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

```
arr1 = np.arange(10, 51)
print("Array from 10 to 50:\n", arr1)
```

**Output :**

```
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.

```
arr2 = np.ones((3,3), dtype=bool)
print("3x3 Array of True values:\n", arr2)
```

**Output:**

```
3x3 Array of True values:
 [[ True  True  True]
 [ True  True  True]
 [ True  True  True]]
```

3. Create a 5x5 identity matrix.

```
arr3 = np.eye(5)
print("5x5 Identity Matrix:\n", arr3)
```

**Output:**

```
5x5 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.

```
arr4 = np.random.rand(10)
print("10 Random Floats (0 to 1):\n", arr4)
```

**Output:**

```
10 Random Floats (0 to 1):
 [0.01263032 0.13708558 0.74069987 0.27115361 0.30463483 0.58551571
 0.81790939 0.60397678 0.68798101 0.61063106]
```

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

```
arr5 = np.linspace(0, 5, 15)
print("15 Equally Spaced Numbers from 0 to 5:\n", arr5)
```

**Output:**

```
15 Equally Spaced Numbers from 0 to 5:
 [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.

```
arr6 = np.arange(1, 13)
reshaped = arr6.reshape(3, 4)
print("Reshaped 3x4 matrix:\n", reshaped)
```

**Output:**

```
Reshaped 3x4 matrix:
 [[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
```

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

```
arr7 = np.array([1,2,3,4,5,6])
arr7[arr7%2==0] = -1
print("Replace even numbers with -1:\n", arr7)
```

**Output:**

```
Replace even numbers with -1:
 [ 1 -1  3 -1  5 -1]
```

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

```
arr8 = np.arange(0,21)
odd_numbers = arr8[arr8%2==1]
print("Odd Numbers from 0 to 20:\n", odd_numbers)
```

**Output:**

```
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.

```
arr9 = np.array([
    [1, 2, 3, 4, 5],
    [6, 7, 8, 9, 10],
    [11, 12, 13, 14, 15],
    [16, 17, 18, 19, 20]
])
column_sums = np.sum(arr9, axis=0)
print("Array:\n", arr9)
print("Sum of each column:", column_sums)
```

**Output:**

Array:

[[ 1  2  3  4  5]

[ 6  7  8  9 10]

[11 12 13 14 15]

[16 17 18 19 20]]

Sum of each column: [34 38 42 46 50]

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

```
arr10a = np.array([[1, 2, 3],
                   [4, 5, 6],
                   [7, 8, 9]])
arr10b = np.array([[9, 8, 7],
                   [6, 5, 4],
                   [3, 2, 1]])
result = arr10a * arr10b
print("Array 1:\n", arr10a)
print("Array 2:\n", arr10b)
print("Element-wise multiplication:\n", result)
```

**Output:**

Array 1:

[[1 2 3]

[4 5 6]

[7 8 9]]

Array 2:

[[9 8 7]

[6 5 4]

[3 2 1]]

Element-wise 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.

```
arr11 = np.arange(1, 101)
divisible = arr11[(arr11 % 3 == 0) & (arr11 % 5 == 0)]
count = len(divisible)
print("Numbers divisible by both 3 and 5:", divisible)
print("Count:", count)
```

**Output:**

Numbers divisible by both 3 and 5: [15 30 45 60 75 90]

Count: 6

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

```
arr12 = np.array([10, 20, 30, 40, 50])
mean = np.mean(arr12)
std = np.std(arr12)
normalized = (arr12 - mean) / std
print("Original array:", arr12)
print("Normalized array:", normalized)
```

**Output:**

Original array: [10 20 30 40 50]

Normalized array: [-1.41421356 -0.70710678  0.          0.70710678  1.41421356]