

4. NUMPY – IMPORTANT METHODS

Contents

1. Numpy Array Methods.....	2
2. min() method.....	2
3. max() method	3
4. sum() method	4
5. reshape() method	5
6. count_nonzero(p) function	8
7. sort() method	9
8. flatten() method	10
9. adding value to array of values.....	11
10. Diagonal of a Matrix.....	12
11. Trace of a Matrix	13
12. Adding and Subtracting Matrices	15

4. NUMPY – IMPORTANT METHODS

1. Numpy Array Methods

- ✓ Numpy array having predefined methods to perform different operations over array.

2. min() method

- ✓ min() is a predefined method in numpy array.
- ✓ We should access this min() method by using numpy array object
- ✓ By using this we can check minimum value from the array.

Program min() method
Name demo1.py

```
import numpy as np

details = [[10, 20, 30], [40, 50, 60]]
sales = np.array(details)
print(sales)
print(sales.min())
```

Output

10

3. max() method

- ✓ max() is a predefined method in numpy array.
- ✓ We should access this max() method by using numpy array object
- ✓ By using this we can check maximum value from the array.

Program max() method
Name demo2.py

```
import numpy as np

details = [[10, 20, 30], [40, 50, 60]]
sales = np.array(details)
print(sales)
print(sales.max())
```

Output

60

4. sum() method

- ✓ sum() is a predefined method in numpy array.
- ✓ We should access this method by using numpy array object
- ✓ By using this we can get sum of all values from array.

Program sum() method
Name demo3.py

```
import numpy as np

details = [[10, 20, 30], [40, 50, 60]]
sales = np.array(details)
print(sales)
print()
print(sales.sum())
```

Output

```
[[10 20 30]
 [40 50 60]]

210
```

5. reshape() method

- ✓ reshape() is a predefined method in numpy array.
- ✓ We should access this method by using numpy array object
- ✓ By using this we can change the shape of an array.

Program reshape() method
Name demo4.py

```
import numpy as np

details = [[10, 20, 30], [40, 50, 60]]
sales = np.array(details)
print(sales)
print()
print(sales.reshape(3, 2))
```

Output

```
[[10 20 30]
 [40 50 60]]

[[10 20]
 [30 40]
 [50 60]]
```

Program reshape() method
Name demo5.py

```
import numpy as np

details = [[10, 20, 30], [40, 50, 60]]
sales = np.array(details)
print(sales)
print()
print(sales.reshape(1, 6))
```

Output

```
[[10 20 30]
 [40 50 60]]

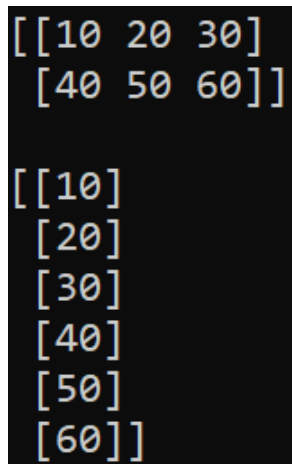
[[10 20 30 40 50 60]]
```

Program reshape() method
Name demo6.py

```
import numpy as np

details = [[10, 20, 30], [40, 50, 60]]
sales = np.array(details)
print(sales)
print()
print(sales.reshape(6, 1))
```

Output



```
[[10 20 30]
 [40 50 60]]

[[10]
 [20]
 [30]
 [40]
 [50]
 [60]]
```

6. count_nonzero(p) function

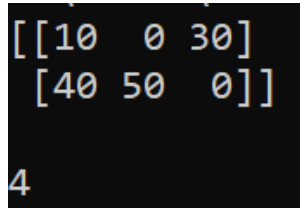
- ✓ count_nonzero(p) is a predefined function in numpy array.
- ✓ We should access this function by using numpy.
- ✓ By using this we can get non zero values from numpy

Program count_nonzero(p) function
Name demo7.py

```
import numpy as np

details = [[10, 20, 30], [40, 50, 60]]
sales = np.array(details)
print(sales)
print()
print(np.count_nonzero(sales))
```

Output



```
[[10  0 30]
 [40 50  0]]
4
```


7. sort() method

- ✓ sort() is a predefined method in numpy array.
- ✓ We should access this method by using numpy array object
- ✓ By using this we can sort values in array.

