

Numpy Practical list

1. Create an array using Numpy.
2. Create more than one dimensions array using Numpy.
3. Create minimum dimensions array using Numpy.
4. Check the data type of following array using Numpy
`type1 = np.array([1, 2, 3, 4, 5, 6])`
`type2 = np.array([1.5, 2.5, 0.5, 6])`
`type3 = np.array(['a', 'b', 'c'])`
`type4 = np.array(["Canada", "Australia"], dtype='U5')`
`type5 = np.array([555, 666], dtype=float)`
5. Check the following array shape using Numpy
`array1d = np.array([1, 2, 3, 4, 5, 6])`
`array2d = np.array([[1, 2, 3], [4, 5, 6]])`
`array3d = np.array([[[1, 2, 3], [4, 5, 6]], [[7, 8, 9], [10, 11, 12]]])`
6. Use the ndim method to determine the dimension of NumPy array.
7. Use the resize and reshape method on Numpy array.
8. Create the Program to Transform List or Tuple into NumPy array.
9. Perform the following Indexing Operations using Numpy array.
`array1d = np.array([1, 2, 3, 4, 5, 6])`
 1. Get first value
 2. Get last value
 3. Get 4th value from first
 4. Get 5th value from last
 5. Get multiple values
10. Perform the following Indexing Operations using Numpy array.
`array2d = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])`
 1. Get first row first col
 2. Get first row second col
 3. Get first row second col
 4. Get second row second col
11. Perform the following Indexing Operations using Numpy array.
`array3d = np.array([[[1, 2, 3], [4, 5, 6]], [[7, 8, 9], [10, 11, 12]]])`
12. Perform the following Single Dimensional Slicing Operations using Numpy array.
`array1d = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])`
 1. from index 4 to last index
 2. From index 0 to 4 index
 3. From index 4(included) up to index 7(excluded)
 4. Excluded last element
 5. Up to second last index(negative index)
 6. From last to first in reverse order(negative step)

7. All odd numbers in reversed order
8. All even numbers in reversed order
9. All elements

13. Perform the following Multidimensional Dimensional Slicing Operations using Numpy array.

```
array2d = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
```

1. 2nd and 3rd col
2. 2nd and 3rd row
3. Reverse an array

14. Perform the following operations to Manipulating the Dimensions and the Shape of Arrays(Flips the order of the Axes)

```
array2d = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
```

1. Permute the dimensions of an array
2. Flip array in the left/right direction
3. Flip array in the up/down direction
4. Rotate an array by 90 degrees in the plane specified by axes

15. Perform the following operations to Manipulating the Dimensions and the Shape of Arrays(Joining and Stacking)

```
array1 = np.array([[1, 2, 3], [4, 5, 6]])
```

```
array2 = np.array([[7, 8, 9], [10, 11, 12]])
```

1. Stack arrays in sequence horizontally (column wise).
2. Stack arrays in sequence vertically (row wise)
3. Stack arrays in sequence depth wise (along third axis)
4. Appending arrays after each other, along a given axis
5. Append values to the end of an array

16. Perform the following Arithmetic Operations using Numpy Array.

```
array1 = np.array([[1, 2, 3], [4, 5, 6]])
```

```
array2 = np.array([[7, 8, 9], [10, 11, 12]])
```

1. array1 + array2
2. array1 - array2
3. array1 * array2
4. array2 / array1
5. array1 ** array2

17. Perform the following Scalar Arithmetic Operations using Numpy Array.

```
array1 = np.array([[10, 20, 30], [40, 50, 60]])
```

1. array1 + 2
2. array1 - 5
3. array1 * 2
4. array1 / 5
5. array1 ** 2

18. Perform the following Elementary Mathematical Functions using Numpy Array.

```
array1 = np.array([[10, 20, 30], [40, 50, 60]])
```

1. `sin(array1)`
2. `cos(array1)`
3. `tan(array1)`
4. `sqrt(array1)`
5. `exp(array1)`
6. `log10(array1)`

19. Perform the following Element-wise Mathematical Operations using Numpy Array.

```
array1 = np.array([[10, 20, 30], [40, 50, 60]])
```

```
array2 = np.array([[2, 3, 4], [4, 6, 8]])
```

```
array3 = np.array([[-2, 3.5, -4], [4.05, -6, 8]])
```

1. Addition of array1 and array2
2. Multiplication of array1 and array2
3. Power of array1 and array2

20. Perform the following Aggregate and Statistical Functions using Numpy Array.

```
array1 = np.array([[10, 20, 30], [40, 50, 60]])
```

1. Mean
2. Standard deviation
3. Variance
4. Sum of array elements
5. Product of array elements

21. Use the Where(), Select() and Choose() function to identify the element is less than 4, mul by 2 else by 3.

```
np.array([[1, 2, 3], [4, 5, 6]])
```

22. Perform the following Logical Operations using Numpy Array.

```
thearray = np.array([[10, 20, 30], [14, 24, 36]])
```

1. `logical_or(Condition array<10, array>15)`
2. `logical_and(Condition array<10, array>15)`
3. `logical_not(Condition array<20)`

23. Perform the following Standard Set Operations using Numpy Array.

```
array1 = np.array([[10, 20, 30], [14, 24, 36]])
```

```
array2 = np.array([[20, 40, 50], [24, 34, 46]])
```

1. Find the union of two arrays
2. Find the intersection of two arrays
3. Find the set difference of two arrays

Answers:-

<https://www.pythonprogramming.in/numpy-tutorial-with-examples-and-solutions.html>