**INTRO TO NUMPY**

**What is NumPy?**

* NumPy is a Python library used for working with arrays.
* It also has functions for working in domain of linear algebra, Fourier transform, and matrices.
* NumPy was created in *2005 by Travis Oliphant*. It is an open-source project and you can use it freely.
* NumPy stands for Numerical Python.

**Why Use NumPy?**

* In Python we have lists that serve the purpose of arrays, but they are slow to process.
* NumPy aims to provide an array object that is up to **50x faster** than traditional Python lists.
* The array object in NumPy is called **ndarray**, it provides a lot of supporting functions that make working with ndarray very easy.
* Arrays are very frequently used in data science, where speed and resources are very important.

**Why is NumPy Faster Than Lists?**

* NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently.
* This behaviour is called locality of reference in computer science.
* This is the main reason why NumPy is faster than lists. Also it is optimized to work with latest CPU architectures.

**Which Language is NumPy written in?**

* NumPy is a Python library and is written partially in Python, but most of the parts that require fast computation are written in C or C++.

**OPERATIONS:**

**#Creating arrays**

arry1 = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]) #1d array

arry2 = np.array([[4, 5, 6], [7, 8, 9]]) #2d array

arry3 = np.array([[[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]]]) #3d array

arry4 = np.array([1, 2, 3, 5, 9], ndmin=4) #4d array

**# Numpy version and data types**

version = np.\_\_version\_\_

print("Numpy Version:", version)

print("Type of Array1:", type(arry1))

**# Checking dimensions**

print("Dimensions of arry1:", arry1.ndim)

**# Accessing elements**

print("Element at index 2 in arr1:", arr1[2])

**# Negative indexing**

print("Last Element from 2nd row in arry2:", arry2[1, -1]) # prints from last

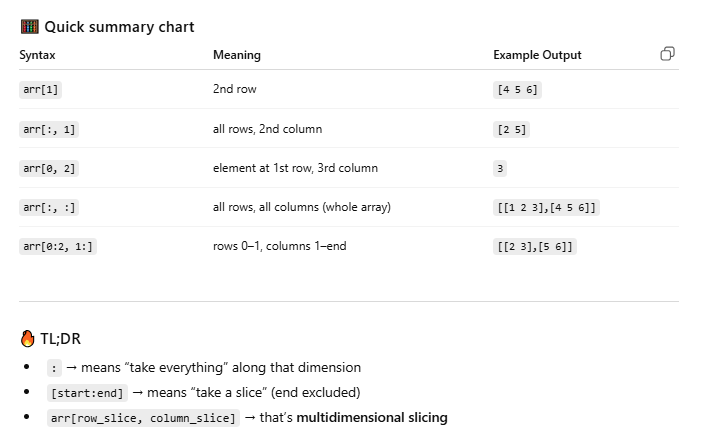
**# Slicing arrays**

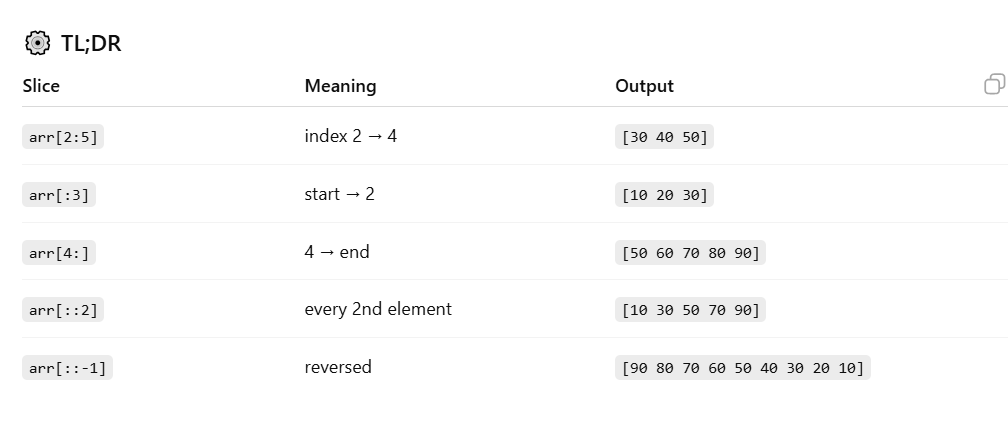
print("Slicing Arrays:")

sliced\_array = arry1[2:5]

print("Original Array:", arry1)

print("Sliced Elements from arry1 (index 2 to 4):", sliced\_array)





**OPERATIONS ON NUMPY**

**# Finding datatype:**

arr=np.array(['apple','banana','pineapple','grapes']) # >U9

arr1=np.array([1.2,2.31,4.56,3.1456]) #float64

arr2=np.array([1,2,6,5,8]) #int64

arr3=np.array(['a','b','c','d','e','f']) # >U1

**# changing from one data type to another**

newarr=arr1.astype('int') #float originally

print("og array:",arr1)

print(arr1.dtype) #converted to int

print("as int:",newarr)

print(newarr.dtype)

**# View and copy**

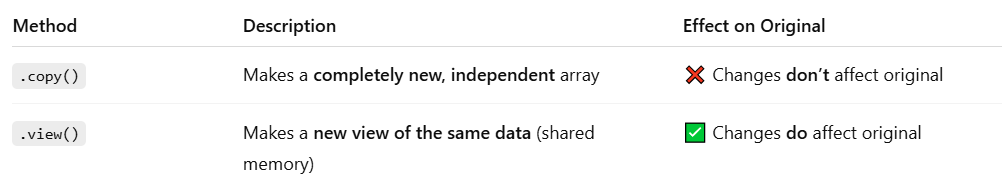
arr2\_copy=arr2.copy()

arr2\_view=arr2.view()

arr2[2]=3

print(arr2\_copy)

print(arr2\_view)



**# Shaping and reshaping**

* “Shape” in NumPy tells you **how many dimensions** your array has and **how many elements are in each**.

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

print(arr.shape) # (2, 3)

* Reshaping means changing the structure of the array without changing its data.

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

reshaped = arr.reshape(2, 3)

print(reshaped)

op:

[[1 2 3]

[4 5 6]]

**# Splitting**

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

print(arra)

split=np.array\_split(arra,2)

for half in split:

print(half)