NumPy Array Creation

In this lesson, array creation using NumPy is explained.

We'll cover the following

If you don't know what an array is, please read this short article.

Creating an array

`NumPy` has the power to create multi-dimensional arrays beyond 2 dimensions, but for this course, we focus on 1-D and 2-D array creation.

1-D array

The following program creates a 1-D array:

```
import numpy as np
# Creating 1-D array
n_array = np.array([1,2,3,4,5])
print(n_array)
```

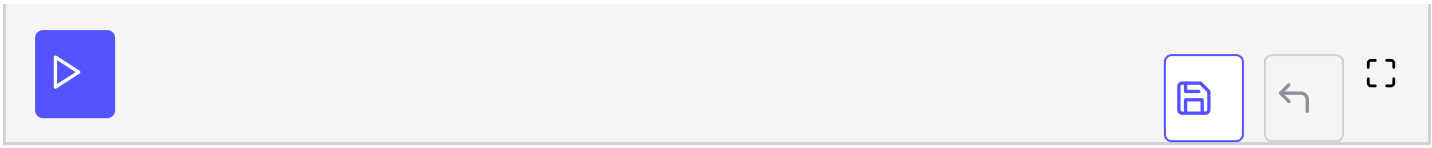
A single `list` of numbers is passed to the `np.array` function to create a 1-D `NumPy` array.

2-D array

The following program creates a 2-D array:

```
import numpy as np
# Creating 2-D array
n_array = np.array([ [1,2,3,4,5],[5,4,3,2,1] ])
```

```
print(n_array)
```



A 2-D array is made by passing a 2-D `list` to the `np.array` function. A 2-D `list` is created by enclosing 2 or more lists as elements of a single `list`.

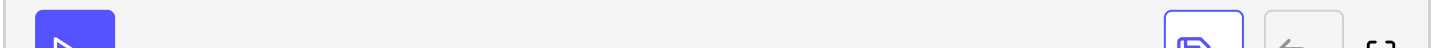
Important functions

The following is a list of some more functions that are frequently used to create `NumPy` arrays.

- `np.empty(int n)`: Generates an empty 1-D array of size depending on the parameter `n`. Initially, this function fills the `n` size array with garbage values that can be overwritten later
- `np.ones(int n)`: Generates a 1-D array of only 1's having length `n`
- `np.zeros(int n)`: Generates a 1-D array of only 0's having length `n`
- `np.eye(int n)`: Generates a square identity matrix or a 2-D array of rows and columns having order `n`
- `np.arange(int start, int stop, int step)`: Generates an array containing elements from the `start` parameter to the `end` parameter (exclusive) with the difference of an amount equal to the `step` parameter

All the above functions are used below to generate `NumPy` arrays.

```
import numpy as np

print("An Empty Array of size 5")
print(np.empty(5)) # Generate empty array of size 5

print("An Array of only 1's of size 5")
print(np.ones(5)) # Generate array containing only 1's of size 5

print("An Array of only 0's of size 5")
print(np.zeros(5)) # Generate array containing only 0's of size 5

print("An Identity Matrix Array of size 5")
print(np.eye(5)) # Generate identity matrix array of size 5

print("Array populated with numbers from 2 to 50 with the difference of 3")
print(np.arange(2, 50, 3)) # Generate array with consecutive numbers
```



It can be observed from the output that all functions generate arrays according to the above explanations.

Also, in the `arange()` function, notice that the `end` value is not included in the resultant array.

In the next lesson, array indexing is explained.