

Scipy.org (<http://scipy.org/>) Docs (<http://docs.scipy.org/>)

NumPy v1.10 Manual ([../index.html](http://docs.scipy.org/doc/numpy-1.10.1/index.html)) NumPy Reference ([../index.html](http://docs.scipy.org/doc/numpy-1.10.1/reference/index.html))

Routines ([../routines.html](http://docs.scipy.org/doc/numpy-1.10.1/routines.html)) Mathematical functions ([../routines.math.html](http://docs.scipy.org/doc/numpy-1.10.1/routines.math.html))

[index \(../genindex.html\)](http://docs.scipy.org/doc/numpy-1.10.1/genindex.html) [next \(numpy.nansum.html\)](http://docs.scipy.org/doc/numpy-1.10.1/numpy.nansum.html) [previous \(numpy.prod.html\)](http://docs.scipy.org/doc/numpy-1.10.1/numpy.prod.html)

This is documentation for an old release of NumPy (version 1.10.1). Read this page (<https://numpy.org/doc/stable/reference/generated/numpy.sum.html>) in the documentation of the latest stable release (<https://numpy.org/doc/stable/>) (version > 1.17).

numpy.sum

numpy.sum(*a*, *axis=None*, *dtype=None*, *out=None*, *keepdims=False*) [[source](#)]
(<http://github.com/numpy/numpy/blob/v1.10.1/numpy/core/fromnumeric.py#L1733-L1831>)

Sum of array elements over a given axis.

Parameters: *a* : *array_like*

Elements to sum.

axis : *None or int or tuple of ints, optional*

Axis or axes along which a sum is performed. The default (*axis = None*) is perform a sum over all the dimensions of the input array. *axis* may be negative, in which case it counts from the last to the first axis.

New in version 1.7.0.

If this is a tuple of ints, a sum is performed on multiple axes, instead of a single axis or all the axes as before.

dtype : *dtype, optional*

The type of the returned array and of the accumulator in which the elements are summed. By default, the dtype of *a* is used. An exception is when *a* has an integer type with less precision than the default platform integer. In that case, the default platform integer is used instead.

out : *ndarray, optional*

Array into which the output is placed. By default, a new array is created. If *out* is given, it must be of the appropriate shape (the shape of *a* with *axis* removed, i.e., `numpy.delete(a.shape, axis)`). Its type is preserved. See doc.ufuncs (Section "Output arguments") for more details.

keepdims : *bool, optional*

If this is set to True, the axes which are reduced are left in the result as dimensions with size one. With this option, the result will broadcast correctly against the original *arr*.

Returns: [`sum_along_axis`](#) : *ndarray*

An array with the same shape as *a*, with the specified axis removed. If *a* is a 0-d array, or if *axis* is None, a scalar is returned. If an output array is specified, a reference to *out* is returned.

See also:

`ndarray.sum` ([numpy.ndarray.sum.html#numpy.ndarray.sum](#)) Equivalent method.

`cumsum` ([numpy.cumsum.html#numpy.cumsum](#)) Cumulative sum of array elements.

`trapz` ([numpy.trapz.html#numpy.trapz](#)) Integration of array values using the composite trapezoidal rule.

`mean` ([numpy.mean.html#numpy.mean](#)), `average` ([numpy.average.html#numpy.average](#))

Notes

Arithmetic is modular when using integer types, and no error is raised on overflow.

The sum of an empty array is the neutral element 0:

```
>>> np.sum([])
0.0
```

Examples

```
>>> np.sum([0.5, 1.5])
2.0
>>> np.sum([0.5, 0.7, 0.2, 1.5], dtype=np.int32)
1
>>> np.sum([[0, 1], [0, 5]])
6
>>> np.sum([[0, 1], [0, 5]], axis=0)
array([0, 6])
>>> np.sum([[0, 1], [0, 5]], axis=1)
array([1, 5])
```

If the accumulator is too small, overflow occurs:

```
>>> np.ones(128, dtype=np.int8).sum(dtype=np.int8)
-128
```

Previous topic

[numpy.prod \(numpy.prod.html\)](#)

Next topic

[numpy.nansum \(numpy.nansum.html\)](#)