

[Scipy.org \(http://scipy.org/\)](http://scipy.org/) [Docs \(http://docs.scipy.org/\)](http://docs.scipy.org/) [NumPy v1.12 Manual \(../index.html\)](#)
[NumPy Reference \(../index.html\)](#) [Array objects \(../arrays.html\)](#)
[Standard array subclasses \(../arrays.classes.html\)](#) [numpy.matrix \(numpy.matrix.html\)](#)
[index \(../genindex.html\)](#) [next \(numpy.matrix.var.html\)](#) [previous \(numpy.matrix.trace.html\)](#)

numpy.matrix.transpose

matrix.transpose(*axes)

Returns a view of the array with axes transposed.

For a 1-D array, this has no effect. (To change between column and row vectors, first cast the 1-D array into a matrix object.) For a 2-D array, this is the usual matrix transpose. For an n-D array, if axes are given, their order indicates how the axes are permuted (see Examples). If axes are not provided and `a.shape = (i[0], i[1], ... i[n-2], i[n-1])`, then `a.transpose().shape = (i[n-1], i[n-2], ... i[1], i[0])`.

Parameters: `axes` : *None, tuple of ints, or n ints*

- None or no argument: reverses the order of the axes.
- tuple of ints: *i* in the *j*-th place in the tuple means *a*'s *i*-th axis becomes *a.transpose()*'s *j*-th axis.
- *n* ints: same as an n-tuple of the same ints (this form is intended simply as a "convenience" alternative to the tuple form)

Returns: `out` : *ndarray*

View of *a*, with axes suitably permuted.

See also:

`ndarray.T` ([numpy.ndarray.T.html#numpy.ndarray.T](#)) Array property returning the array transposed.

Examples

```
>>> a = np.array([[1, 2], [3, 4]]) >>>
>>> a
array([[1, 2],
       [3, 4]])
>>> a.transpose()
array([[1, 3],
       [2, 4]])
>>> a.transpose((1, 0))
array([[1, 3],
       [2, 4]])
>>> a.transpose(1, 0)
array([[1, 3],
       [2, 4]])
```

Previous topic

[numpy.matrix.trace \(numpy.matrix.trace.html\)](#)

Next topic

[numpy.matrix.var \(numpy.matrix.var.html\)](#)