

Date: 13-01-2025

## NumPy - Indexing & Slicing

**Class Assignment:** Create a NumPy array of 20 elements (from 1 to 20). Using slicing, extract:

1. The first 5 elements.
2. The elements from index 5 to 15.
3. The last 3 elements

```
import numpy as np

array = np.arange(1, 21)
part1 = array[:5]
part2 = array[5:16]
part3 = array[-3:]

print("Array:", array)
print("First 5 elements:", part1)
print("Elements from index 5 to 15:", part2)
print("Last 3 elements:", part3)
```

Array: [ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20]  
First 5 elements: [1 2 3 4 5]  
Elements from index 5 to 15: [ 6 7 8 9 10 11 12 13 14 15 16]  
Last 3 elements: [18 19 20]

## Sorting Arrays in NumPy

**Class Assignment:** Create a NumPy array of 10 random integers. Use np.argsort() to get the indices that would sort the array in ascending order, then print the sorted array using these indices.

```
import numpy as np
random_array = np.random.randint(1, 101, size=10)
sorted_indices = np.argsort(random_array)
sorted_array = random_array[sorted_indices]

print("Original Array:", random_array)
print("Sorted Indices:", sorted_indices)
print("Sorted Array:", sorted_array)
```

Original Array: [67 32 4 69 88 30 79 80 28 74]  
Sorted Indices: [2 8 5 1 0 3 9 6 7 4]  
Sorted Array: [ 4 28 30 32 67 69 74 79 80 88]

## Handling Missing Data in Arrays

**Class Assignment:** Create a 1D NumPy array of 10 integers and randomly replace 2 elements with np.nan. Replace all NaN values in the array with the number 0.

```
import numpy as np

array = np.random.randint(1, 101, size=10)

nan_indices = np.random.choice(array.size, size=2, replace=False)
array = array.astype(float)
array[nan_indices] = np.nan

array[np.isnan(array)] = 0

print("Original Array with NaNs:", array)
print("Array after replacing NaNs with 0:", array)
```

Original Array with NaNs: [ 0. 68. 19. 56. 0. 71. 13. 39. 51. 77.]  
Array after replacing NaNs with 0: [ 0. 68. 19. 56. 0. 71. 13. 39.  
51. 77.]