NumPy arange()

Summary: in this tutorial, you'll learn how to use the numpy `arange()` function to create a numpy array with evenly spaced numbers.

Introduction to the numpy arange() function

The numpy `arange()` function creates a new numpy array (https://www.pythontutorial.net/python-numpy/create-numpy-array/) with evenly spaced numbers between `start` (inclusive) and `stop` (exclusive) with a given `step`:

```
numpy.arange(start, stop, step, dtype=None, *, like=None)
```

For example, the following uses `arange()` function to create a numpy array:

```
import numpy as np

a = np.arange(1, 10, 2)

print(a)
```

Output:

```
[1 3 5 7 9]
```

The numpy array starts at 1 and ends at 9. Note that it doesn't include the stop value (10). Because the step is 2, the numpy array contains 1, 3, 5, 7, and 9.

Because we pass 1 and 10 as integers, the `arange()` function creates a new array of integers.

If you want to create an array of floats (https://www.pythontutorial.net/advanced-python/python-float/) , you can pass the start and stop values as floats like this:

```
import numpy as np

a = np.arange(1.0, 10.0, 2)

print(a)
```

Output:

```
[1. 3. 5. 7. 9.]
```

Or you can explicitly specify the type of the numpy array's elements using the `dtype` argument:

```
import numpy as np

a = np.arange(1, 10, 2, dtype=np.float64)

print(a)
```

Output:

```
[1. 3. 5. 7. 9.]
```

Summary

- Use numpy `arange()` function to create a new numpy array with evenly spaced numbers between `start` (inclusive) and `stop` (exclusive) with a given interval.