columns to best possible dtypes using dtypes supporting pd.NA. |
| [**DataFrame.infer\_objects**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.infer_objects.html#pandas.DataFrame.infer_objects)() | Attempt to infer better dtypes for object columns. |
| [**DataFrame.copy**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.copy.html#pandas.DataFrame.copy)([deep]) | Make a copy of this object’s indices and data. |
| [**DataFrame.bool**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.bool.html#pandas.DataFrame.bool)() | Return the bool of a single element Series or DataFrame. |

## Indexing, iteration

|  |  |
| --- | --- |
| [**DataFrame.head**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.head.html#pandas.DataFrame.head)([n]) | Return the first n rows. |
| [**DataFrame.at**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.at.html#pandas.DataFrame.at) | Access a single value for a row/column label pair. |
| [**DataFrame.iat**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.iat.html#pandas.DataFrame.iat) | Access a single value for a row/column pair by integer position. |
| [**DataFrame.loc**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.loc.html#pandas.DataFrame.loc) | Access a group of rows and columns by label(s) or a boolean array. |
| [**DataFrame.iloc**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.iloc.html#pandas.DataFrame.iloc) | Purely integer-location based indexing for selection by position. |
| [**DataFrame.insert**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.insert.html#pandas.DataFrame.insert)(loc, column, value[, …]) | Insert column into DataFrame at specified location. |
| [**DataFrame.\_\_iter\_\_**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.__iter__.html#pandas.DataFrame.__iter__)() | Iterate over info axis. |
| [**DataFrame.items**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.items.html#pandas.DataFrame.items)() | Iterate over (column name, Series) pairs. |
| [**DataFrame.iteritems**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.iteritems.html#pandas.DataFrame.iteritems)() | Iterate over (column name, Series) pairs. |
| [**DataFrame.keys**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.keys.html#pandas.DataFrame.keys)() | Get the ‘info axis’ (see Indexing for more). |
| [**DataFrame.iterrows**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.iterrows.html#pandas.DataFrame.iterrows)() | Iterate over DataFrame rows as (index, Series) pairs. |
| [**DataFrame.itertuples**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.itertuples.html#pandas.DataFrame.itertuples)([index, name]) | Iterate over DataFrame rows as namedtuples. |
| [**DataFrame.lookup**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.lookup.html#pandas.DataFrame.lookup)(row\_labels, col\_labels) | (DEPRECATED) Label-based “fancy indexing” function for DataFrame. |
| [**DataFrame.pop**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.pop.html#pandas.DataFrame.pop)(item) | Return item and drop from frame. |
| [**DataFrame.tail**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.tail.html#pandas.DataFrame.tail)([n]) | Return the last n rows. |
| [**DataFrame.xs**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.xs.html#pandas.DataFrame.xs)(key[, axis, level, drop\_level]) | Return cross-section from the Series/DataFrame. |
| [**DataFrame.get**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.get.html#pandas.DataFrame.get)(key[, default]) | Get item from object for given key (ex: DataFrame column). |
| [**DataFrame.isin**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.isin.html#pandas.DataFrame.isin)(values) | Whether each element in the DataFrame is contained in values. |
| [**DataFrame.where**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.where.html#pandas.DataFrame.where)(cond[, other, inplace, …]) | Replace values where the condition is False. |
| [**DataFrame.mask**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.mask.html#pandas.DataFrame.mask)(cond[, other, inplace, axis, …]) | Replace values where the condition is True. |
| [**DataFrame.query**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.query.html#pandas.DataFrame.query)(expr[, inplace]) | Query the columns of a DataFrame with a boolean expression. |

## Binary operator functions

|  |  |
| --- | --- |
| [**DataFrame.add**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.add.html#pandas.DataFrame.add)(other[, axis, level, fill\_value]) | Get Addition of dataframe and other, element-wise (binary operator add). |
| [**DataFrame.sub**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.sub.html#pandas.DataFrame.sub)(other[, axis, level, fill\_value]) | Get Subtraction of dataframe and other, element-wise (binary operator sub). |
| [**DataFrame.mul**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.mul.html#pandas.DataFrame.mul)(other[, axis, level, fill\_value]) | Get Multiplication of dataframe and other, element-wise (binary operator mul). |
| [**DataFrame.div**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.div.html#pandas.DataFrame.div)(other[, axis, level, fill\_value]) | Get Floating division of dataframe and other, element-wise (binary operator truediv). |
| [**DataFrame.truediv**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.truediv.html#pandas.DataFrame.truediv)(other[, axis, level, …]) | Get Floating division of dataframe and other, element-wise (binary operator truediv). |
| [**DataFrame.floordiv**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.floordiv.html#pandas.DataFrame.floordiv)(other[, axis, level, …]) | Get Integer division of dataframe and other, element-wise (binary operator floordiv). |
| [**DataFrame.mod**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.mod.html#pandas.DataFrame.mod)(other[, axis, level, fill\_value]) | Get Modulo of dataframe and other, element-wise (binary operator mod). |
| [**DataFrame.pow**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.pow.html#pandas.DataFrame.pow)(other[, axis, level, fill\_value]) | Get Exponential power of dataframe and other, element-wise (binary operator pow). |
| [**DataFrame.dot**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.dot.html#pandas.DataFrame.dot)(other) | Compute the matrix multiplication between the DataFrame and other. |
| [**DataFrame.radd**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.radd.html#pandas.DataFrame.radd)(other[, axis, level, fill\_value]) | Get Addition of dataframe and other, element-wise (binary operator radd). |
| [**DataFrame.rsub**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.rsub.html#pandas.DataFrame.rsub)(other[, axis, level, fill\_value]) | Get Subtraction of dataframe and other, element-wise (binary operator rsub). |
| [**DataFrame.rmul**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.rmul.html#pandas.DataFrame.rmul)(other[, axis, level, fill\_value]) | Get Multiplication of dataframe and other, element-wise (binary operator rmul). |
| [**DataFrame.rdiv**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.rdiv.html#pandas.DataFrame.rdiv)(other[, axis, level, fill\_value]) | Get Floating division of dataframe and other, element-wise (binary operator rtruediv). |
| [**DataFrame.rtruediv**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.rtruediv.html#pandas.DataFrame.rtruediv)(other[, axis, level, …]) | Get Floating division of dataframe and other, element-wise (binary operator rtruediv). |
| [**DataFrame.rfloordiv**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.rfloordiv.html#pandas.DataFrame.rfloordiv)(other[, axis, level, …]) | Get Integer division of dataframe and other, element-wise (binary operator rfloordiv). |
| [**DataFrame.rmod**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.rmod.html#pandas.DataFrame.rmod)(other[, axis, level, fill\_value]) | Get Modulo of dataframe and other, element-wise (binary operator rmod). |
| [**DataFrame.rpow**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.rpow.html#pandas.DataFrame.rpow)(other[, axis, level, fill\_value]) | Get Exponential power of dataframe and other, element-wise (binary operator rpow). |
| [**DataFrame.lt**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.lt.html#pandas.DataFrame.lt)(other[, axis, level]) | Get Less than of dataframe and other, element-wise (binary operator lt). |
| [**DataFrame.gt**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.gt.html#pandas.DataFrame.gt)(other[, axis, level]) | Get Greater than of dataframe and other, element-wise (binary operator gt). |
| [**DataFrame.le**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.le.html#pandas.DataFrame.le)(other[, axis, level]) | Get Less than or equal to of dataframe and other, element-wise (binary operator le). |
| [**DataFrame.ge**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.ge.html#pandas.DataFrame.ge)(other[, axis, level]) | Get Greater than or equal to of dataframe and other, element-wise (binary operator ge). |
| [**DataFrame.ne**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.ne.html#pandas.DataFrame.ne)(other[, axis, level]) | Get Not equal to of dataframe and other, element-wise (binary operator ne). |
| [**DataFrame.eq**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.eq.html#pandas.DataFrame.eq)(other[, axis, level]) | Get Equal to of dataframe and other, element-wise (binary operator eq). |
| [**DataFrame.combine**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.combine.html#pandas.DataFrame.combine)(other, func[, fill\_value, …]) | Perform column-wise combine with another DataFrame. |
| [**DataFrame.combine\_first**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.combine_first.html#pandas.DataFrame.combine_first)(other) | Update null elements with value in the same location in other. |

## Function application, GroupBy & window

|  |  |
| --- | --- |
| [**DataFrame.apply**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.apply.html#pandas.DataFrame.apply)(func[, axis, raw, …]) | Apply a function along an axis of the DataFrame. |
| [**DataFrame.applymap**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.applymap.html#pandas.DataFrame.applymap)(func[, na\_action]) | Apply a function to a Dataframe elementwise. |
| [**DataFrame.pipe**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.pipe.html#pandas.DataFrame.pipe)(func, \*args, \*\*kwargs) | Apply func(self, \*args, \*\*kwargs). |
| [**DataFrame.agg**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.agg.html#pandas.DataFrame.agg)([func, axis]) | Aggregate using one or more operations over the specified axis. |
| [**DataFrame.aggregate**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.aggregate.html#pandas.DataFrame.aggregate)([func, axis]) | Aggregate using one or more operations over the specified axis. |
| [**DataFrame.transform**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.transform.html#pandas.DataFrame.transform)(func[, axis]) | Call func on self producing a DataFrame with transformed values. |
| [**DataFrame.groupby**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.groupby.html#pandas.DataFrame.groupby)([by, axis, level, …]) | Group DataFrame using a mapper or by a Series of columns. |
| [**DataFrame.rolling**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.rolling.html#pandas.DataFrame.rolling)(window[, min\_periods, …]) | Provide rolling window calculations. |
| [**DataFrame.expanding**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.expanding.html#pandas.DataFrame.expanding)([min\_periods, center, axis]) | Provide expanding transformations. |
| [**DataFrame.ewm**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.ewm.html#pandas.DataFrame.ewm)([com, span, halflife, alpha, …]) | Provide exponential weighted (EW) functions. |

## Computations / descriptive stats

|  |  |
| --- | --- |
| [**DataFrame.abs**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.abs.html#pandas.DataFrame.abs)() | Return a Series/DataFrame with absolute numeric value of each element. |
| [**DataFrame.all**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.all.html#pandas.DataFrame.all)([axis, bool\_only, skipna, level]) | Return whether all elements are True, potentially over an axis. |
| [**DataFrame.any**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.any.html#pandas.DataFrame.any)([axis, bool\_only, skipna, level]) | Return whether any element is True, potentially over an axis. |
| [**DataFrame.clip**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.clip.html#pandas.DataFrame.clip)([lower, upper, axis, inplace]) | Trim values at input threshold(s). |
| [**DataFrame.corr**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.corr.html#pandas.DataFrame.corr)([method, min\_periods]) | Compute pairwise correlation of columns, excluding NA/null values. |
| [**DataFrame.corrwith**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.corrwith.html#pandas.DataFrame.corrwith)(other[, axis, drop, method]) | Compute pairwise correlation. |
| [**DataFrame.count**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.count.html#pandas.DataFrame.count)([axis, level, numeric\_only]) | Count non-NA cells for each column or row. |
| [**DataFrame.cov**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.cov.html#pandas.DataFrame.cov)([min\_periods, ddof]) | Compute pairwise covariance of columns, excluding NA/null values. |
| [**DataFrame.cummax**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.cummax.html#pandas.DataFrame.cummax)([axis, skipna]) | Return cumulative maximum over a DataFrame or Series axis. |
| [**DataFrame.cummin**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.cummin.html#pandas.DataFrame.cummin)([axis, skipna]) | Return cumulative minimum over a DataFrame or Series axis. |
| [**DataFrame.cumprod**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.cumprod.html#pandas.DataFrame.cumprod)([axis, skipna]) | Return cumulative product over a DataFrame or Series axis. |
| [**DataFrame.cumsum**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.cumsum.html#pandas.DataFrame.cumsum)([axis, skipna]) | Return cumulative sum over a DataFrame or Series axis. |
| [**DataFrame.describe**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.describe.html#pandas.DataFrame.describe)([percentiles, include, …]) | Generate descriptive statistics. |
| [**DataFrame.diff**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.diff.html#pandas.DataFrame.diff)([periods, axis]) | First discrete difference of element. |
| [**DataFrame.eval**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.eval.html#pandas.DataFrame.eval)(expr[, inplace]) | Evaluate a string describing operations on DataFrame columns. |
| [**DataFrame.kurt**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.kurt.html#pandas.DataFrame.kurt)([axis, skipna, level, …]) | Return unbiased kurtosis over requested axis. |
| [**DataFrame.kurtosis**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.kurtosis.html#pandas.DataFrame.kurtosis)([axis, skipna, level, …]) | Return unbiased kurtosis over requested axis. |
| [**DataFrame.mad**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.mad.html#pandas.DataFrame.mad)([axis, skipna, level]) | Return the mean absolute deviation of the values over the requested axis. |
| [**DataFrame.max**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.max.html#pandas.DataFrame.max)([axis, skipna, level, …]) | Return the maximum of the values over the requested axis. |
| [**DataFrame.mean**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.mean.html#pandas.DataFrame.mean)([axis, skipna, level, …]) | Return the mean of the values over the requested axis. |
| [**DataFrame.median**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.median.html#pandas.DataFrame.median)([axis, skipna, level, …]) | Return the median of the values over the requested axis. |
| [**DataFrame.min**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.min.html#pandas.DataFrame.min)([axis, skipna, level, …]) | Return the minimum of the values over the requested axis. |
| [**DataFrame.mode**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.mode.html#pandas.DataFrame.mode)([axis, numeric\_only, dropna]) | Get the mode(s) of each element along the selected axis. |
| [**DataFrame.pct\_change**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.pct_change.html#pandas.DataFrame.pct_change)([periods, fill\_method, …]) | Percentage change between the current and a prior element. |
| [**DataFrame.prod**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.prod.html#pandas.DataFrame.prod)([axis, skipna, level, …]) | Return the product of the values over the requested axis. |
| [**DataFrame.product**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.product.html#pandas.DataFrame.product)([axis, skipna, level, …]) | Return the product of the values over the requested axis. |
| [**DataFrame.quantile**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.quantile.html#pandas.DataFrame.quantile)([q, axis, numeric\_only, …]) | Return values at the given quantile over requested axis. |
| [**DataFrame.rank**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.rank.html#pandas.DataFrame.rank)([axis, method, numeric\_only, …]) | Compute numerical data ranks (1 through n) along axis. |
| [**DataFrame.round**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.round.html#pandas.DataFrame.round)([decimals]) | Round a DataFrame to a variable number of decimal places. |
| [**DataFrame.sem**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.sem.html#pandas.DataFrame.sem)([axis, skipna, level, ddof, …]) | Return unbiased standard error of the mean over requested axis. |
| [**DataFrame.skew**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.skew.html#pandas.DataFrame.skew)([axis, skipna, level, …]) | Return unbiased skew over requested axis. |
| [**DataFrame.sum**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.sum.html#pandas.DataFrame.sum)([axis, skipna, level, …]) | Return the sum of the values over the requested axis. |
| [**DataFrame.std**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.std.html#pandas.DataFrame.std)([axis, skipna, level, ddof, …]) | Return sample standard deviation over requested axis. |
| [**DataFrame.var**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.var.html#pandas.DataFrame.var)([axis, skipna, level, ddof, …]) | Return unbiased variance over requested axis. |
| [**DataFrame.nunique**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.nunique.html#pandas.DataFrame.nunique)([axis, dropna]) | Count distinct observations over requested axis. |
| [**DataFrame.value\_counts**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.value_counts.html#pandas.DataFrame.value_counts)([subset, normalize, …]) | Return a Series containing counts of unique rows in the DataFrame. |

