## 4. NUMPY – IMPORTANT METHODS

## Contents

## 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 Name | min() method<br>demo1.py<br><br>import numpy as np<br><br>details = np.array([[10, 20, 30], [40, 50, 60]])<br>sales = np.array(details)<br>print(sales)<br>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 Name | max() method<br>demo2.py<br><br>```python
import numpy as np

details = np.array([[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 Name** | sum() method<br>demo3.py<br><br>```python<br>import numpy as np<br><br>details = np.array([[10, 20, 30], [40, 50, 60]])<br>sales = np.array(details)<br>print(sales)<br>print()<br>print(sales.sum())<br>``` |
| **Output** | ```<br>[[10 20 30]<br> [40 50 60]]<br><br>210<br>``` |

## 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 Name | reshape() method<br>demo4.py<br><br>```python<br>import numpy as np<br><br>details = np.array([[10, 20, 30], [40, 50, 60]])<br>sales = np.array(details)<br>print(sales)<br>print()<br>print(sales.reshape(3, 2))<br>``` |
| Output | ```<br>[[10 20 30]<br> [40 50 60]]<br><br>[[10 20]<br> [30 40]<br> [50 60]]<br>``` |

| | |
|---|---|
| Program Name | reshape() method<br>demo5.py<br><br>import numpy as np<br><br>details = np.array([[10, 20, 30], [40, 50, 60]])<br>sales = np.array(details)<br>print(sales)<br>print()<br>print(sales.reshape(1, 6)) |
| Output | |

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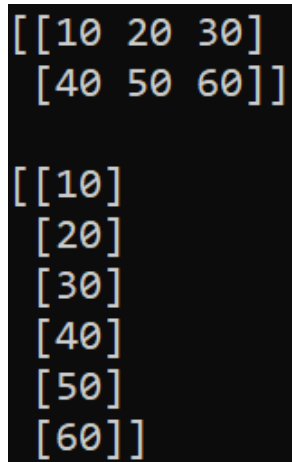[[10 20 30]
 [40 50 60]]

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

| Program Name | reshape() method<br>demo6.py |
|---|---|
| | ```python<br>import numpy as np<br><br>details = np.array([[10, 20, 30], [40, 50, 60]])<br>sales = np.array(details)<br>print(sales)<br>print()<br>print(sales.reshape(6, 1))<br>``` |
| 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 Name | count_nonzero(p) function<br>demo7.py |
|---|---|

```
import numpy as np

details = np.array([[10, 0, 30], [40, 50, 0]])
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 Name | sort() method<br>demo8.py<br><br>```python<br>import numpy as np<br><br>details = np.array([[55, 13, 12], [99, 2, 1]])<br>sales = np.array(details)<br>print(sales)<br>sales.sort()<br><br>print()<br>print(sales)<br>``` |
| Output | ```<br>[[55 13 12]<br> [99  2  1]]<br><br>[[12 13 55]<br> [ 1  2 99]]<br>``` |

## 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 Name | flatten() method<br>demo9.py<br><br>```python<br>import numpy as np<br><br>details = np.array([[10, 20, 30], [40, 50, 60]])<br>sales = np.array(details)<br>print(sales)<br>print()<br>print(sales.flatten())<br>``` |
| Output | ```<br>[[10 20 30]<br> [40 50 60]]<br><br>[[10 20]<br> [30 40]<br> [50 60]]<br>``` |

## 9. adding value to array of values

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

| | |
|---|---|
| Program Name | Adding value to array of values <br> demo10.py <br><br> import numpy as np <br><br> details = np.array([[10, 20, 30], [40, 50, 60]]) <br> sales = np.array(details) <br> print(sales) <br> print() <br> print(sales + 2) |
| Output | |

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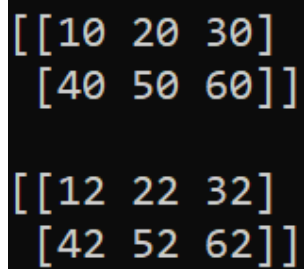[[10 20 30]
 [40 50 60]]

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

## 10. Diagonal of a Matrix

✓ Diagonal elements of a matrix.

| | |
|---|---|
| Program Name | Diagonal matrix<br>demo11.py<br><br>import numpy as np<br><br>matrix = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])<br><br>print(matrix)<br>print()<br>print(matrix.diagonal()) |
| Output | |

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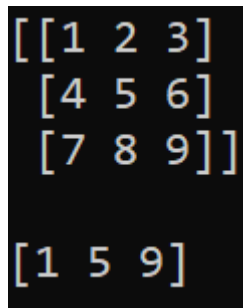[[1 2 3]
 [4 5 6]
 [7 8 9]]

[1 5 9]
```

**11. Trace of a Matrix**

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

---

| | |
|---|---|
| Program Name | Trace of the matrix<br>demo12.py |

```python
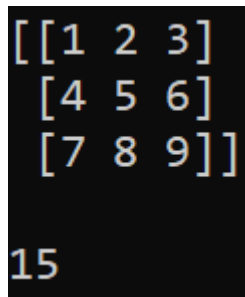import numpy as np

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

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

Output

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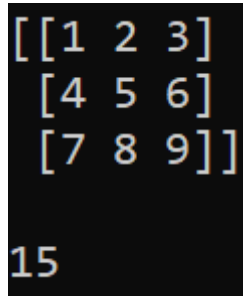[[1 2 3]
 [4 5 6]
 [7 8 9]]

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<br>demo14.py<br><br>```python<br>import numpy as np<br><br>matrix_a = np.array([[1, 1, 1], [1, 1, 1], [1, 1, 2]])<br>matrix_b = np.array([[1, 3, 1], [1, 3, 1], [1, 3, 8]])<br><br>print(matrix_a)<br>print()<br>print(matrix_b)<br>print()<br>print(np.add(matrix_a, matrix_b))<br>``` |
| Output | ```<br>[[1 1 1]<br> [1 1 1]<br> [1 1 2]]<br><br>[[1 3 1]<br> [1 3 1]<br> [1 3 8]]<br><br>[[ 2  4  2]<br> [ 2  4  2]<br> [ 2  4 10]]<br>``` |

Program Name
Subtracting two matrices
demo15.py

```python
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<br>demo16.py<br><br>import numpy as np<br><br>matrix_a = np.array([[1, 1, 1], [1, 1, 1], [1, 1, 2]])<br>matrix_b = np.array([[1, 3, 1], [1, 3, 1], [1, 3, 8]])<br><br>print(matrix_a + matrix_b) |
| **Output** | ```
[[ 2  4  2]
 [ 2  4  2]
 [ 2  4 10]]
``` |

| | |
|---|---|
| **Program Name** | Subtracting two matrices<br>demo17.py<br><br>import numpy as np<br><br>matrix_a = np.array([[1, 1, 1], [1, 1, 1], [1, 1, 2]])<br>matrix_b = np.array([[1, 3, 1], [1, 3, 1], [1, 3, 8]])<br><br>print(matrix_a - matrix_b) |
| **Output** | ```
[[ 0 -2  0]
 [ 0 -2  0]
 [ 0 -2 -6]]
``` |