



NumPy Array Indexing

If this Python Tutorial saves you hours of work, please **whitelist it in your ad blocker** 🙏 and

Donate Now

(<https://www.pythontutorial.net/donation/>)

to help us ❤️ pay for the web hosting fee and CDN to keep the

website running.

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

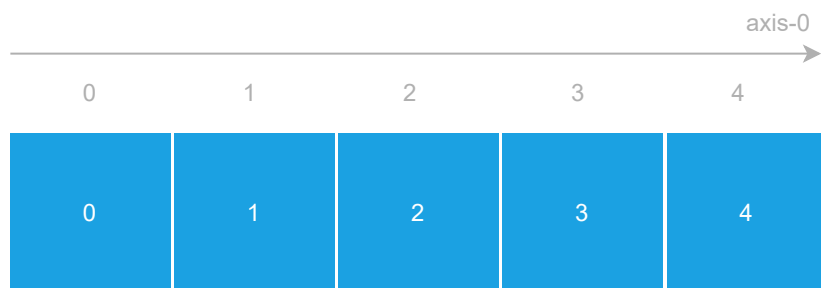
Like a [list](https://www.pythontutorial.net/python-basics/python-list/) (<https://www.pythontutorial.net/python-basics/python-list/>), you can use the square bracket notation (`[]`) to access elements of a [numpy array](https://www.pythontutorial.net/python-numpy/create-numpy-array/) (<https://www.pythontutorial.net/python-numpy/create-numpy-array/>).

NumPy array indexing on 1-D arrays

Along a single axis, you can select elements using indices. The first element starts with index 0, the second element starts with index 1, and so on.

Besides the non-negative indices, you can use negative indices to locate elements. For example, the last element has an index -1, the second last element has an index -2, and so on.

The following example shows how to access elements of a one-dimensional array:



```
import numpy as np

a = np.arange(0, 5)
print(a)

print(a[0])
print(a[1])
print(a[-1])
```

Output:

```
[0 1 2 3 4]
0
```

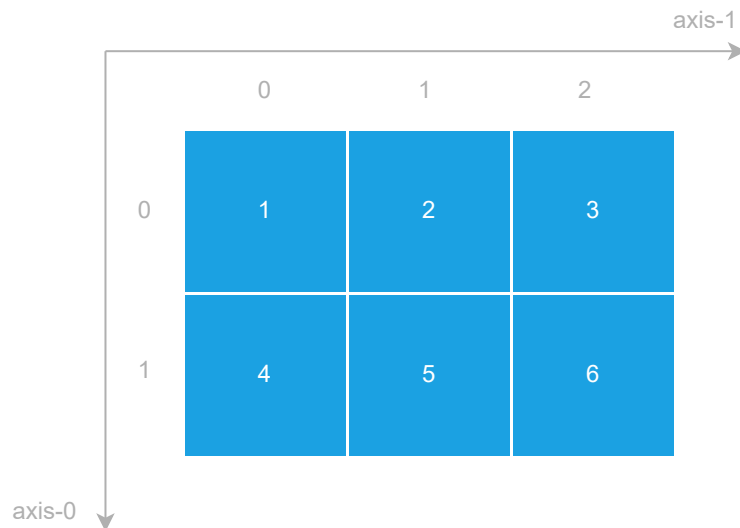
```
1  
4  
3
```

In this example:

- The `a[0]` returns the first element (0)
- The `a[1]` returns the second element (1)
- The `a[-1]` returns the last element (4)
- The `a[-2]` returns the second last element (3)

NumPy array indexing on 2-D arrays

With 2-D and multidimensional arrays, you can select elements as you do with 1-D arrays but for each dimension (or axis). For example:



```
import numpy as np

a = np.array([
    [1, 2, 3],
    [4, 5, 6]
])

print(a.shape)

print(a[0]) # [1 2 3]
print(a[1]) # [4 5 6]

print(a[0, 0]) # 1
print(a[1, 0]) # 4
print(a[0, 2]) # 3
print(a[1, 2]) # 6
print(a[0, -1]) # 3
print(a[1, -1]) # 6
```

Output:

```
(2, 3)
[1 2 3]
[4 5 6]
```

```
1  
4  
3  
6  
3  
6
```

In this example, the numpy array `a` has the shape `(2,3)` therefore it has two axes:

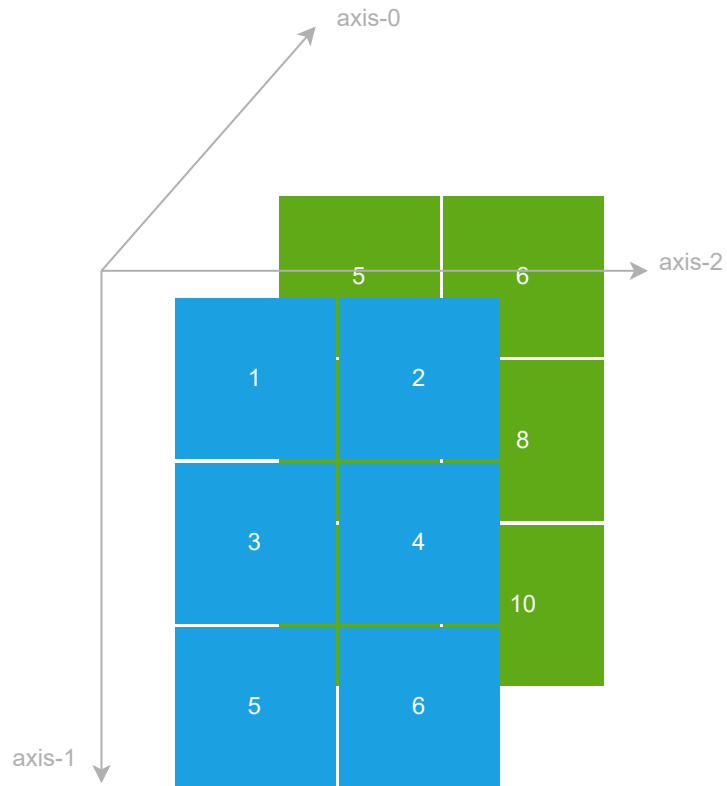
- The first axis has 2 elements (2 lists)
- The second axis has three elements (3 numbers)

The following explains how the array indexing works:

- The `a[0]` returns the first element of the first axis, which is `[1 2 3]`.
- The `a[1]` returns the second element of the first axis, which is `[4 5 6]`
- The `a[0, 0] = 1`: 0 selects the first element of the first axis (`[1 2 3]`) and the first element of the second axis.
- The `a[1, 0] = 4`: 1 selects the second element of the first axis (`[4 5 6]`) and 0 selects the first element of the second axis.
- The `a[0, 2] = 3`: 0 selects the first element of the first axis (`[1 2 3]`) axis and 2 selects the third element of the second axis.
- The `a[1, 2] = 6`: 1 selects the second element of the first axis (`[4 5 6]`) and 2 selects the third element of the second axis.
- The `a[0, -1] = 3`: 0 selects the first element of the first axis (`[1 2 3]`) and -1 selects the last element of the second axis.
- The `a[1, -1] = 6`: 1 selects the second element of the first axis (`[1 2 3]`) and -1 selects the last element of the second axis.

NumPy array indexing on 3-D arrays

The following example creates a 3-D numpy array:



```
import numpy as np

a = np.array([
    [[1, 2], [3, 4], [5, 6]],
    [[5, 6], [7, 8], [9, 10]]
])
```

```
] )  
  
print(a.shape)
```

Output:

```
(2, 3, 2)
```

The array has three axes.

- The first axis has 2 elements (2 lists of lists of numbers)
- The second axis has 3 elements (3 lists of numbers)
- The third axis has 2 elements (2 numbers)

For example:

```
import numpy as np  
  
a = np.array([  
    [[1, 2], [3, 4], [5, 6]],  
    [[5, 6], [7, 8], [9, 10]],  
])  
  
print(a[0, 0, 1]) # 2
```

The following expression returns 2:

```
a[0,0,1]
```

The first number 0 selects the first element of the first axis so it returns:

```
[[1, 2], [3, 4], [5, 6]]
```

The second number 0 selects the first element of the second axis so it returns:

```
[1, 2]
```



The third number (1) selects the second element of the third axis which returns 2.

Summary

- Use square bracket notation [] with an index to access elements of a numpy array.
- Use zero and positive indexes to start selecting from the beginning of the array.
- Use negative indexes to start selecting from the end of the array.