NumPy all()

Summary: in this tutorial, you'll learn how to use the numpy `all()` function that returns `True` if all elements in an array evaluate `True`.

Introduction to the numpy all() function

The numpy `all()` function returns `True` if all elements in an array (or along a given axis) evaluate to `True`.

The following shows the syntax of the `all()` function:

```
numpy.all(a, axis=None, out=None, keepdims=<no value>, *, where=<no value>)
```

In this syntax, `a` is a numpy array (https://www.pythontutorial.net/python-numpy/create-numpy-array/) or an array-like object e.g., a list (https://www.pythontutorial.net/python-basics/python-list/).

If the input array contains all numbers, the `all()` function returns `True` if all numbers are nonzero or `False` if least one number is zero. The reason is that all non-zero numbers evaluate to `True` while zero evaluates to `False`.

NumPy all() function examples

Let's take some examples of using the `all()` function.

1) Using numpy all() function on 1-D array examples

The following example uses the `all()` function to test whether all numbers in an array are non-zero:

```
import numpy as np

result = np.all([0, 1, 2, 3])
print(result)
```

Output:

```
False
```

The result is `False` because the array has zero at index 0.

```
import numpy as np


result = np.all(np.array([-1, 2, 3]))
print(result)
```

Output:

```
True
```

This example returns `True` because all numbers in the array are nonzero. You can pass an array-like object e.g., a list to the `all()` function. For example:

```
import numpy as np
```

```
result = np.all([-1, 2, 3])
print(result)
```

Output:

```
True
```

2) Using the numpy all() function with a multidimensional array example

The following example uses the `all()` function to test if all elements of a multidimensional array evaluate to `True`:

```
import numpy as np

a = np.array([[0, 1], [2, 3]])
result = np.all(a, axis=0)
print(result)
```

Output:

```
import numpy as np

a = np.array([
    [0, 1],
    [2, 3]
])
result = np.all(a, axis=0)
print(result)
```

Output:

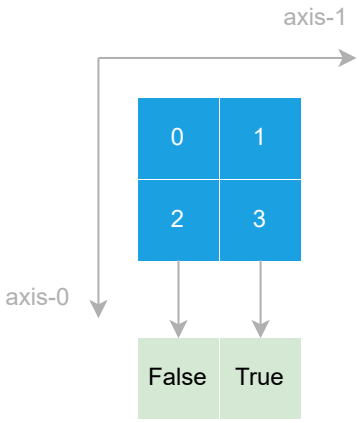
```
False
```

Also, you can evaluate elements along an axis by passing the `axis` argument like this:

```
import numpy as np

a = np.array([
    [0, 1],
    [2, 3]]
)
result = np.all(a, axis=0)
print(result)
```

Output:

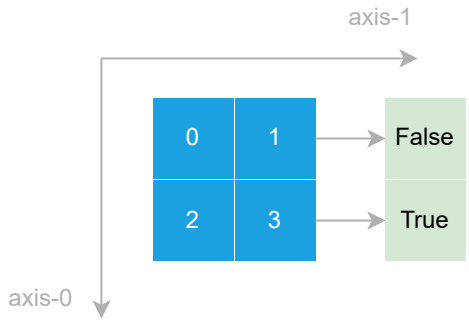



```
[False  True]
```

And axis-1:

```
import numpy as np

a = np.array([
    [0, 1],
    [2, 3]
])
result = np.all(a, axis=1)
print(result)
```



Output:

```
[False  True]
```

Summary

- Use the numpy `all()` function to test whether all elements in an array or along an axis evaluate to `True`.