Numpy

1. Example:

```
# Python Program to create array with all zeros

import numpy as np

a = np.zeros(3, dtype = int)

print("Matrix a : \n", a)

b = np.zeros([3, 3], dtype = int)

print("\nMatrix b : \n", b)
```

2. Example:

```
import numpy as np
a = np.array([[1,2,3],
              [4,5,6]])

b = np.array([[10,11,12],
              [13,14,15]])

c = a + b

print(c)
```

```
   3. Example-3

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

   b = 2*a # multiplying the numpy array a(matrix) by 2

   print(b)
```

Example 4: Identity Matrix

```
   i = np.eye(4)
   print(i)
```

Example 5: Array re-dimensioning

```
   a = np.array([x for x in range(27)])
   o = a.reshape((3,3,3))
   print(o)
```

Example 6: Array datatype conversion

```
   a = np.array([[2.5, 3.8, 1.5],
                 [4.7, 2.9, 1.56]])

   o = a.astype('int')
   print(o)
```

example 7: Boolean array:

```
   a = np.array([[1, 0, 0],
                 [1, 1, 1],
                 [0, 0, 0]])

   o = a.astype('bool')

   print(o)
```

EXERCISE 8 - Horizontal Stacking of Numpy Arrays

Stack 2 numpy arrays horizontally i.e., 2 arrays having the same 1st dimension (number of rows in 2D arrays)

```
a1 = np.array([[1,2,3],
               [4,5,6]])

a2 = np.array([[7,8,9],
               [10,11,12]])

o = np.hstack((a1, a2))

print(o)
```

EXERCISE 9 - Vertically Stacking of Numpy Arrays

Stack 2 numpy arrays vertically i.e., 2 arrays having the same last dimension (number of columns in 2D arrays)

Sample Solution

```
a1 = np.array([[1,2],
               [3,4],
               [5,6]])

a2 = np.array([[7,8],
               [9,10],
               [10,11]])

o = np.vstack((a1, a2))

print(o)
```

EXERCISE 9 - Custom Sequence Generation

Generate a sequence of numbers in the form of a numpy array from 0 to 100 with gaps of 2 numbers, for example: 0, 2, 4 ....

Sample Solution

```
list_of_numbers = [x for x in range(0, 101, 2)]

o = np.array(list_of_numbers)

print(o)
```

Alternative Solution

```
o = np.arange(0, 101, 2)

print(o)
```

EXERCISE 10 - Getting the positions (indexes) where elements of 2 numpy arrays match

From 2 numpy arrays, extract the indexes in which the elements in the 2 arrays match

Sample Solution

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

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

```
print(np.where(a == b))
```

EXERCISE 11 - Generation of given count of equally spaced numbers within a specified range

Output a sequence of equally gapped 5 numbers in the range 0 to 100 (both inclusive)

Sample Solution

```
o = np.linspace(0, 100, 5)

print(o)
```

EXERCISE 12 - Matrix Generation with one particular value

Output a matrix (numpy array) of dimension 2-by-3 with each and every value equal to 5

Sample Solution

```
o = np.full((2, 3), 5)

print(o)
```

Alternate Solution

```
a = np.ones((2, 3))

o = 5*a

print(o)
```

EXERCISE 13 - Array Generation by repeatition of a small array across each dimension

Output an array by repeating a smaller array of 2 dimensions, 10 times

Sample Solution

```
a = np.array([[1,2,3],
              [4,5,6]])
o = np.tile(a, 10)

print(o)
```

```
example 14: slicing

arr = np.arange(9).reshape(3,3)
print('Original array')
arr


print("\nModified array")
arr[:, ::-1]
```

self learn exercise:

1. Python Program to find sum of array
2. Python Program to find largest element in an one dimensional array.
3. **Python code to extract all numbers between a given range from a numpy array**

Hint:

```
# Question: Get all items between 5 and 10 from a.

# Input: a = np.array([2, 6, 1, 9, 10, 3, 27])
# Output: (array([6, 9, 10]),)
```

4. Write a function rotate(arr[], d, n) that rotates arr[] of size n by d elements. In this article, we will explore the Reversal Algorithm for array rotation and implement it in Python.

Example

```
Input:  arr[] = [1, 2, 3, 4, 5, 6, 7]
          d = 2
Output: arr[] = [3, 4, 5, 6, 7, 1, 2]
```

5. Python Program to Split the array and add the first part to the end.
6. Write a Python program for a given multiple numbers and a number **n**, the task is to print the remainder after multiplying all the numbers divided by **n.**

Example Explanation:

Input: arr[] = {100, 10, 5, 25, 35, 14},
n = 11
Output: 9
Explanation: 100 x 10 x 5 x 25 x 35 x 14 = 61250000 % 11 = 9

Input : arr[] = {100, 10},
n = 5
Output : 0
Explanation: 100 x 10 = 1000 % 5 = 0

7. Python Code for given 2 numpy arrays as matrices, output the result of multiplying the 2 matrices (as a numpy array)
8. Python Program for matrix transpose
9. Write a python code to swap the column of matrix

Original array

```
array([[0, 1, 2],
       [3, 4, 5],
       [6, 7, 8]])
```

Modified array

```
array([[1, 0, 2],
       [4, 3, 5],
       [7, 6, 8]])
```

Solution:

```
arr = np.arange(9).reshape(3,3)

print('Original array')
arr

# Solution

print("\nModified array")
arr[:, [1,0,2]]
```

10. Write a python code to swap the row of matrix.

Original array

```
array([[0, 1, 2],
       [3, 4, 5],
       [6, 7, 8]])
```

Modified array

```
array([[3, 4, 5],
       [0, 1, 2],
       [6, 7, 8]])
```

Solution:

```
arr = np.arange(9).reshape(3,3)
print('Original array')
arr

# Solution

print("\nModified array")
arr[[1,0,2], :]
```

