

In [2]: `import numpy as np`

In [3]: `# Q1. Create a numpy array starting from 2 till 50 with a stepsize of 3`
`np.arange(2, 50, 3)`

Out[3]: `array([2, 5, 8, 11, 14, 17, 20, 23, 26, 29, 32, 35, 38, 41, 44, 47])`

In [13]: `# Q2. Accept two lists of 5 elements each from the user.`
`# Convert them to numpy arrays. Concatenate these arrays`
`# and print it. Also sort these arrays and print it.`
`l = []`
`for i in range(5):`
 `l.append(float(input("Enter 5 numbers:")))`
`print(l)`
`arr = np.array(l)`
`print(arr)`
`print(type(arr))`
`l1 = []`
`for i in range(5):`
 `l1.append(float(input("Enter 5 numbers:")))`
`print(l1)`
`arr1 = np.array(l1)`
`print(arr1)`
`print(type(arr1))`
`print(np.concatenate((arr, arr1)))`
`print(np.sort(np.concatenate((arr, arr1))))`

```
Enter 5 numbers:5
Enter 5 numbers:3
Enter 5 numbers:4
Enter 5 numbers:6
Enter 5 numbers:2
[5.0, 3.0, 4.0, 6.0, 2.0]
[5. 3. 4. 6. 2.]
<class 'numpy.ndarray'>
Enter 5 numbers:8
Enter 5 numbers:9
Enter 5 numbers:3
Enter 5 numbers:1
Enter 5 numbers:5
[8.0, 9.0, 3.0, 1.0, 5.0]
[8. 9. 3. 1. 5.]
<class 'numpy.ndarray'>
[5. 3. 4. 6. 2. 8. 9. 3. 1. 5.]
[1. 2. 3. 3. 4. 5. 5. 6. 8. 9.]
```

In [15]: *# Q3. Write a code snippet to find the dimensions of a ndarray and its size.*

```
arr = np.array([[[0, 1, 2, 3],
...             [4, 5, 6, 7]],
...            [[0, 1, 2, 3],
...             [4, 5, 6, 7]],
...            [[0, 1, 2, 3],
...             [4, 5, 6, 7]]])
print(arr.ndim)
print(arr.size)
```

3

24

In [21]: *# Q4. How to convert a 1D array into a 2D array? Demonstrate with the help of a code snippet.
Hint: np.newaxis, np.expand_dims*

```
a = np.array([1, 2, 3, 4, 5, 6])
print(a)
print(a.shape)
a2 = a[np.newaxis, :]
print(a2)
print(a2.shape)
a3 = np.expand_dims(a, axis=0)
print(a3)
print(a3.shape)
```

[1 2 3 4 5 6]

(6,)

[[1 2 3 4 5 6]]

(1, 6)

[[1 2 3 4 5 6]]

(1, 6)

In [24]: *# Q5. Consider two square numpy. Stack them vertically and horizontally.
Hint: Use vstack(), hstack()*

```
a1 = np.array([[1, 1],
               [2, 2]])

a2 = np.array([[3, 3],
               [4, 4]])

print(a1)
print(a2)
print(np.vstack((a1, a2)))
print(np.hstack((a1, a2)))
```

```
[[1 1]
 [2 2]]
[[3 3]
 [4 4]]
[[1 1]
 [2 2]
 [3 3]
 [4 4]]
[[1 1 3 3]
 [2 2 4 4]]
```

In [27]: *# Q6. How to get unique items and counts of unique items?*

```
a = np.array([11, 11, 12, 13, 14, 15, 16, 17, 12, 13, 11, 14, 18, 19, 20])
print(a)
print(np.unique(a))
unique_values, occurrence_count = np.unique(a, return_counts=True)
print(occurrence_count)
```

```
[11 11 12 13 14 15 16 17 12 13 11 14 18 19 20]
[11 12 13 14 15 16 17 18 19 20]
[3 2 2 2 1 1 1 1 1 1]
```