## Reindexing / selection / label manipulation

|  |  |
| --- | --- |
| [**DataFrame.add\_prefix**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.add_prefix.html#pandas.DataFrame.add_prefix)(prefix) | Prefix labels with string prefix. |
| [**DataFrame.add\_suffix**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.add_suffix.html#pandas.DataFrame.add_suffix)(suffix) | Suffix labels with string suffix. |
| [**DataFrame.align**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.align.html#pandas.DataFrame.align)(other[, join, axis, level, …]) | Align two objects on their axes with the specified join method. |
| [**DataFrame.at\_time**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.at_time.html#pandas.DataFrame.at_time)(time[, asof, axis]) | Select values at particular time of day (e.g., 9:30AM). |
| [**DataFrame.between\_time**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.between_time.html#pandas.DataFrame.between_time)(start\_time, end\_time) | Select values between particular times of the day (e.g., 9:00-9:30 AM). |
| [**DataFrame.drop**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.drop.html#pandas.DataFrame.drop)([labels, axis, index, …]) | Drop specified labels from rows or columns. |
| [**DataFrame.drop\_duplicates**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.drop_duplicates.html#pandas.DataFrame.drop_duplicates)([subset, keep, …]) | Return DataFrame with duplicate rows removed. |
| [**DataFrame.duplicated**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.duplicated.html#pandas.DataFrame.duplicated)([subset, keep]) | Return boolean Series denoting duplicate rows. |
| [**DataFrame.equals**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.equals.html#pandas.DataFrame.equals)(other) | Test whether two objects contain the same elements. |
| [**DataFrame.filter**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.filter.html#pandas.DataFrame.filter)([items, like, regex, axis]) | Subset the dataframe rows or columns according to the specified index labels. |
| [**DataFrame.first**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.first.html#pandas.DataFrame.first)(offset) | Select initial periods of time series data based on a date offset. |
| [**DataFrame.head**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.head.html#pandas.DataFrame.head)([n]) | Return the first n rows. |
| [**DataFrame.idxmax**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.idxmax.html#pandas.DataFrame.idxmax)([axis, skipna]) | Return index of first occurrence of maximum over requested axis. |
| [**DataFrame.idxmin**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.idxmin.html#pandas.DataFrame.idxmin)([axis, skipna]) | Return index of first occurrence of minimum over requested axis. |
| [**DataFrame.last**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.last.html#pandas.DataFrame.last)(offset) | Select final periods of time series data based on a date offset. |
| [**DataFrame.reindex**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.reindex.html#pandas.DataFrame.reindex)([labels, index, columns, …]) | Conform Series/DataFrame to new index with optional filling logic. |
| [**DataFrame.reindex\_like**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.reindex_like.html#pandas.DataFrame.reindex_like)(other[, method, …]) | Return an object with matching indices as other object. |
| [**DataFrame.rename**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.rename.html#pandas.DataFrame.rename)([mapper, index, columns, …]) | Alter axes labels. |
| [**DataFrame.rename\_axis**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.rename_axis.html#pandas.DataFrame.rename_axis)([mapper, index, …]) | Set the name of the axis for the index or columns. |
| [**DataFrame.reset\_index**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.reset_index.html#pandas.DataFrame.reset_index)([level, drop, …]) | Reset the index, or a level of it. |
| [**DataFrame.sample**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.sample.html#pandas.DataFrame.sample)([n, frac, replace, …]) | Return a random sample of items from an axis of object. |
| [**DataFrame.set\_axis**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.set_axis.html#pandas.DataFrame.set_axis)(labels[, axis, inplace]) | Assign desired index to given axis. |
| [**DataFrame.set\_index**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.set_index.html#pandas.DataFrame.set_index)(keys[, drop, append, …]) | Set the DataFrame index using existing columns. |
| [**DataFrame.tail**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.tail.html#pandas.DataFrame.tail)([n]) | Return the last n rows. |
| [**DataFrame.take**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.take.html#pandas.DataFrame.take)(indices[, axis, is\_copy]) | Return the elements in the given positional indices along an axis. |
| [**DataFrame.truncate**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.truncate.html#pandas.DataFrame.truncate)([before, after, axis, copy]) | Truncate a Series or DataFrame before and after some index value. |

## Missing data handling

|  |  |
| --- | --- |
| [**DataFrame.backfill**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.backfill.html#pandas.DataFrame.backfill)([axis, inplace, limit, …]) | Synonym for [**DataFrame.fillna()**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.fillna.html#pandas.DataFrame.fillna) with method='bfill'. |
| [**DataFrame.bfill**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.bfill.html#pandas.DataFrame.bfill)([axis, inplace, limit, downcast]) | Synonym for [**DataFrame.fillna()**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.fillna.html#pandas.DataFrame.fillna) with method='bfill'. |
| [**DataFrame.dropna**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.dropna.html#pandas.DataFrame.dropna)([axis, how, thresh, …]) | Remove missing values. |
| [**DataFrame.ffill**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.ffill.html#pandas.DataFrame.ffill)([axis, inplace, limit, downcast]) | Synonym for [**DataFrame.fillna()**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.fillna.html#pandas.DataFrame.fillna) with method='ffill'. |
| [**DataFrame.fillna**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.fillna.html#pandas.DataFrame.fillna)([value, method, axis, …]) | Fill NA/NaN values using the specified method. |
| [**DataFrame.interpolate**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.interpolate.html#pandas.DataFrame.interpolate)([method, axis, limit, …]) | Fill NaN values using an interpolation method. |
| [**DataFrame.isna**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.isna.html#pandas.DataFrame.isna)() | Detect missing values. |
| [**DataFrame.isnull**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.isnull.html#pandas.DataFrame.isnull)() | Detect missing values. |
| [**DataFrame.notna**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.notna.html#pandas.DataFrame.notna)() | Detect existing (non-missing) values. |
| [**DataFrame.notnull**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.notnull.html#pandas.DataFrame.notnull)() | Detect existing (non-missing) values. |
| [**DataFrame.pad**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.pad.html#pandas.DataFrame.pad)([axis, inplace, limit, downcast]) | Synonym for [**DataFrame.fillna()**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.fillna.html#pandas.DataFrame.fillna) with method='ffill'. |
| [**DataFrame.replace**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.replace.html#pandas.DataFrame.replace)([to\_replace, value, …]) | Replace values given in to\_replace with value. |

## Reshaping, sorting, transposing

|  |  |
| --- | --- |
| [**DataFrame.droplevel**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.droplevel.html#pandas.DataFrame.droplevel)(level[, axis]) | Return DataFrame with requested index / column level(s) removed. |
| [**DataFrame.pivot**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.pivot.html#pandas.DataFrame.pivot)([index, columns, values]) | Return reshaped DataFrame organized by given index / column values. |
| [**DataFrame.pivot\_table**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.pivot_table.html#pandas.DataFrame.pivot_table)([values, index, …]) | Create a spreadsheet-style pivot table as a DataFrame. |
| [**DataFrame.reorder\_levels**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.reorder_levels.html#pandas.DataFrame.reorder_levels)(order[, axis]) | Rearrange index levels using input order. |
| [**DataFrame.sort\_values**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.sort_values.html#pandas.DataFrame.sort_values)(by[, axis, ascending, …]) | Sort by the values along either axis. |
| [**DataFrame.sort\_index**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.sort_index.html#pandas.DataFrame.sort_index)([axis, level, …]) | Sort object by labels (along an axis). |
| [**DataFrame.nlargest**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.nlargest.html#pandas.DataFrame.nlargest)(n, columns[, keep]) | Return the first n rows ordered by columns in descending order. |
| [**DataFrame.nsmallest**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.nsmallest.html#pandas.DataFrame.nsmallest)(n, columns[, keep]) | Return the first n rows ordered by columns in ascending order. |
| [**DataFrame.swaplevel**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.swaplevel.html#pandas.DataFrame.swaplevel)([i, j, axis]) | Swap levels i and j in a MultiIndex on a particular axis. |
| [**DataFrame.stack**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.stack.html#pandas.DataFrame.stack)([level, dropna]) | Stack the prescribed level(s) from columns to index. |
| [**DataFrame.unstack**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.unstack.html#pandas.DataFrame.unstack)([level, fill\_value]) | Pivot a level of the (necessarily hierarchical) index labels. |
| [**DataFrame.swapaxes**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.swapaxes.html#pandas.DataFrame.swapaxes)(axis1, axis2[, copy]) | Interchange axes and swap values axes appropriately. |
| [**DataFrame.melt**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.melt.html#pandas.DataFrame.melt)([id\_vars, value\_vars, …]) | Unpivot a DataFrame from wide to long format, optionally leaving identifiers set. |
| [**DataFrame.explode**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.explode.html#pandas.DataFrame.explode)(column[, ignore\_index]) | Transform each element of a list-like to a row, replicating index values. |
| [**DataFrame.squeeze**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.squeeze.html#pandas.DataFrame.squeeze)([axis]) | Squeeze 1 dimensional axis objects into scalars. |
| [**DataFrame.to\_xarray**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_xarray.html#pandas.DataFrame.to_xarray)() | Return an xarray object from the pandas object. |
| [**DataFrame.T**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.T.html#pandas.DataFrame.T) |  |
| [**DataFrame.transpose**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.transpose.html#pandas.DataFrame.transpose)(\*args[, copy]) | Transpose index and columns. |

