1. Create a NumPy array of integers from 10 to 50 (inclusive).

import numpy as np

arr = np.arange(10, 51)

print("Array from 10 to 50:\n", arr)

o/p:

Array from 10 to 50:

[10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50]

2. Create a 3x3 NumPy array of all True values.

arr = np.ones((3,3),dtype=bool)

print("\n3x3 Array is:\n", arr)

o/p:

Array is:

[[ True True True]

[ True True True]

[ True True True]]

3. Create a 5x5 identity matrix.

identity = np.eye(5)

print("\nIdentity Matrix:\n", identity)

o/p:

Identity Matrix:

[[1. 0. 0. 0. 0.]

[0. 1. 0. 0. 0.]

[0. 0. 1. 0. 0.]

[0. 0. 0. 1. 0.]

[0. 0. 0. 0. 1.]]

4. Generate an array of 10 random float numbers between 0 and 1.

rand\_arr = np.random.rand(10)

print("\n10 random float numbers:\n", rand\_arr)

o/p:

10 random float numbers:

[0.57354204 0.65266513 0.20615747 0.25258238 0.44403529 0.36888491

0.53215558 0.89132919 0.67002485 0.13522243]

5. Create a 1D array of 15 numbers equally spaced between 0 and 5.

space = np.linspace(0, 5, 15)

print("\nequally spaced:\n", space)

o/p:

equally spaced:

[0. 0.35714286 0.71428571 1.07142857 1.42857143 1.78571429

2.14285714 2.5 2.85714286 3.21428571 3.57142857 3.92857143

4.28571429 4.64285714 5. ]

6. Reshape an array of 12 elements into a 3x4 matrix.

reshaped = np.arange(12).reshape(3, 4)

print("\nReshaped array is:\n", reshaped)

o/p:

Reshaped array is:

[[ 0 1 2 3]

[ 4 5 6 7]

[ 8 9 10 11]]

7. Replace all even numbers in the array [1, 2, 3, 4, 5, 6] with -1.

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

arr[arr % 2 == 0] = -1

print("\nReplaced Even Numbers is:\n", arr)

o/p:

Replaced Even Numbers is:

[ 1 -1 3 -1 5 -1]

8. Extract all odd numbers from a 1D array ranging from 0 to 20.

arr = np.arange(21)

odds = arr[arr % 2 == 1]

print("\nOdd Numbers from 0 to 20:\n", odds)

o/p:

Odd Numbers from 0 to 20:

[ 1 3 5 7 9 11 13 15 17 19]

9. Create a 2D array of shape (4, 5) and calculate the sum of each column.

arr = np.arange(20).reshape(4, 5)

print("2D Array:", arr)

print("\nSum :\n", arr.sum(axis=0))

o/p:

2D Array: [[ 0 1 2 3 4]

[ 5 6 7 8 9]

[10 11 12 13 14]

[15 16 17 18 19]]

Sum :

[30 34 38 42 46]

10. Create two 3x3 arrays and perform element-wise multiplication.

arr1 = np.array([[1,2,3],[4,5,6],[7,8,9]])

arr2 = np.array([[9,8,7],[6,5,4],[3,2,1]])

product = arr1 \* arr2

print("\nMultiplication:\n", product)

o/p:

Multiplication:

[[ 9 16 21]

[24 25 24]

[21 16 9]]

11. Create an array from 1 to 100 and count how many numbers are divisible by both 3 and 5.

arr = np.arange(1, 101)

count = np.sum((arr % 3 == 0) & (arr % 5 == 0))

print("\nNo of Numbers Divisible by 3 and 5 :\n", count)

o/p:No of Numbers Divisible by 3 and 5:

6

12. Normalize a NumPy array: subtract its mean and divide by its standard deviation.

arr = np.array([10, 20, 30, 40, 50])

mean= np.mean(arr)

std= np.std(arr)

describe = (arr - arr.mean()) / arr.std()

print("\nArray:\n", describe)

o/p:

Array:

[-1.41421356 -0.70710678 0. 0.70710678 1.41421356]