

Experiment No : 07

Aim : Write python program to understand basic Array operation on 1D and multidimensional array using Numpy.

Description :

Python: Operations on Numpy Arrays.

NumPy is a Python package which means 'Numerical Python'. It is the library for logical computing, which contains a powerful n-dimensional array object, gives tools to integrate C, C++ and so on.

It is likewise helpful in linear based math, arbitrary number capacity and so on. NumPy exhibits can likewise be utilized as an effective multi-dimensional compartment for generic data.

NumPy Array: Numpy array is a powerful N-dimensional array object which is in the form of rows and columns. We can initialize NumPy arrays from nested Python lists and access it elements. A Numpy array on a structural level is made up of a combination of:

The Data pointer indicates the memory address of the first byte in the array. The Data type or dtype pointer describes the kind of elements that are contained within the array.

The shape indicates the shape of the array. The strides are the number of bytes that should be skipped in memory to go to the next element. Operations on Numpy Array.

Implementation :

Code :

```
import numpy as np

# Initializing the array
arr1 = np.arange(4, dtype = np.float_).reshape(2, 2)
print('First array:')
print(arr1)
print('\nSecond array:')
arr2 = np.array([12, 12])
print(arr2)
print('\nAdding the two arrays:')
print(np.add(arr1, arr2))
print('\nSubtracting the two arrays:')
print(np.subtract(arr1, arr2))
print('\nMultiplying the two arrays:')
print(np.multiply(arr1, arr2))
print('\nDividing the two arrays:')
print(np.divide(arr1, arr2))

#Example:

# Python code to perform reciprocal operation
# on NumPy array
import numpy as np
```

```
arr = np.array([25, 1.33, 1, 1, 100])
print('Our array is:')
print(arr)
print('\nAfter applying reciprocal function:')
print(np.reciprocal(arr))
arr2 = np.array([25], dtype = int)
print('\nThe second array is:')
print(arr2)
print('\nAfter applying reciprocal function:')
print(np.reciprocal(arr2))
# Python code to perform power operation
# on NumPy array
import numpy as np
arr = np.array([5, 10, 15])
print('First array is:')
print(arr)
print('\nApplying power function:')
print(np.power(arr, 2))
print('\nSecond array is:')
arr1 = np.array([1, 2, 3])
print(arr1)
```

```
print('\nApplying power function again:')
```

```
print(np.power(arr, arr1))
```

```
"""
```

This function returns the remainder of division of the corresponding elements

in the input array. The function `numpy.remainder()` also produces the same result.

```
# Python code to perform mod function
```

```
# on NumPy array
```

```
"""
```

```
import numpy as np
```

```
arr = np.array([5, 15, 20])
```

```
arr1 = np.array([2, 5, 9])
```

```
print('First array:')
```

```
print(arr)
```

```
print("\nSecond array:")
```

```
print(arr1)
```

```
print("\nApplying mod() function:")
```

```
print(np.mod(arr, arr1))
```

```
print("\nApplying remainder() function:")
```

```
print(np.remainder(arr, arr1))
```

Output :

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\LENOVO> python -u "c:\Users\LENOVO\Desktop\python exps\exp7.py"
First array:
[[0. 1.]
 [2. 3.]]

Second array:
[12 12]

Adding the two arrays:
[[12. 13.]
 [14. 15.]]

Subtracting the two arrays:
[[-12. -11.]
 [-10. -9.]]

Multiplying the two arrays:
[[ 0. 12.]
 [24. 36.]]

Dividing the two arrays:
[[0.         0.08333333]
 [0.16666667 0.25      ]]
Our array is:
[ 25.    1.33  1.    1.  100. ]

After applying reciprocal function:
[0.04    0.7518797 1.    1.    0.01    ]
```

```
After applying reciprocal function:
[0.04      0.7518797 1.         1.         0.01      ]

The second array is:
[25]

After applying reciprocal function:
[0]
First array is:
[ 5 10 15]

Applying power function:
[ 25 100 225]

Second array is:
[1 2 3]

Applying power function again:
[  5 100 3375]
First array:
[ 5 15 20]

Second array:
[2 5 9]

Applying mod() function:
[1 0 2]

Applying remainder() function:
[1 0 2]
PS C:\Users\LENOVO> █
```

Conclusion : Therefore we have successfully implemented python program to understand basic Array operation on 1D and multidimensional array using Numpy.