
🧱 1–10: Array Creation and Basics

1. Create a 1D NumPy array of numbers from 0 to 9.

```
import numpy as np

arr = np.arange(10)

print(arr)
```

2. Create a 3×3 NumPy array filled with True.

```
np.full((3,3), True)
```

3. Create an array of even numbers from 10 to 20.

```
np.arange(10, 21, 2)
```

4. Create an array of 10 zeros and replace the fifth element with 5.

```
arr = np.zeros(10)

arr[4] = 5

print(arr)
```

5. Create a 3×3 identity matrix.

```
np.eye(3)
```

6. Create a 3×3 matrix with values ranging from 0 to 8.

```
np.arange(9).reshape(3,3)
```

7. Create a 1D array of 50 evenly spaced numbers between 0 and 1.

```
np.linspace(0, 1, 50)
```

8. Create an array of shape (3,4) with random integers between 10 and 50.

```
np.random.randint(10, 50, (3,4))
```

9. Create a 5×5 matrix with 1 on the border and 0 inside.

```
arr = np.ones((5,5))

arr[1:-1,1:-1] = 0

print(arr)
```

10. Create a diagonal matrix from a given array.

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

np.diag(arr)
```

🧩 11–20: Indexing, Slicing, and Manipulation

11. Extract all odd numbers from an array.

```
arr = np.arange(1, 20)

arr[arr % 2 != 0]
```

12. Replace all odd numbers with -1.

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

13. Reverse a 1D array.

```
arr[::-1]
```

14. Reverse the rows of a 2D array.

```
arr = np.arange(9).reshape(3,3)

arr[::-1]
```

15. Reverse the columns of a 2D array.

```
arr[:, ::-1]
```

16. Flatten a 2D array into 1D.

```
arr.flatten()
```

17. Stack two arrays vertically.

```
a = np.array([1,2,3])

b = np.array([4,5,6])

np.vstack((a,b))
```

18. Stack two arrays horizontally.

```
np.hstack((a,b))
```

19. Split a 1D array into 3 equal parts.

```
arr = np.arange(9)

np.split(arr, 3)
```

20. Add a new axis to make a 1D array 2D (column vector).

```
arr[:, np.newaxis]
```

⚙️ 21–30: Mathematical Operations

21. Compute element-wise addition of two arrays.

```
a = np.array([1,2,3])

b = np.array([4,5,6])

a + b
```

22. Multiply two matrices.

```
A = np.array([[1,2],[3,4]])

B = np.array([[5,6],[7,8]])

A.dot(B)
```

23. Find the square root of each element.

```
np.sqrt(np.array([1,4,9,16]))
```

24. Get the exponential of each element.

```
np.exp(np.array([1,2,3]))
```

25. Compute the sine of array elements.

```
np.sin(np.linspace(0, np.pi, 5))
```

26. Compute the mean, median, and standard deviation.

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

np.mean(arr), np.median(arr), np.std(arr)
```

27. Find the sum of all array elements.

```
np.sum(arr)
```

28. Find the cumulative sum.

```
np.cumsum(np.array([1,2,3,4]))
```

29. Find the minimum and maximum element and their indices.

```
arr = np.random.randint(0,100,10)

arr.min(), arr.max(), arr.argmin(), arr.argmax()
```

30. Compute row-wise and column-wise sum.

```
arr = np.arange(12).reshape(3,4)

arr.sum(axis=0), arr.sum(axis=1)
```

🎲 31–40: Random, Conditional, and Unique Operations

31. Generate a 4×4 matrix of random floats between 0 and 1.

```
np.random.random((4,4))
```

32. Generate a random integer matrix (5×5) between 10 and 100.

```
np.random.randint(10,100,(5,5))
```

33. Set random seed for reproducibility.

```
np.random.seed(42)

np.random.randint(0,10,5)
```

34. Replace all values greater than 10 with 10.

```
arr[arr > 10] = 10
```

35. Find positions where elements are even.

```
np.where(arr % 2 == 0)
```

36. Count unique elements and their frequency.

```
arr = np.array([1,2,2,3,3,3])

np.unique(arr, return_counts=True)
```

37. Clip values of an array between 5 and 10.

```
np.clip(arr, 5, 10)
```

38. Sort an array.

```
np.sort(np.array([5,2,9,1]))
```

39. Get top 3 largest values from an array.

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

arr[np.argsort(arr)[-3:]]
```

40. Check for NaN values and replace them with 0.

```
arr = np.array([1, np.nan, 3])

np.nan_to_num(arr)
```

🧮 41–50: Linear Algebra and Advanced Operations

41. Compute the transpose of a matrix.

```
A = np.arange(9).reshape(3,3)

A.T
```

42. Compute the determinant of a 3×3 matrix.

```
np.linalg.det(A)
```

43. Compute the inverse of a matrix.

```
np.linalg.inv(A)
```

44. Multiply a matrix by a scalar.

```
2 * A
```

45. Solve a system of linear equations: Ax = b.

```
A = np.array([[3,1],[1,2]])

b = np.array([9,8])

np.linalg.solve(A,b)
```

46. Find eigenvalues and eigenvectors.

```
np.linalg.eig(A)
```

47. Compute the trace (sum of diagonal elements).

```
np.trace(A)
```

48. Perform element-wise comparison of two arrays.

```
np.equal([1,2,3],[1,4,3])
```

49. Compute dot product of two 1D arrays.

```
np.dot([1,2,3],[4,5,6])
```

50. Normalize an array (scale values between 0 and 1).

```
arr = np.array([2,4,6,8,10])
```

```
(arr - arr.min()) / (arr.max() - arr.min())
```

✅ **Total:** 50 Numeric NumPy Questions + Answers

✅ **Covers:** Basics → Indexing → Math → Random → Linear Algebra

---