1. BASIC ARRAY MANIPULATION

Create a 1D array with values ranging from 0 to 9.

In [1]:

```
pip install numpy
```

```
Requirement already satisfied: numpy in c:\users\lenovo\anaconda3\lib\site-package
s (1.24.3)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:

```
import numpy as np


arr = np.arange(10)


print(arr)
```

```
[0 1 2 3 4 5 6 7 8 9]
```

Reshape the array into a 3x3 matrix.

Explore Reshape Function

In [3]:

```
import numpy as np


arr = np.arange(9)


matrix = arr.reshape(3, 3)


print(matrix)
```

```
[[0 1 2]
 [3 4 5]
 [6 7 8]]
```

Access the element at the second row, second column.

In [4]:

```
import numpy as np


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


element = arr[1, 1]


print(element)
```

```
5
```

Perform element-wise addition, subtraction, multiplication, and division on two arrays [1, 2, 3] and [4, 5, 6].

In [5]:

```
import numpy as np
```

```
arr1 = np.array([1, 2, 3])
arr2 = np.array([4, 5, 6])


addition = np.add(arr1, arr2)
print(addition)


subtraction = np.subtract(arr1, arr2)
print(subtraction)


multiplication = np.multiply(arr1, arr2)
print(multiplication)


division = np.divide(arr1, arr2)
print(division)
```

```
[5 7 9]
[-3 -3 -3]
[ 4 10 18]
[0.25 0.4  0.5 ]
```

In [ ]:

Find the sum, mean, and standard deviation of the array [1, 2, 3, 4, 5].

In [6]:

```
import numpy as np


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


sum_arr = np.sum(arr)
print("Sum:", sum_arr)


mean_arr = np.mean(arr)
print("Mean:", mean_arr)


std_arr = np.std(arr)
print("Standard Deviation:", std_arr)
```

```
Sum: 15
Mean: 3.0
Standard Deviation: 1.4142135623730951
```

In [ ]:

Reshape the array [1, 2, 3, 4, 5, 6] into a 2x3 array.

Explore Reshape

In [7]:

```
import numpy as np


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


reshaped_arr = np.reshape(arr, (2, 3))
```

```
print(reshaped_arr)
```

```
[[1 2 3]
 [4 5 6]]
```

2. Indexing and Slicing

Create a 5x5 array with random integers.

In [8]:

```
import numpy as np


arr = np.random.randint(low=0, high=10, size=(5, 5))


print(arr)
```

```
[[0 8 7 3 5]
 [0 0 0 8 5]
 [9 9 0 4 9]
 [9 5 7 4 5]
 [1 9 9 1 0]]
```

Replace all values in the sub-array with a specific value.

In [9]:

```
import numpy as np


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


row, col = np.where(arr == 5)


arr[row, col] = 10


print(arr)
```

```
[[ 1  2  3]
 [ 4 10  6]
 [ 7  8  9]]
```

Extract the subarray [[3, 4], [7, 8]] from the array [[1, 2], [3, 4], [5, 6], [7, 8]].

In [10]:

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


sub_arr = arr[1:3, :]


print(sub_arr)
```

```
[[3 4]
 [5 6]]
```

3. Broadcasting:

Create a 2D array of shape (3, 3) with values from 0 to 2.

In [11]:

```
import numpy as np


arr = np.zeros((3, 3), dtype=int)
arr += np.arange(3)[:, np.newaxis]


print(arr)
```

```
[[0 0 0]
 [1 1 1]
 [2 2 2]]
```

Add a 1D array of shape (3,) to each row of the 2D array using broadcasting.

In [12]:

```
import numpy as np


arr = np.zeros((3, 3))


row_vec = np.array([1, 2, 3])


arr += row_vec[:, np.newaxis]


print(arr)
```

```
[[1. 1. 1.]
 [2. 2. 2.]
 [3. 3. 3.]]
```

4. Concatenation and Splitting

1. Create two 2D arrays of shape (3, 3) with random integers.
2. Concatenate them horizontally and vertically.
3. Split the concatenated arrays back into the original arrays.

In [13]:

```
import numpy as np


arr1 = np.random.randint(0, 10, size=(3, 3))
arr2 = np.random.randint(0, 10, size=(3, 3))


print("Array 1:")
print(arr1)
print("\nArray 2:")
print(arr2)
```

```
Array 1:
[[2 1 9]
 [3 0 3]
 [4 2 5]]

Array 2:
[[1 7 7]
 [1 5 9]
 [8 6 8]]
```

In [14]:

```
import numpy as np


arr1 = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
arr2 = np.array([[10, 11, 12], [13, 14, 15], [16, 17, 18]])


result_horiz = np.hstack((arr1, arr2))


print("Concatenated horizontally:")
print(result_horiz)
```

```
Concatenated horizontally:
[[ 1  2  3 10 11 12]
 [ 4  5  6 13 14 15]
 [ 7  8  9 16 17 18]]
```

In [15]:

```
import numpy as np

arr1 = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
arr2 = np.array([[10, 11, 12], [13, 14, 15], [16, 17, 18]])

horiz_concat = np.concatenate((arr1, arr2), axis=1)
print(horiz_concat)
```

```
[[ 1  2  3 10 11 12]
 [ 4  5  6 13 14 15]
 [ 7  8  9 16 17 18]]
```

Bonus Question

Find mode of this array = [1,2,1,2,3,4,5,6,8,9,1,2,2,22,21,1,1,1,1,2,5]

Hint: Use count and also loop

In [16]:

```
import collections

arr = [1, 2, 1, 2, 3, 4, 5, 6, 8, 9, 1, 2, 2, 22, 21, 1, 1, 1, 1, 2, 5]


count_dict = collections.Counter(arr)


max_count = max(count_dict.values())


mode = [num for num, count in count_dict.items() if count == max_count]

print("Mode of the array:", mode)
```

```
Mode of the array: [1]
```

```
In [ ]:
```