What are ways of creating 1D, 2D and 3D arrays in NumPy?

1D:

import numpy as np

arr = [1,2,3,4] #python list

numpy\_arr = np.array(arr) #numpy array

2D:

 import numpy as np

arr = [[1,2,3,4],[4,5,6,7]]

numpy\_arr = np.array(arr)

3D:

**import** numpy **as** np

arr = [[[1,2,3,4],[4,5,6,7],[7,8,9,10]]]

numpy\_arr = np.array(arr)

### How can you reverse a NumPy array?

* **Method 1:** Using the slicing method: We can make use of [::-1] for reversing the array. The following example demonstrates this:

**import** numpy **as** np

# create numpy array

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

# To reverse array

reverse\_arr = arr[::-1]

print(reverse\_arr)

* **np.mean()** method calculates the arithmetic mean and provides additional options for input and results. For example, it has the option to specify what data types have to be taken, where the result has to be placed etc.
* **np.average()**
* computes the weighted average if the weights parameter is specified. In the case of weighted average, instead of considering that each data point is contributing equally to the final average, it considers that some data points have more weightage than the others (unequal contribution).

How do we check for an empty array (or zero elements array)?

* We can check for the emptiness of a NumPy array by making use of the size attribute.  
  Let us consider the below example. We have NumPy array arr filled with zeros. If the size element returns zero, that means the array is empty or it only consists of zeros.

import numpy as np

arr = np.zeros((1,0)) #returns empty array

* print(arr.size) #returns 0

### How do you multiply 2 NumPy array matrices?

**import** numpy **as** np

# NumPy matrices

A = np.arange(15,24).reshape(3,3)

B = np.arange(20,29).reshape(3,3)

print("A: ",A)

print("B: ",B)

# Multiply A and B

result = A.dot(B)

* print("Result: ", result)

### How do you concatenate 2 NumPy arrays?

* np.concatenate((a1, a2, ...), axis=0, out=None)

How do you convert Pandas DataFrame to a NumPy array?

The to\_numpy() method of the NumPy package can be used to convert Pandas DataFrame, Index and Series objects.

Consider we have a DataFrame df, we can either convert the whole Pandas DataFrame df to NumPy array or even select a subset of Pandas DataFrame to NumPy array by using the to\_numpy() method as shown in the example below:

import pandas as pd

import numpy as np

# Pandas DataFrame

df = pd.DataFrame(data={'A': [3, 2, 1], 'B': [6,5,4], 'C': [9, 8, 7]},

index=['i', 'j', 'k'])

print("Pandas DataFrame: ")

print(df)

# Convert Pandas DataFrame to NumPy Array

np\_arr = df.to\_numpy()

print("Pandas DataFrame to NumPy array: ")

print(np\_arr)

# Convert specific columns of Pandas DataFrame to NumPy array

arr = df[['B', 'C']].to\_numpy()

print("Convert B and C columns of Pandas DataFrame to NumPy Array: ")

print (arr)

### Write a program to convert a string element to uppercase, lowercase, capitalise the first letter, title-case and swapcase of a given NumPy array.

### Write a program to transform elements of a given string to a numeric string of 10 digits by making all the elements of a given string to a numeric string of 8 digits with zeros on the left.

### Write a program for creating an integer array with values belonging to the range 10 and 60.

### Write a program to add a border of zeros around the existing array.

1. **Concatenate two data frames using the following conditions.**

GermanCars = {'Company': ['Ford', 'Mercedes', 'BMV', 'Audi'], 'Price': [23845, 171995, 135925 , 71400]}

japaneseCars = {'Company': ['Toyota', 'Honda', 'Nissan', 'Mitsubishi '], 'Price': [29995, 23600, 61500 , 58900]}

1. **Merge two data frames using the following condition.**

Create two data frames using the following two Dicts, Merge two data frames, and append the second data frame as a new column to the first data frame.

Car\_Price = {'Company': ['Toyota', 'Honda', 'BMV', 'Audi'], 'Price': [23845, 17995, 135925 , 71400]}

car\_Horsepower = {'Company': ['Toyota', 'Honda', 'BMV', 'Audi'], 'horsepower': [141, 80, 182 , 160]}