## Combining / comparing / joining / merging

|  |  |
| --- | --- |
| [**DataFrame.append**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.append.html#pandas.DataFrame.append)(other[, ignore\_index, …]) | Append rows of other to the end of caller, returning a new object. |
| [**DataFrame.assign**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.assign.html#pandas.DataFrame.assign)(\*\*kwargs) | Assign new columns to a DataFrame. |
| [**DataFrame.compare**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.compare.html#pandas.DataFrame.compare)(other[, align\_axis, …]) | Compare to another DataFrame and show the differences. |
| [**DataFrame.join**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.join.html#pandas.DataFrame.join)(other[, on, how, lsuffix, …]) | Join columns of another DataFrame. |
| [**DataFrame.merge**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.merge.html#pandas.DataFrame.merge)(right[, how, on, left\_on, …]) | Merge DataFrame or named Series objects with a database-style join. |
| [**DataFrame.update**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.update.html#pandas.DataFrame.update)(other[, join, overwrite, …]) | Modify in place using non-NA values from another DataFrame. |

## Time Series-related

|  |  |
| --- | --- |
| [**DataFrame.asfreq**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.asfreq.html#pandas.DataFrame.asfreq)(freq[, method, how, …]) | Convert TimeSeries to specified frequency. |
| [**DataFrame.asof**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.asof.html#pandas.DataFrame.asof)(where[, subset]) | Return the last row(s) without any NaNs before where. |
| [**DataFrame.shift**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.shift.html#pandas.DataFrame.shift)([periods, freq, axis, …]) | Shift index by desired number of periods with an optional time freq. |
| [**DataFrame.first\_valid\_index**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.first_valid_index.html#pandas.DataFrame.first_valid_index)() | Return index for first non-NA/null value. |
| [**DataFrame.last\_valid\_index**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.last_valid_index.html#pandas.DataFrame.last_valid_index)() | Return index for last non-NA/null value. |
| [**DataFrame.resample**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.resample.html#pandas.DataFrame.resample)(rule[, axis, closed, …]) | Resample time-series data. |
| [**DataFrame.to\_period**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_period.html#pandas.DataFrame.to_period)([freq, axis, copy]) | Convert DataFrame from DatetimeIndex to PeriodIndex. |
| [**DataFrame.to\_timestamp**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_timestamp.html#pandas.DataFrame.to_timestamp)([freq, how, axis, copy]) | Cast to DatetimeIndex of timestamps, at beginning of period. |
| [**DataFrame.tz\_convert**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.tz_convert.html#pandas.DataFrame.tz_convert)(tz[, axis, level, copy]) | Convert tz-aware axis to target time zone. |
| [**DataFrame.tz\_localize**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.tz_localize.html#pandas.DataFrame.tz_localize)(tz[, axis, level, …]) | Localize tz-naive index of a Series or DataFrame to target time zone. |

## Flags

Flags refer to attributes of the pandas obj. Dataset properties (like the date is was recorded, URL accessed from, etc.) should be stored in [**DataFrame.attrs**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.attrs.html#pandas.DataFrame.attrs).

|  |  |
| --- | --- |
| [**Flags**](https://pandas.pydata.org/docs/reference/api/pandas.Flags.html#pandas.Flags)(obj, \*, allows\_duplicate\_labels) | Flags that apply to pandas objects. |

## Metadata

[**DataFrame.attrs**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.attrs.html#pandas.DataFrame.attrs) is a dict for storing global metadata for this DataFrame. DataFrame.attrs is considered experimental and may change without warning.

|  |  |
| --- | --- |
|  |  |

## Plotting

## DataFrame.plot is both a callable method and a namespace attribute for specific plotting methods of the form DataFrame.plot.<kind>

|  |  |
| --- | --- |
| [**DataFrame.plot**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.html#pandas.DataFrame.plot)([x, y, kind, ax, ….]) | DataFrame plotting accessor and method |
| [**DataFrame.plot.area**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.area.html#pandas.DataFrame.plot.area)([x, y]) | Draw a stacked area plot. |
| [**DataFrame.plot.bar**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.bar.html#pandas.DataFrame.plot.bar)([x, y]) | Vertical bar plot. |
| [**DataFrame.plot.barh**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.barh.html#pandas.DataFrame.plot.barh)([x, y]) | Make a horizontal bar plot. |
| [**DataFrame.plot.box**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.box.html#pandas.DataFrame.plot.box)([by]) | Make a box plot of the DataFrame columns. |
| [**DataFrame.plot.density**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.density.html#pandas.DataFrame.plot.density)([bw\_method, ind]) | Generate Kernel Density Estimate plot using Gaussian kernels. |
| [**DataFrame.plot.hexbin**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.hexbin.html#pandas.DataFrame.plot.hexbin)(x, y[, C, …]) | Generate a hexagonal binning plot. |
| [**DataFrame.plot.hist**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.hist.html#pandas.DataFrame.plot.hist)([by, bins]) | Draw one histogram of the DataFrame’s columns. |
| [**DataFrame.plot.kde**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.kde.html#pandas.DataFrame.plot.kde)([bw\_method, ind]) | Generate Kernel Density Estimate plot using Gaussian kernels. |
| [**DataFrame.plot.line**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.line.html#pandas.DataFrame.plot.line)([x, y]) | Plot Series or DataFrame as lines. |
| [**DataFrame.plot.pie**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.pie.html#pandas.DataFrame.plot.pie)(\*\*kwargs) | Generate a pie plot. |
| [**DataFrame.plot.scatter**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.scatter.html#pandas.DataFrame.plot.scatter)(x, y[, s, c]) | Create a scatter plot with varying marker point size and color. |
| [**DataFrame.boxplot**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.boxplot.html#pandas.DataFrame.boxplot)([column, by, ax, …]) | Make a box plot from DataFrame columns. |
| [**DataFrame.hist**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.hist.html#pandas.DataFrame.hist)([column, by, grid, …]) | Make a histogram of the DataFrame’s. |

## Sparse accessor

Sparse-dtype specific methods and attributes are provided under the DataFrame.sparse accessor.

|  |  |
| --- | --- |
| [**DataFrame.sparse.density**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.sparse.density.html#pandas.DataFrame.sparse.density) | Ratio of non-sparse points to total (dense) data points. |
| [**DataFrame.sparse.from\_spmatrix**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.sparse.from_spmatrix.html#pandas.DataFrame.sparse.from_spmatrix)(data[, …]) | Create a new DataFrame from a scipy sparse matrix. |
| [**DataFrame.sparse.to\_coo**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.sparse.to_coo.html#pandas.DataFrame.sparse.to_coo)() | Return the contents of the frame as a sparse SciPy COO matrix. |
| [**DataFrame.sparse.to\_dense**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.sparse.to_dense.html#pandas.DataFrame.sparse.to_dense)() | Convert a DataFrame with sparse values to dense. |