Program sort() method
Name demo8.py

```
import numpy as np

details = [[55, 13, 12], [99, 2, 1]]
sales = np.array(details)
print(sales)
sales.sort()

print()
print(sales)
```

Output

```
[[55 13 12]
 [99  2  1]]

[[12 13 55]
 [ 1  2 99]]
```

8. flatten() method

- ✓ flatten() is a predefined method in numpy array.
- ✓ We should access this method by using numpy array object
- ✓ This method keeps all values in one dimension array.

Program flatten() method
Name demo9.py

```
import numpy as np

details = [[10, 20, 30], [40, 50, 60]]
sales = np.array(details)
print(sales)
print()
print(sales.flatten())
```

Output

```
[[10 20 30]
 [40 50 60]]

[[10 20]
 [30 40]
 [50 60]]
```

9. adding value to array of values

- ✓ Based on requirement we can add value to array of values.

Program Adding value to array of values
Name demo10.py

```
import numpy as np

details = [[10, 20, 30], [40, 50, 60]]
sales = np.array(details)
print(sales)
print()
print(sales + 2)
```

Output

```
[[10 20 30]
 [40 50 60]]

[[12 22 32]
 [42 52 62]]
```

10. Diagonal of a Matrix

- ✓ Diagonal elements of a matrix.

Program Diagonal matrix
Name demo11.py

```
import numpy as np

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

print(matrix)
print()
print(matrix.diagonal())
```

Output

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]

[1 5 9]
```

11. Trace of a Matrix

- ✓ The trace of a matrix is the sum of the diagonal elements.

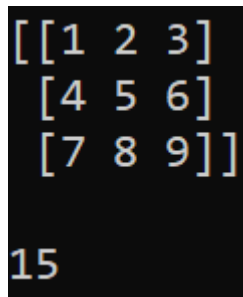
Program Trace of the matrix
Name demo12.py

```
import numpy as np

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

print(matrix)
print()
print(matrix.trace())
```

Output



```
[[1 2 3]
 [4 5 6]
 [7 8 9]]

15
```

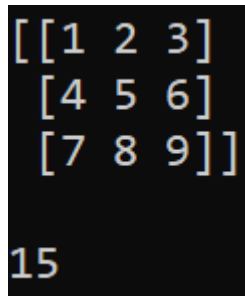
Program Name Trace of the matrix
demo13.py

```
import numpy as np

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

print(matrix)
print()
print(sum(matrix.diagonal()))
```

Output



```
[[1 2 3]
 [4 5 6]
 [7 8 9]]

15
```

12. Adding and Subtracting Matrices

- ✓ We can add & subtract two matrices.
- ✓ We need to call add and subtract functions

Program Name Adding two matrices
demo14.py

```
import numpy as np

matrix_a = np.array([[1, 1, 1], [1, 1, 1], [1, 1, 2]])
matrix_b = np.array([[1, 3, 1], [1, 3, 1], [1, 3, 8]])

print(matrix_a)
print()
print(matrix_b)
print()
print(np.add(matrix_a, matrix_b))
```

Output

```
[[1 1 1]
 [1 1 1]
 [1 1 2]]

[[1 3 1]
 [1 3 1]
 [1 3 8]]

[[ 2  4  2]
 [ 2  4  2]
 [ 2  4 10]]
```

Program Name Subtracting two matrices
demo15.py

```
import numpy as np

matrix_a = np.array([[1, 1, 1], [1, 1, 1], [1, 1, 2]])
matrix_b = np.array([[1, 3, 1], [1, 3, 1], [1, 3, 8]])

print(matrix_a)
print()
print(matrix_b)
print()
print(np.subtract(matrix_a, matrix_b))
```

Output

```
[[ 0 -2  0]
 [ 0 -2  0]
 [ 0 -2 -6]]
```


Program Name Adding two matrices
demo16.py

```
import numpy as np

matrix_a = np.array([[1, 1, 1], [1, 1, 1], [1, 1, 2]])
matrix_b = np.array([[1, 3, 1], [1, 3, 1], [1, 3, 8]])

print(matrix_a + matrix_b)
```

Output

```
[[ 2  4  2]
 [ 2  4  2]
 [ 2  4 10]]
```

Program Name Subtracting two matrices
demo17.py

```
import numpy as np

matrix_a = np.array([[1, 1, 1], [1, 1, 1], [1, 1, 2]])
matrix_b = np.array([[1, 3, 1], [1, 3, 1], [1, 3, 8]])

print(matrix_a - matrix_b)
```

Output

```
[[ 0 -2  0]
 [ 0 -2  0]
 [ 0 -2 -6]]
```