Python
TUTORIAL

Boolean Indexing

Summary: in this tutorial, you'll learn how to access elements of a numpy array using boolean indexing.

Introduction to numpy array boolean indexing

Numpy allows you to use an array of boolean (https://www.pythontutorial.net/advanced-python/python-bool/) values as an index of another array. Each element of the boolean array indicates whether or not to select the elements from the array.

If the value is `True`, the element of that index is selected. In case the value is `False`, the element of that index is not selected.

The following example uses boolean indexing to select elements of a numpy array using an array of boolean values:

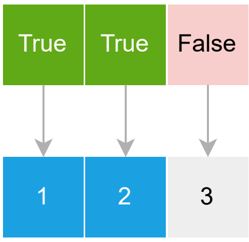
```
import numpy as np


a = np.array([1, 2, 3])
b = np.array([True, True, False])
```

```
c = a[b]
print(c)
```

Output:

```
[1 2]
```

How it works.



First, create a new numpy array (https://www.pythontutorial.net/python-numpy/create-numpy-array/) that includes three numbers from 1 to 3:

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

Second, create another numpy array with three boolean values `True` , `True` , and `False` :

```
b = np.array([True, True, False])
```

Third, use the boolean array `b` as the index of the array `a` and assign the selected elements to the variable `c` :

```
c = a[b]
```

Because the first and second elements of the array `b` are `True` , the `a[b]` returns a new array with the first and second elements of the array `a` .

Typically, you'll use boolean indexing to filter an array. For example:

```
import numpy as np

a = np.arange(1, 10)
```

```
b = a > 5
print(b)

c = a[b]
print(c)
```

Output:

```
[False False False False False  True  True  True  True]
[6 7 8 9]
```

How it works.

First, create an array that has 9 numbers from 1 to 9 using the `arange()` `(https://www.pythontutorial.net/python-numpy/numpy-arange/)` function:

```
a = np.arange(1, 10)
```

Second, create a boolean array from the following expression:

```
b = a > 5
```

This expression compares each element of the array `a` with `5` and returns `True` if it is greater than `5` or `False` otherwise. The variable `b` is an array of boolean values:

```
[False False False False False  True  True  True  True]
```

Third, use the array `b` as the index of array `a` and assign the result to the variable `c`:

```
c = a[b]
```

The array `c` contains only numbers from array `a`, which are greater than `5`.

Summary