## Serialization / IO / conversion

|  |  |
| --- | --- |
| [**DataFrame.from\_dict**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.from_dict.html#pandas.DataFrame.from_dict)(data[, orient, dtype, …]) | Construct DataFrame from dict of array-like or dicts. |
| [**DataFrame.from\_records**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.from_records.html#pandas.DataFrame.from_records)(data[, index, …]) | Convert structured or record ndarray to DataFrame. |
| [**DataFrame.to\_parquet**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_parquet.html#pandas.DataFrame.to_parquet)([path, engine, …]) | Write a DataFrame to the binary parquet format. |
| [**DataFrame.to\_pickle**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_pickle.html#pandas.DataFrame.to_pickle)(path[, compression, …]) | Pickle (serialize) object to file. |
| [**DataFrame.to\_csv**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_csv.html#pandas.DataFrame.to_csv)([path\_or\_buf, sep, na\_rep, …]) | Write object to a comma-separated values (csv) file. |
| [**DataFrame.to\_hdf**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_hdf.html#pandas.DataFrame.to_hdf)(path\_or\_buf, key[, mode, …]) | Write the contained data to an HDF5 file using HDFStore. |
| [**DataFrame.to\_sql**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_sql.html#pandas.DataFrame.to_sql)(name, con[, schema, …]) | Write records stored in a DataFrame to a SQL database. |
| [**DataFrame.to\_dict**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_dict.html#pandas.DataFrame.to_dict)([orient, into]) | Convert the DataFrame to a dictionary. |
| [**DataFrame.to\_excel**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_excel.html#pandas.DataFrame.to_excel)(excel\_writer[, …]) | Write object to an Excel sheet. |
| [**DataFrame.to\_json**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_json.html#pandas.DataFrame.to_json)([path\_or\_buf, orient, …]) | Convert the object to a JSON string. |
| [**DataFrame.to\_html**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_html.html#pandas.DataFrame.to_html)([buf, columns, col\_space, …]) | Render a DataFrame as an HTML table. |
| [**DataFrame.to\_feather**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_feather.html#pandas.DataFrame.to_feather)(path, \*\*kwargs) | Write a DataFrame to the binary Feather format. |
| [**DataFrame.to\_latex**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_latex.html#pandas.DataFrame.to_latex)([buf, columns, …]) | Render object to a LaTeX tabular, longtable, or nested table/tabular. |
| [**DataFrame.to\_stata**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_stata.html#pandas.DataFrame.to_stata)(path[, convert\_dates, …]) | Export DataFrame object to Stata dta format. |
| [**DataFrame.to\_gbq**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_gbq.html#pandas.DataFrame.to_gbq)(destination\_table[, …]) | Write a DataFrame to a Google BigQuery table. |
| [**DataFrame.to\_records**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_records.html#pandas.DataFrame.to_records)([index, column\_dtypes, …]) | Convert DataFrame to a NumPy record array. |
| [**DataFrame.to\_string**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_string.html#pandas.DataFrame.to_string)([buf, columns, …]) | Render a DataFrame to a console-friendly tabular output. |
| [**DataFrame.to\_clipboard**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_clipboard.html#pandas.DataFrame.to_clipboard)([excel, sep]) | Copy object to the system clipboard. |
| [**DataFrame.to\_markdown**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_markdown.html#pandas.DataFrame.to_markdown)([buf, mode, index, …]) | Print DataFrame in Markdown-friendly format. |
| [**DataFrame.style**](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.style.html#pandas.DataFrame.style) | Returns a Styler object. |

# NumPy Array creation routines

## From shape or value

|  |  |
| --- | --- |
| [**empty**](https://numpy.org/doc/stable/reference/generated/numpy.empty.html#numpy.empty)(shape[, dtype, order, like]) | Return a new array of given shape and type, without initializing entries. |
| [**empty\_like**](https://numpy.org/doc/stable/reference/generated/numpy.empty_like.html#numpy.empty_like)(prototype[, dtype, order, subok, …]) | Return a new array with the same shape and type as a given array. |
| [**eye**](https://numpy.org/doc/stable/reference/generated/numpy.eye.html#numpy.eye)(N[, M, k, dtype, order, like]) | Return a 2-D array with ones on the diagonal and zeros elsewhere. |
| [**identity**](https://numpy.org/doc/stable/reference/generated/numpy.identity.html#numpy.identity)(n[, dtype, like]) | Return the identity array. |
| [**ones**](https://numpy.org/doc/stable/reference/generated/numpy.ones.html#numpy.ones)(shape[, dtype, order, like]) | Return a new array of given shape and type, filled with ones. |
| [**ones\_like**](https://numpy.org/doc/stable/reference/generated/numpy.ones_like.html#numpy.ones_like)(a[, dtype, order, subok, shape]) | Return an array of ones with the same shape and type as a given array. |
| [**zeros**](https://numpy.org/doc/stable/reference/generated/numpy.zeros.html#numpy.zeros)(shape[, dtype, order, like]) | Return a new array of given shape and type, filled with zeros. |
| [**zeros\_like**](https://numpy.org/doc/stable/reference/generated/numpy.zeros_like.html#numpy.zeros_like)(a[, dtype, order, subok, shape]) | Return an array of zeros with the same shape and type as a given array. |
| [**full**](https://numpy.org/doc/stable/reference/generated/numpy.full.html#numpy.full)(shape, fill\_value[, dtype, order, like]) | Return a new array of given shape and type, filled with fill\_value. |
| [**full\_like**](https://numpy.org/doc/stable/reference/generated/numpy.full_like.html#numpy.full_like)(a, fill\_value[, dtype, order, …]) | Return a full array with the same shape and type as a given array. |

## From existing data

|  |  |
| --- | --- |
| [**array**](https://numpy.org/doc/stable/reference/generated/numpy.array.html#numpy.array)(object[, dtype, copy, order, subok, …]) | Create an array. |
| [**asarray**](https://numpy.org/doc/stable/reference/generated/numpy.asarray.html#numpy.asarray)(a[, dtype, order, like]) | Convert the input to an array. |
| [**asanyarray**](https://numpy.org/doc/stable/reference/generated/numpy.asanyarray.html#numpy.asanyarray)(a[, dtype, order, like]) | Convert the input to an ndarray, but pass ndarray subclasses through. |
| [**ascontiguousarray**](https://numpy.org/doc/stable/reference/generated/numpy.ascontiguousarray.html#numpy.ascontiguousarray)(a[, dtype, like]) | Return a contiguous array (ndim >= 1) in memory (C order). |
| [**asmatrix**](https://numpy.org/doc/stable/reference/generated/numpy.asmatrix.html#numpy.asmatrix)(data[, dtype]) | Interpret the input as a matrix. |
| [**copy**](https://numpy.org/doc/stable/reference/generated/numpy.copy.html#numpy.copy)(a[, order, subok]) | Return an array copy of the given object. |
| [**frombuffer**](https://numpy.org/doc/stable/reference/generated/numpy.frombuffer.html#numpy.frombuffer)(buffer[, dtype, count, offset, like]) | Interpret a buffer as a 1-dimensional array. |
| [**fromfile**](https://numpy.org/doc/stable/reference/generated/numpy.fromfile.html#numpy.fromfile)(file[, dtype, count, sep, offset, like]) | Construct an array from data in a text or binary file. |
| [**fromfunction**](https://numpy.org/doc/stable/reference/generated/numpy.fromfunction.html#numpy.fromfunction)(function, shape, \*[, dtype, like]) | Construct an array by executing a function over each coordinate. |
| [**fromiter**](https://numpy.org/doc/stable/reference/generated/numpy.fromiter.html#numpy.fromiter)(iter, dtype[, count, like]) | Create a new 1-dimensional array from an iterable object. |
| [**fromstring**](https://numpy.org/doc/stable/reference/generated/numpy.fromstring.html#numpy.fromstring)(string[, dtype, count, sep, like]) | A new 1-D array initialized from text data in a string. |
| [**loadtxt**](https://numpy.org/doc/stable/reference/generated/numpy.loadtxt.html#numpy.loadtxt)(fname[, dtype, comments, delimiter, …]) | Load data from a text file. |

## Creating record arrays (numpy.rec)

**numpy.rec** is the preferred alias for **numpy.core.records**.

|  |  |
| --- | --- |
| [**core.records.array**](https://numpy.org/doc/stable/reference/generated/numpy.core.records.array.html#numpy.core.records.array)(obj[, dtype, shape, …]) | Construct a record array from a wide-variety of objects. |
| [**core.records.fromarrays**](https://numpy.org/doc/stable/reference/generated/numpy.core.records.fromarrays.html#numpy.core.records.fromarrays)(arrayList[, dtype, …]) | Create a record array from a (flat) list of arrays |
| [**core.records.fromrecords**](https://numpy.org/doc/stable/reference/generated/numpy.core.records.fromrecords.html#numpy.core.records.fromrecords)(recList[, dtype, …]) | Create a recarray from a list of records in text form. |
| [**core.records.fromstring**](https://numpy.org/doc/stable/reference/generated/numpy.core.records.fromstring.html#numpy.core.records.fromstring)(datastring[, dtype, …]) | Create a record array from binary data |
| [**core.records.fromfile**](https://numpy.org/doc/stable/reference/generated/numpy.core.records.fromfile.html#numpy.core.records.fromfile)(fd[, dtype, shape, …]) | Create an array from binary file data |

## Creating character arrays ([numpy.char](https://numpy.org/doc/stable/reference/routines.char.html#module-numpy.char))

[**numpy.char**](https://numpy.org/doc/stable/reference/routines.char.html#module-numpy.char) is the preferred alias for **numpy.core.defchararray**.

|  |  |
| --- | --- |
| [**core.defchararray.array**](https://numpy.org/doc/stable/reference/generated/numpy.core.defchararray.array.html#numpy.core.defchararray.array)(obj[, itemsize, …]) | Create a [**chararray**](https://numpy.org/doc/stable/reference/generated/numpy.chararray.html#numpy.chararray). |
| [**core.defchararray.asarray**](https://numpy.org/doc/stable/reference/generated/numpy.core.defchararray.asarray.html#numpy.core.defchararray.asarray)(obj[, itemsize, …]) | Convert the input to a [**chararray**](https://numpy.org/doc/stable/reference/generated/numpy.chararray.html#numpy.chararray), copying the data only if necessary. |

## Numerical ranges

|  |  |
| --- | --- |
| [**arange**](https://numpy.org/doc/stable/reference/generated/numpy.arange.html#numpy.arange)([start,] stop[, step,][, dtype, like]) | Return evenly spaced values within a given interval. |
| [**linspace**](https://numpy.org/doc/stable/reference/generated/numpy.linspace.html#numpy.linspace)(start, stop[, num, endpoint, …]) | Return evenly spaced numbers over a specified interval. |
| [**logspace**](https://numpy.org/doc/stable/reference/generated/numpy.logspace.html#numpy.logspace)(start, stop[, num, endpoint, base, …]) | Return numbers spaced evenly on a log scale. |
| [**geomspace**](https://numpy.org/doc/stable/reference/generated/numpy.geomspace.html#numpy.geomspace)(start, stop[, num, endpoint, …]) | Return numbers spaced evenly on a log scale (a geometric progression). |
| [**meshgrid**](https://numpy.org/doc/stable/reference/generated/numpy.meshgrid.html#numpy.meshgrid)(\*xi[, copy, sparse, indexing]) | Return coordinate matrices from coordinate vectors. |
| [**mgrid**](https://numpy.org/doc/stable/reference/generated/numpy.mgrid.html#numpy.mgrid) | nd\_grid instance which returns a dense multi-dimensional “meshgrid”. |
| [**ogrid**](https://numpy.org/doc/stable/reference/generated/numpy.ogrid.html#numpy.ogrid) | nd\_grid instance which returns an open multi-dimensional “meshgrid”. |

## Building matrices

|  |  |
| --- | --- |
| [**diag**](https://numpy.org/doc/stable/reference/generated/numpy.diag.html#numpy.diag)(v[, k]) | Extract a diagonal or construct a diagonal array. |
| [**diagflat**](https://numpy.org/doc/stable/reference/generated/numpy.diagflat.html#numpy.diagflat)(v[, k]) | Create a two-dimensional array with the flattened input as a diagonal. |
| [**tri**](https://numpy.org/doc/stable/reference/generated/numpy.tri.html#numpy.tri)(N[, M, k, dtype, like]) | An array with ones at and below the given diagonal and zeros elsewhere. |
| [**tril**](https://numpy.org/doc/stable/reference/generated/numpy.tril.html#numpy.tril)(m[, k]) | Lower triangle of an array. |
| [**triu**](https://numpy.org/doc/stable/reference/generated/numpy.triu.html#numpy.triu)(m[, k]) | Upper triangle of an array. |
| [**vander**](https://numpy.org/doc/stable/reference/generated/numpy.vander.html#numpy.vander)(x[, N, increasing]) | Generate a Vandermonde matrix. |

## The Matrix class

|  |  |
| --- | --- |
| [**mat**](https://numpy.org/doc/stable/reference/generated/numpy.mat.html#numpy.mat)(data[, dtype]) | Interpret the input as a matrix. |
| [**bmat**](https://numpy.org/doc/stable/reference/generated/numpy.bmat.html#numpy.bmat)(obj[, ldict, gdict]) | Build a matrix object from a string, nested sequence, or array. |

# Array manipulation routines

## Basic operations

|  |  |
| --- | --- |
| [**copyto**](https://numpy.org/doc/stable/reference/generated/numpy.copyto.html#numpy.copyto)(dst, src[, casting, where]) | Copies values from one array to another, broadcasting as necessary. |
| [**shape**](https://numpy.org/doc/stable/reference/generated/numpy.shape.html#numpy.shape)(a) | Return the shape of an array. |

## Changing array shape

|  |  |
| --- | --- |
| [**reshape**](https://numpy.org/doc/stable/reference/generated/numpy.reshape.html#numpy.reshape)(a, newshape[, order]) | Gives a new shape to an array without changing its data. |
| [**ravel**](https://numpy.org/doc/stable/reference/generated/numpy.ravel.html#numpy.ravel)(a[, order]) | Return a contiguous flattened array. |
| [**ndarray.flat**](https://numpy.org/doc/stable/reference/generated/numpy.ndarray.flat.html#numpy.ndarray.flat) | A 1-D iterator over the array. |
| [**ndarray.flatten**](https://numpy.org/doc/stable/reference/generated/numpy.ndarray.flatten.html#numpy.ndarray.flatten)([order]) | Return a copy of the array collapsed into one dimension. |
| [**moveaxis**](https://numpy.org/doc/stable/reference/generated/numpy.moveaxis.html#numpy.moveaxis)(a, source, destination) | Move axes of an array to new positions. |
| [**rollaxis**](https://numpy.org/doc/stable/reference/generated/numpy.rollaxis.html#numpy.rollaxis)(a, axis[, start]) | Roll the specified axis backwards, until it lies in a given position. |
| [**swapaxes**](https://numpy.org/doc/stable/reference/generated/numpy.swapaxes.html#numpy.swapaxes)(a, axis1, axis2) | Interchange two axes of an array. |
| [**ndarray.T**](https://numpy.org/doc/stable/reference/generated/numpy.ndarray.T.html#numpy.ndarray.T) | The transposed array. |
| [**transpose**](https://numpy.org/doc/stable/reference/generated/numpy.transpose.html#numpy.transpose)(a[, axes]) | Reverse or permute the axes of an array; returns the modified array. |

