

How can I use NumPy's arange function to create an array that includes the endpoint?

Authored by
stats writer

June 24, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I use NumPy's arange function to create an array that includes the endpoint?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=150831>

NumPy's arange function is a useful tool for creating arrays with evenly spaced values. It allows you to specify the start, stop, and step size of the array. By default, the endpoint is not included in the array. However, if you want to include the endpoint, you can simply add 1 to the stop value. This will create an array that includes the endpoint as the last element. For example, `arange(0, 10, 2)` will create an array with values 0, 2, 4, 6, 8. But if you want to include the endpoint, you can use `arange(0, 11, 2)` which will create an array with values 0, 2, 4, 6, 8, 10. This can be useful in various applications where you need the endpoint to be included in the array.

NumPy: Use arange and Include Endpoint

The NumPy arange function can be used to create a sequence of values.

By default, this function doesn't include the endpoint as part of the sequence of values.

There are two ways to get around this:

Method 1: Add the Step Size to the Endpoint

```
np.arange(start, stop + step, step)
```

Method 2: Use the linspace Function Instead

```
np.linspace(start, stop, num)
```

The following examples show how to use each method in practice.

Example 1: Add Step Size to the Endpoint

Suppose we would like to create a sequence of values ranging from 0 to 50 with a step size of 5.

If we use the NumPy arange function, the endpoint of 50 will not be included in the sequence by default:

```
import numpy as np

#specify start, stop, and step size
start = 0
stop = 50
step = 5

#create array
np.arange(start, stop, step)

array()
```

To include the endpoint of 50, we can simply add the step size to the stop argument:

```
import numpy as np

#specify start, stop, and step size
```

```
start = 0
stop = 50
step = 5

#create array
np.arange(start, stop + step, step)

array()
```

Notice that the endpoint of 50 is now included in the sequence of values.

Note: You can find the complete documentation for the NumPy `arange()` function .

Example 2: Use the linspace Function Instead

The following code shows how to use this function to create a sequence of values ranging from 0 to 50:

```
import numpy as np

#specify start, stop, and number of total values in
sequence
start = 0
stop = 50
```

```
num = 11
```

```
#create array
```

```
np.linspace(start, stop, num)
```

```
array()
```

Notice that the endpoint of 50 is included in the sequence of values by default.

Note: You can find the complete documentation for the NumPy arange() function .

The following tutorials explain how to perform other common operations in NumPy: