Numpy Library

PART 1: INTRODUCTION

1.1 What is NumPy?

NumPy (Numerical Python) is a library for working with arrays and mathematical operations. It's faster than regular Python lists.

1.2 How to Import Libraries

Syntax

```
import numpy as np
import pandas as pd
```

PART 2: NUMPY - MOST IMPORTANT FUNCTIONS

1- Array Creation Functions

Syntax

array_name = np.array([elems])

Converts a Python list to a NumPy array.

Example

```
import numpy as np
Arr1 = np.array([1, 2, 3])
print(Arr1)
```

Output

[1 2 3]

Example

```
zero = np.zeros(4)
print(zero)
print('='*50)
ones = np.ones(4)
print(ones)
print('='*50)
```

```
arr_arrange = np.arange(0, 10, 2)
print(arr_arrange)
```

Output

```
[0. 0. 0. 0.]
==================================================
[1. 1. 1. 1.]
==================================================
[0 2 4 6 8]
```

2- Accessing Elements

For 1D Array:
Role: Access element by its index (position)
Syntax:

```
array_name[index]
```

Example:

```
import numpy as np
arr = np.array([10, 20, 30, 40, 50])
print(arr[0])  # First element
print(arr[2])  # Third element
print(arr[4])  # Last element
```

Output:

```
10
30
50
```

3- Looping Through All Elements

Role: Loop through each element one by one
Syntax:

```
for element in array_name:
    print(element)
```

Example:

```
arr = np.array([10, 20, 30, 40, 50])
for element in arr:
    print(element)
print('='*50)
for i in range(len(arr)):
    print(arr[i])
```

Output:

```
10
20
30
40
50
==================================================
10
20
30
40
50
```

4- Adding Elements to a NumPy Array

Syntax

```
new_array_name = np.append(array_name, value)
new_array_name = np.insert(array_name, index, value)
```

Example

1-using append

```
arr = np.array([1, 2, 3])
Arr_app = np.append(arr, 4)
print(Arr_app)
```

Output:

```
[1 2 3 4]
```

2-using insert

```
arr = np.array([1, 2, 4])
arr_inser = np.insert(arr, 2, 3)
print(arr_inser)
```

Output:

```
[1 2 3 4]
```

5- Deleting Elements from a NumPy Array

Array 1D

Syntax

```
new_arr_name = np.delete(array_name, index)
```

Example

```
arr = np.array([10, 20, 30, 40])
Arr_del = np.delete(arr, 1)
print(Arr_del)
```

Output:

[10 30 40]

6- Statistical Functions

- min() – Returns the smallest value in the array.
- max() – Returns the largest value in the array.
- sum() – Returns the sum of all elements in the array.
- mean() – Returns the average of all elements in the array.

Syntax:

```
np.min(arr_name)
np.max(arr_name)
np.sum(arr_name)
np.mean(arr_name)
```

Example

```
import numpy as np
arr = np.array([3, 1, 4, 2, 5])
min_result = np.min(arr)
max_result = np.max(arr)
sum_result = np.sum(arr)
mean_result = np.mean(arr)
print("min =", min_result)
print("max =", max_result)
print("sum =", sum_result)
print("mean =", mean_result)
```

Output:

min = 1
max = 5
sum = 15
mean = 3.0

7- Function sort

Arranges numbers from smallest to largest
Syntax:

```
np.sort(array_name)
np.sort(array_name)[::-1]
```

Example

```
import numpy as np
arr = np.array([5, 2, 8, 1])
new_arr = np.sort(arr)
print(new_arr)
new_arr_des = np.sort(arr)[::-1]
print(new_arr_des)
```

Output:

```
[1 2 5 8]
[8 5 2 1]
```

8- Arithmetic Operations

Syntax

```
Array_name + n
Array_name - n
Array_name * n
Array_name / n
Array_name ** n
```

Operations are applied to each element automatically.

Example

```
import numpy as np
arr = np.array([10, 20, 30])
print(arr + 10)

print(arr - 5)

print(arr * 2)

print(arr ** 2)
```

Output

```
[20 30 40]
[5 15 25]
[20 40 60]
[100 400 900]
```

9- Operations Between Arrays

Syntax

Arr1 + Arr2
Arr1 * Arr2
Arr1 - Arr2

Example

```
arr1 = np.array([10, 100, 1000])
arr2 = np.array([20, 200, 2000])
arr3 = arr1 + arr2
arr4 = arr1 * arr2
print(arr3)
print(arr4)
```

Output

```
[30 300 3000]
[200 20000 2000000]
```