## Changing number of dimensions

|  |  |
| --- | --- |
| [**atleast\_1d**](https://numpy.org/doc/stable/reference/generated/numpy.atleast_1d.html#numpy.atleast_1d)(\*arys) | Convert inputs to arrays with at least one dimension. |
| [**atleast\_2d**](https://numpy.org/doc/stable/reference/generated/numpy.atleast_2d.html#numpy.atleast_2d)(\*arys) | View inputs as arrays with at least two dimensions. |
| [**atleast\_3d**](https://numpy.org/doc/stable/reference/generated/numpy.atleast_3d.html#numpy.atleast_3d)(\*arys) | View inputs as arrays with at least three dimensions. |
| [**broadcast**](https://numpy.org/doc/stable/reference/generated/numpy.broadcast.html#numpy.broadcast) | Produce an object that mimics broadcasting. |
| [**broadcast\_to**](https://numpy.org/doc/stable/reference/generated/numpy.broadcast_to.html#numpy.broadcast_to)(array, shape[, subok]) | Broadcast an array to a new shape. |
| [**broadcast\_arrays**](https://numpy.org/doc/stable/reference/generated/numpy.broadcast_arrays.html#numpy.broadcast_arrays)(\*args[, subok]) | Broadcast any number of arrays against each other. |
| [**expand\_dims**](https://numpy.org/doc/stable/reference/generated/numpy.expand_dims.html#numpy.expand_dims)(a, axis) | Expand the shape of an array. |
| [**squeeze**](https://numpy.org/doc/stable/reference/generated/numpy.squeeze.html#numpy.squeeze)(a[, axis]) | Remove axes of length one from a. |

## Changing kind of array

|  |  |
| --- | --- |
| [**asarray**](https://numpy.org/doc/stable/reference/generated/numpy.asarray.html#numpy.asarray)(a[, dtype, order, like]) | Convert the input to an array. |
| [**asanyarray**](https://numpy.org/doc/stable/reference/generated/numpy.asanyarray.html#numpy.asanyarray)(a[, dtype, order, like]) | Convert the input to an ndarray, but pass ndarray subclasses through. |
| [**asmatrix**](https://numpy.org/doc/stable/reference/generated/numpy.asmatrix.html#numpy.asmatrix)(data[, dtype]) | Interpret the input as a matrix. |
| [**asfarray**](https://numpy.org/doc/stable/reference/generated/numpy.asfarray.html#numpy.asfarray)(a[, dtype]) | Return an array converted to a float type. |
| [**asfortranarray**](https://numpy.org/doc/stable/reference/generated/numpy.asfortranarray.html#numpy.asfortranarray)(a[, dtype, like]) | Return an array (ndim >= 1) laid out in Fortran order in memory. |
| [**ascontiguousarray**](https://numpy.org/doc/stable/reference/generated/numpy.ascontiguousarray.html#numpy.ascontiguousarray)(a[, dtype, like]) | Return a contiguous array (ndim >= 1) in memory (C order). |
| [**asarray\_chkfinite**](https://numpy.org/doc/stable/reference/generated/numpy.asarray_chkfinite.html#numpy.asarray_chkfinite)(a[, dtype, order]) | Convert the input to an array, checking for NaNs or Infs. |
| [**asscalar**](https://numpy.org/doc/stable/reference/generated/numpy.asscalar.html#numpy.asscalar)(a) | Convert an array of size 1 to its scalar equivalent. |
| [**require**](https://numpy.org/doc/stable/reference/generated/numpy.require.html#numpy.require)(a[, dtype, requirements, like]) | Return an ndarray of the provided type that satisfies requirements. |

## Joining/Splitting arrays

|  |  |
| --- | --- |
| [**concatenate**](https://numpy.org/doc/stable/reference/generated/numpy.concatenate.html#numpy.concatenate)([axis, out, dtype, casting]) | Join a sequence of arrays along an existing axis. |
| [**stack**](https://numpy.org/doc/stable/reference/generated/numpy.stack.html#numpy.stack)(arrays[, axis, out]) | Join a sequence of arrays along a new axis. |
| [**block**](https://numpy.org/doc/stable/reference/generated/numpy.block.html#numpy.block)(arrays) | Assemble an nd-array from nested lists of blocks. |
| [**vstack**](https://numpy.org/doc/stable/reference/generated/numpy.vstack.html#numpy.vstack)(tup) | Stack arrays in sequence vertically (row wise). |
| [**hstack**](https://numpy.org/doc/stable/reference/generated/numpy.hstack.html#numpy.hstack)(tup) | Stack arrays in sequence horizontally (column wise). |
| [**dstack**](https://numpy.org/doc/stable/reference/generated/numpy.dstack.html#numpy.dstack)(tup) | Stack arrays in sequence depth wise (along third axis). |
| [**column\_stack**](https://numpy.org/doc/stable/reference/generated/numpy.column_stack.html#numpy.column_stack)(tup) | Stack 1-D arrays as columns into a 2-D array. |
| [**row\_stack**](https://numpy.org/doc/stable/reference/generated/numpy.row_stack.html#numpy.row_stack)(tup) | Stack arrays in sequence vertically (row wise). |
| [**split**](https://numpy.org/doc/stable/reference/generated/numpy.split.html#numpy.split)(ary, indices\_or\_sections[, axis]) | Split an array into multiple sub-arrays as views into ary. |
| [**array\_split**](https://numpy.org/doc/stable/reference/generated/numpy.array_split.html#numpy.array_split)(ary, indices\_or\_sections[, axis]) | Split an array into multiple sub-arrays. |
| [**dsplit**](https://numpy.org/doc/stable/reference/generated/numpy.dsplit.html#numpy.dsplit)(ary, indices\_or\_sections) | Split array into multiple sub-arrays along the 3rd axis (depth). |
| [**hsplit**](https://numpy.org/doc/stable/reference/generated/numpy.hsplit.html#numpy.hsplit)(ary, indices\_or\_sections) | Split an array into multiple sub-arrays horizontally (column-wise). |
| [**vsplit**](https://numpy.org/doc/stable/reference/generated/numpy.vsplit.html#numpy.vsplit)(ary, indices\_or\_sections) | Split an array into multiple sub-arrays vertically (row-wise). |

## Tiling arrays

|  |  |
| --- | --- |
| [**tile**](https://numpy.org/doc/stable/reference/generated/numpy.tile.html#numpy.tile)(A, reps) | Construct an array by repeating A the number of times given by reps. |
| [**repeat**](https://numpy.org/doc/stable/reference/generated/numpy.repeat.html#numpy.repeat)(a, repeats[, axis]) | Repeat elements of an array. |

## Adding and removing elements

|  |  |
| --- | --- |
| [**delete**](https://numpy.org/doc/stable/reference/generated/numpy.delete.html#numpy.delete)(arr, obj[, axis]) | Return a new array with sub-arrays along an axis deleted. |
| [**insert**](https://numpy.org/doc/stable/reference/generated/numpy.insert.html#numpy.insert)(arr, obj, values[, axis]) | Insert values along the given axis before the given indices. |
| [**append**](https://numpy.org/doc/stable/reference/generated/numpy.append.html#numpy.append)(arr, values[, axis]) | Append values to the end of an array. |
| [**resize**](https://numpy.org/doc/stable/reference/generated/numpy.resize.html#numpy.resize)(a, new\_shape) | Return a new array with the specified shape. |
| [**trim\_zeros**](https://numpy.org/doc/stable/reference/generated/numpy.trim_zeros.html#numpy.trim_zeros)(filt[, trim]) | Trim the leading and/or trailing zeros from a 1-D array or sequence. |
| [**unique**](https://numpy.org/doc/stable/reference/generated/numpy.unique.html#numpy.unique)(ar[, return\_index, return\_inverse, …]) | Find the unique elements of an array. |

