Search the docs ...

Array objects
Array API Standard Compatibility
Constants
Universal functions (ufunc)
Routines
Array creation routines
Array manipulation routines
numpy.copyto
numpy.shape
numpy.reshape
numpy.ravel
numpy.ndarray.flat
numpy.ndarray.flatten
numpy.moveaxis
numpy.rollaxis
numpy.swapaxes
numpy.ndarray.T
numpy.transpose
numpy.atleast_1d
numpy.atleast_2d
numpy.atleast_3d
numpy.broadcast
numpy.broadcast_to
numpy.broadcast_arrays
numpy.expand_dims
numpy.squeeze
numpy.asarray
numpy.asanyarray
numpy.asmatrix
numpy.asfarray
numpy.asfortranarray
numpy.ascontiguousarray
numpy.asarray_chkfinite
numpy.require
numpy.concatenate
numpy.stack
numpy.block
numpy.vstack
numpy.hstack
numpy.dstack
numpy.column_stack
numpy.row_stack
numpy.split
numpy.array_split

numpy.transpose

numpy.transpose(*a*, *axes=None*) [source]

Returns an array with axes transposed.

For a 1-D array, this returns an unchanged view of the original array, as a transposed vector is simply the same vector. To convert a 1-D array into a 2-D column vector, an additional dimension must be added, e.g., `np.atleast2d(a).T` achieves this, as does `a[:, np.newaxis]`. For a 2-D array, this is the standard 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, then `transpose(a).shape == a.shape[::-1]`.

Parameters: **a** : *array_like*
Input array.

axes : *tuple or list of ints, optional*
If specified, it must be a tuple or list which contains a permutation of [0,1,...,N-1] where N is the number of axes of *a*. The *i*'th axis of the returned array will correspond to the axis numbered `axes[i]` of the input. If not specified, defaults to `range(a.ndim)[::-1]`, which reverses the order of the axes.

Returns: **p** : *ndarray*
a with its axes permuted. A view is returned whenever possible.

See also

ndarray.transpose
Equivalent method.

moveaxis
Move axes of an array to new positions.

argsort
Return the indices that would sort an array.

Notes

Use `transpose(a, argsort(axes))` to invert the transposition of tensors when using the *axes* keyword argument.

Examples

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

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

```
>>> a = np.ones((1, 2, 3))
>>> np.transpose(a, (1, 0, 2)).shape
(2, 1, 3)
```

```
>>> a = np.ones((2, 3, 4, 5))
>>> np.transpose(a).shape
(5, 4, 3, 2)
```

Previous
numpy.swapaxes

Next
numpy.atleast_1d