Transposing of NumPy Array

In this lesson, transposing of NumPy arrays are explained.

We'll cover the following

- Transpose
- More 'NumPy' array functions

Transpose

As the name suggests, we will take the transpose of a given 2-D `NumPy` array. Just like matrix transpose in linear algebra, we convert the rows of a 2-D NumPy array into columns and convert its columns into rows. The function `transpose()` or simply `T` can be used to take the transpose of a 2-D array.

```
import numpy as np
# Creating 2-D array
arr = np.arange(0,50,1).reshape(10,5) # Declare a 2-D array

print("The original array")
print(arr)

print("\nThe transposed array")
print(arr.transpose()) # Print the transposed array
#print(arr.T) # This can also be used and same result will be produced
```

On **line 3**, a 2-D array is declared using the `arange` and `reshape` functions. The `arange` function generates the required array elements, and the `reshape` function arranges them in the form of a 2-D matrix depending upon the parameters. The **1st** parameter of the `reshape` function is the number of rows, and the **2nd** parameter is the number of columns.

> **Note**: The generated elements must be equal to the product of the row and column parameters; otherwise, the `reshape` function produces an error.

More ‘NumPy’ array functions

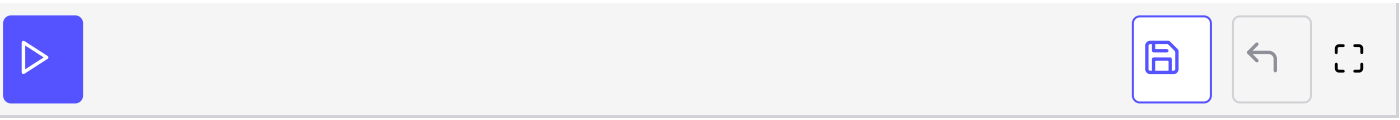
Function	Description
`np.exp(arr)`	Takes the exponent of every element of the array
`np.square(arr)`	Takes the square of every element of the array
`np.sqrt(arr)`	Takes the square root of every element of the array
`np.cbrt(arr)`	Takes the cube root of every element of the array
`np.add(arr1, arr2)`	Adds the corresponding elements of the two arrays
`np.subtract(arr1, arr2)`	Subtracts the corresponding elements of the two arrays

> **Note**: The `add` and `sub` functions only give results if the shape or dimensions of both of the input arrays are the same.

The following example makes use of all these functions.

```
import numpy as np
# Declare 2 array
arr1 = np.arange(1,40,4)
arr2 = np.arange(1,30,3)

print("The first array\n", arr1, "\nThe second array\n", arr2, '\n')

print("The Exponent Function")
print(np.exp(arr1))

print("\nThe Square Function")
print(np.square(arr1))

print("\nThe Square root Function")
```

```
print("\nThe Square root Function")
print(np.sqrt(arr1))


print("\nThe Cube root Function")
print(np.cbrt(arr1))

print("\nThe Addition Function")
print(np.add(arr1, arr2))

print("\nThe Subtraction Function")
print(np.subtract(arr1, arr2))
```



More functions like these can be found here. The link points to the `scipy` package, which contains universal functions for n-dimensional arrays.

In the next lesson, some array processing techniques are discussed.