## Rearranging elements

|  |  |
| --- | --- |
| [**flip**](https://numpy.org/doc/stable/reference/generated/numpy.flip.html#numpy.flip)(m[, axis]) | Reverse the order of elements in an array along the given axis. |
| [**fliplr**](https://numpy.org/doc/stable/reference/generated/numpy.fliplr.html#numpy.fliplr)(m) | Reverse the order of elements along axis 1 (left/right). |
| [**flipud**](https://numpy.org/doc/stable/reference/generated/numpy.flipud.html#numpy.flipud)(m) | Reverse the order of elements along axis 0 (up/down). |
| [**reshape**](https://numpy.org/doc/stable/reference/generated/numpy.reshape.html#numpy.reshape)(a, newshape[, order]) | Gives a new shape to an array without changing its data. |
| [**roll**](https://numpy.org/doc/stable/reference/generated/numpy.roll.html#numpy.roll)(a, shift[, axis]) | Roll array elements along a given axis. |
| [**rot90**](https://numpy.org/doc/stable/reference/generated/numpy.rot90.html#numpy.rot90)(m[, k, axes]) | Rotate an array by 90 degrees in the plane specified by axes. |

# Pandas - general functions

## Data manipulations

|  |  |
| --- | --- |
| [**melt**](https://pandas.pydata.org/docs/reference/api/pandas.melt.html#pandas.melt)(frame[, id\_vars, value\_vars, var\_name, …]) | Unpivot a DataFrame from wide to long format, optionally leaving identifiers set. |
| [**pivot**](https://pandas.pydata.org/docs/reference/api/pandas.pivot.html#pandas.pivot)(data[, index, columns, values]) | Return reshaped DataFrame organized by given index / column values. |
| [**pivot\_table**](https://pandas.pydata.org/docs/reference/api/pandas.pivot_table.html#pandas.pivot_table)(data[, values, index, columns, …]) | Create a spreadsheet-style pivot table as a DataFrame. |
| [**crosstab**](https://pandas.pydata.org/docs/reference/api/pandas.crosstab.html#pandas.crosstab)(index, columns[, values, rownames, …]) | Compute a simple cross tabulation of two (or more) factors. |
| [**cut**](https://pandas.pydata.org/docs/reference/api/pandas.cut.html#pandas.cut)(x, bins[, right, labels, retbins, …]) | Bin values into discrete intervals. |
| [**qcut**](https://pandas.pydata.org/docs/reference/api/pandas.qcut.html#pandas.qcut)(x, q[, labels, retbins, precision, …]) | Quantile-based discretization function. |
| [**merge**](https://pandas.pydata.org/docs/reference/api/pandas.merge.html#pandas.merge)(left, right[, how, on, left\_on, …]) | Merge DataFrame or named Series objects with a database-style join. |
| [**merge\_ordered**](https://pandas.pydata.org/docs/reference/api/pandas.merge_ordered.html#pandas.merge_ordered)(left, right[, on, left\_on, …]) | Perform merge with optional filling/interpolation. |
| [**merge\_asof**](https://pandas.pydata.org/docs/reference/api/pandas.merge_asof.html#pandas.merge_asof)(left, right[, on, left\_on, …]) | Perform an asof merge. |
| [**concat**](https://pandas.pydata.org/docs/reference/api/pandas.concat.html#pandas.concat)(objs[, axis, join, ignore\_index, …]) | Concatenate pandas objects along a particular axis with optional set logic along the other axes. |
| [**get\_dummies**](https://pandas.pydata.org/docs/reference/api/pandas.get_dummies.html#pandas.get_dummies)(data[, prefix, prefix\_sep, …]) | Convert categorical variable into dummy/indicator variables. |
| [**factorize**](https://pandas.pydata.org/docs/reference/api/pandas.factorize.html#pandas.factorize)(values[, sort, na\_sentinel, size\_hint]) | Encode the object as an enumerated type or categorical variable. |
| [**unique**](https://pandas.pydata.org/docs/reference/api/pandas.unique.html#pandas.unique)(values) | Hash table-based unique. |
| [**wide\_to\_long**](https://pandas.pydata.org/docs/reference/api/pandas.wide_to_long.html#pandas.wide_to_long)(df, stubnames, i, j[, sep, suffix]) | Wide panel to long format. |

## Top-level missing data

|  |  |
| --- | --- |
| [**isna**](https://pandas.pydata.org/docs/reference/api/pandas.isna.html#pandas.isna)(obj) | Detect missing values for an array-like object. |
| [**isnull**](https://pandas.pydata.org/docs/reference/api/pandas.isnull.html#pandas.isnull)(obj) | Detect missing values for an array-like object. |
| [**notna**](https://pandas.pydata.org/docs/reference/api/pandas.notna.html#pandas.notna)(obj) | Detect non-missing values for an array-like object. |
| [**notnull**](https://pandas.pydata.org/docs/reference/api/pandas.notnull.html#pandas.notnull)(obj) | Detect non-missing values for an array-like object. |

## Top-level conversions

|  |  |
| --- | --- |
| [**to\_numeric**](https://pandas.pydata.org/docs/reference/api/pandas.to_numeric.html#pandas.to_numeric)(arg[, errors, downcast]) | Convert argument to a numeric type. |

## Top-level dealing with datetimelike

|  |  |
| --- | --- |
| [**to\_datetime**](https://pandas.pydata.org/docs/reference/api/pandas.to_datetime.html#pandas.to_datetime)(arg[, errors, dayfirst, …]) | Convert argument to datetime. |
| [**to\_timedelta**](https://pandas.pydata.org/docs/reference/api/pandas.to_timedelta.html#pandas.to_timedelta)(arg[, unit, errors]) | Convert argument to timedelta. |
| [**date\_range**](https://pandas.pydata.org/docs/reference/api/pandas.date_range.html#pandas.date_range)([start, end, periods, freq, tz, …]) | Return a fixed frequency DatetimeIndex. |
| [**bdate\_range**](https://pandas.pydata.org/docs/reference/api/pandas.bdate_range.html#pandas.bdate_range)([start, end, periods, freq, tz, …]) | Return a fixed frequency DatetimeIndex, with business day as the default frequency. |
| [**period\_range**](https://pandas.pydata.org/docs/reference/api/pandas.period_range.html#pandas.period_range)([start, end, periods, freq, name]) | Return a fixed frequency PeriodIndex. |
| [**timedelta\_range**](https://pandas.pydata.org/docs/reference/api/pandas.timedelta_range.html#pandas.timedelta_range)([start, end, periods, freq, …]) | Return a fixed frequency TimedeltaIndex, with day as the default frequency. |
| [**infer\_freq**](https://pandas.pydata.org/docs/reference/api/pandas.infer_freq.html#pandas.infer_freq)(index[, warn]) | Infer the most likely frequency given the input index. |

## Top-level dealing with intervals

|  |  |
| --- | --- |
| [**interval\_range**](https://pandas.pydata.org/docs/reference/api/pandas.interval_range.html#pandas.interval_range)([start, end, periods, freq, …]) | Return a fixed frequency IntervalIndex. |

## Top-level evaluation

|  |  |
| --- | --- |
| [**eval**](https://pandas.pydata.org/docs/reference/api/pandas.eval.html#pandas.eval)(expr[, parser, engine, truediv, …]) | Evaluate a Python expression as a string using various backends. |

## Hashing

|  |  |
| --- | --- |
| [**util.hash\_array**](https://pandas.pydata.org/docs/reference/api/pandas.util.hash_array.html#pandas.util.hash_array)(vals[, encoding, hash\_key, …]) | Given a 1d array, return an array of deterministic integers. |
| [**util.hash\_pandas\_object**](https://pandas.pydata.org/docs/reference/api/pandas.util.hash_pandas_object.html#pandas.util.hash_pandas_object)(obj[, index, …]) | Return a data hash of the Index/Series/DataFrame. |

## Testing

|  |  |
| --- | --- |
| [**test**](https://pandas.pydata.org/docs/reference/api/pandas.test.html#pandas.test)([extra\_args]) |  |