1. Python program to add two Matrices
2. Python program to multiply two matrices
3. Python program for Matrix Product
4. Adding and Subtracting Matrices in Python
5. Transpose a matrix in Single line in Python
6. Python | Matrix creation of n*n

1. Write a NumPy program to convert a list of numeric values into a one-dimensional NumPy array.
   Expected Output:
   ```
   Original List: [12.23, 13.32, 100, 36.32]
   One-dimensional NumPy array: [ 12.23 13.32 100. 36.32]
   ```
2. Write a NumPy program to create a 3x3 matrix with values ranging from 2 to 10.
   Expected Output:
   ```
   [[ 2 3 4]
   [ 5 6 7]
   [ 8 9 10]]
   ```
3. Write a NumPy program to create a null vector of size 10 and update the sixth value to 11.
   ```
   [ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
   Update sixth value to 11
   [ 0. 0. 0. 0. 0. 0. 11. 0. 0. 0.]
   ```
4. Write a NumPy program to create an array with values ranging from 12 to 38.
   Expected Output:
   ```
   [12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37]
   ```
5. Write a NumPy program to reverse an array (the first element becomes the last).
   ```
   Original array:
   [12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37]
   Reverse array:
   [37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12]
   ```
6. Write a NumPy program to convert an array to a floating type.
   Sample output:
   ```
   Original array
   [1, 2, 3, 4]
   Array converted to a float type:
   [ 1. 2. 3. 4.]
   ```
7. Write a NumPy program to create an 8x8 matrix and fill it with a checkerboard pattern.
   ```
   Checkerboard pattern:
   [[0 1 0 1 0 1 0 1]
   ..........
   ```

```
[0 1 0 1 0 1 0 1]
[1 0 1 0 1 0 1 0]]
```

8. Write a NumPy program to convert a list and tuple into arrays.
   ```
   List to array:
   [1 2 3 4 5 6 7 8]
   Tuple to array:
   [[8 4 6]
   [1 2 3]]
   ```
9. Write a NumPy program to append values to the end of an array.
   Expected Output:
   ```
   Original array:
   [10, 20, 30]
   After append values to the end of the array:
   [10 20 30 40 50 60 70 80 90]
   ```
10. Write a NumPy program to test whether each element of a 1-D array is also present in a second array.
    Expected Output:
    ```
    Array1: [ 0 10 20 40 60]
    Array2: [0, 40]
    Compare each element of array1 and array2
    [ True False False True False]
    ```
11. Write a NumPy program to find common values between two arrays.
    Expected Output:
    ```
    Array1: [ 0 10 20 40 60]
    Array2: [10, 30, 40]
    Common values between two arrays:
    [10 40]
    ```
12. Write a NumPy program to get the unique elements of an array.
    Expected Output:
    ```
    Original array:
    [10 10 20 20 30 30]
    Unique elements of the above array:
    [10 20 30]
    Original array:
    [[1 1]
    [2 3]]
    Unique elements of the above array:
    [1 2 3]
    ```
13. Write a NumPy program to find the set difference between two arrays. The set difference will return sorted, distinct values in array1 that are not in array2.
    Expected Output:
    ```
    Array1: [ 0 10 20 40 60 80]
    Array2: [10, 30, 40, 50, 70, 90]
    Set difference between two arrays:
    [ 0 20 60 80]
    ```
14. Write a NumPy program to find the set exclusive-or of two arrays. Set exclusive-or will return sorted, distinct values in only one (not both) of the input arrays.
    ```
    Array1: [ 0 10 20 40 60 80]
    ```

Array2: [10, 30, 40, 50, 70]
Unique values that are in only one (not both) of the input arrays:
[0 20 30 50 60 70 80]

15. Write a NumPy program to find the union of two arrays. Union will return a unique, sorted array of values in each of the two input arrays.
    Array1: [0 10 20 40 60 80]
    Array2: [10, 30, 40, 50, 70]
    Unique sorted array of values that are in either of the two input arrays:
    [0 10 20 30 40 50 60 70 80]

16. Write a NumPy program to construct an array by repeating.

    Sample array: [1, 2, 3, 4]
    Expected Output:
    Original array
    [1, 2, 3, 4]
    Repeating 2 times
    [1 2 3 4 1 2 3 4]
    Repeating 3 times
    [1 2 3 4 1 2 3 4 1 2 3 4]

17. Write a NumPy program to find the indices of the maximum and minimum values along the given axis of an array.
    Original array: [1 2 3 4 5 6]
    Maximum Values: 5
    Minimum Values: 0
18. Write a NumPy program to change an array's dimension.
    Expected Output:
    6 rows and 0 columns
    (6,)
    (3, 3) -> 3 rows and 3 columns
    [[1 2 3]
    [4 5 6]
    [7 8 9]]
    Change array shape to (3, 3) -> 3 rows and 3 columns
    [[1 2 3]
    [4 5 6]
    [7 8 9]]
19. Write a NumPy program to create another shape from an array without changing its data.
    Reshape 3x2:
    [[1 2]
    [3 4]
    [5 6]]
    Reshape 2x3:
    [[1 2 3]
    [4 5 6]]
20. Write a NumPy program to create a new array of 3*5, filled with 2.
    Expected Output:
    [[2 2 2 2 2]

```
[2 2 2 2 2]
[2 2 2 2 2]]
[[2 2 2 2 2]
[2 2 2 2 2]
[2 2 2 2 2]]
```

21. Write a NumPy program to find the 4th element of a specified array.

    Expected Output:

    ```
    [[ 2 4 6]
    [ 6 8 10]]
    Forth e1ement of the array:
    6
    ```

22.