

 Search the docs ...

[Array objects](#)

[Array API Standard](#)

[Compatibility](#)

[Constants](#)

[Universal functions \(ufunc\)](#)

[Routines](#)

^

[Array creation routines](#)

[Array manipulation routines](#)

[Binary operations](#)

[String operations](#)

[C-Types foreign function interface \(numpy.ctypeslib\)](#)

[Datetime support functions](#)

[Data type routines](#)

[Mathematical functions with automatic domain](#)

[Floating point error handling](#)

[Discrete Fourier Transform \(numpy.fft\)](#)

[Functional programming](#)

[NumPy-specific help functions](#)

[Input and output](#)

[Linear algebra \(numpy.linalg\)](#)

[Logic functions](#)

^

[numpy.all](#)

[numpy.any](#)

[numpy.isfinite](#)

[numpy.isinf](#)

[numpy.isnan](#)

[numpy.isnat](#)

[numpy.isneginf](#)

[numpy.isposinf](#)

[numpy.iscomplex](#)

[numpy.iscomplexobj](#)

[numpy.isfortran](#)

[numpy.isreal](#)

[numpy.isrealobj](#)

[numpy.isscalar](#)

[numpy.logical_and](#)

[numpy.logical_or](#)

[numpy.logical_not](#)

[numpy.logical_xor](#)

numpy.isclose

numpy.isclose(*a*, *b*, *rtol*=1e-05, *atol*=1e-08, *equal_nan*=False) [source]

Returns a boolean array where two arrays are element-wise equal within a tolerance.

The tolerance values are positive, typically very small numbers. The relative difference (*rtol* * abs(*b*)) and the absolute difference *atol* are added together to compare against the absolute difference between *a* and *b*.

Warning

The default *atol* is not appropriate for comparing numbers that are much smaller than one (see Notes).

Parameters: *a*, *b* : *array_like*

Input arrays to compare.

rtol : *float*

The relative tolerance parameter (see Notes).

atol : *float*

The absolute tolerance parameter (see Notes).

equal_nan : *bool*

Whether to compare NaN's as equal. If True, NaN's in *a* will be considered equal to NaN's in *b* in the output array.

Returns: *y* : *array_like*

Returns a boolean array of where *a* and *b* are equal within the given tolerance. If both *a* and *b* are scalars, returns a single boolean value.

See also

- [allclose](#)
- [math.isclose](#)

Notes

New in version 1.7.0.

For finite values, isclose uses the following equation to test whether two floating point values are equivalent.

$$\text{absolute}(a - b) \leq (\text{atol} + \text{rtol} * \text{absolute}(b))$$

Unlike the built-in [math.isclose](#), the above equation is not symmetric in *a* and *b* – it assumes *b* is the reference value – so that *isclose(a, b)* might be different from *isclose(b, a)*. Furthermore, the default value of *atol* is not zero, and is used to determine what small values should be considered close to zero. The default value is appropriate for expected values of order unity: if the expected values are significantly smaller than one, it can result in false positives. *atol* should be carefully selected for the use case at hand. A zero value for *atol* will result in *False* if either *a* or *b* is zero.

[isclose](#) is not defined for non-numeric data types. *bool* is considered a numeric data-type for this purpose.

Examples

https://numpy.org/doc/stable/reference/generated/numpy.isclose.html

1/2

```
>>> np.isclose([1e10, 1e-7], [1.00001e10, 1e-8])
array([ True, False])
>>> np.isclose([1e10, 1e-8], [1.00001e10, 1e-9])
array([ True, True])
>>> np.isclose([1e10, 1e-8], [1.0001e10, 1e-9])
array([False,  True])
>>> np.isclose([1.0, np.nan], [1.0, np.nan])
array([ True, False])
>>> np.isclose([1.0, np.nan], [1.0, np.nan], equal_nan=True)
array([ True, True])
>>> np.isclose([1e-8, 1e-7], [0.0, 0.0])
array([ True, False])
>>> np.isclose([1e-100, 1e-7], [0.0, 0.0], atol=0.0)
array([False, False])
>>> np.isclose([1e-10, 1e-10], [1e-20, 0.0])
array([ True,  True])
>>> np.isclose([1e-10, 1e-10], [1e-20, 0.999999e-10], atol=0.0)
array([False,  True])
```

◀ Previous
[numpy.allclose](#)

Next ▶
[numpy.array_equal](#)