In [1]:

```
pip install numpy
```

```
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (1.25.2)
```

Creating array

Creating a NumPy ndarray Object

In [2]:

```
import numpy as np
arr=np.array([1,2,3,4,5])
print(arr)
print(type(arr))
```

```
[1 2 3 4 5]
<class 'numpy.ndarray'>
```

In [3]:

```
print(np.__version__)
```

```
1.25.2
```

Dimensions in Arrays

Creating 0 Dimension

In [4]:

```
import numpy as np

arr = np.array(42)

print(arr)
```

```
42
```

1-D Arrays

In [5]:

```
import numpy as np

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

print(arr)
```

```
[1 2 3 4 5]
```

2-D Array

In [6]:

```
import numpy as np
```

```
arr = np.array([[1, 2, 3], [4, 5, 6]])

print(arr)
```

```
[[1 2 3]
 [4 5 6]]
```

3-D Array

In [7]:

```
import numpy as np

arr = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])

print(arr)
```

```
[[[1 2 3]
  [4 5 6]]

 [[1 2 3]
  [4 5 6]]]
```

Checking No.of Dimensions

In [8]:

```
import numpy as np

a = np.array(42)
b = np.array([1, 2, 3, 4, 5])
c = np.array([[1, 2, 3], [4, 5, 6]])
d = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])

print(a.ndim)
print(b.ndim)
print(c.ndim)
print(d.ndim)
```

```
0
1
2
3
```

NumPy Array Indexing

In [9]:

```
import numpy as np

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

print(arr[0])
```

```
1
```

In [10]:

```
import numpy as np

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

print(arr[2],arr[3])
```

```
3 4
```

Access 2-D Arrays

In [11]:

```
import numpy as np

arr = np.array([[1,2,3,4,5], [6,7,8,9,10]])
print(arr)
print('2nd element on 1st row: ', arr[1, 4])
```

```
[[ 1  2  3  4  5]
 [ 6  7  8  9 10]]
2nd element on 1st row:  10
```

In [12]:

```
import numpy as np

arr = np.array([[1,2,3,4,5], [6,7,8,9,10]])

print('5th element on 2nd row: ', arr[1, 4])
```

```
5th element on 2nd row:  10
```

Access 3-D Arrays

In [13]:

```
import numpy as np

arr = np.array([[[1, 2, 3], [4, 5, 6]], [[7, 8, 9], [10, 11, 12]]])

print(arr[1, 1, 1])
```

```
11
```

Negative Indexing

In [14]:

```
import numpy as np

arr = np.array([[1,2,3,4,5], [6,7,8,9,10]])

print('Last element from 2nd dim: ', arr[1, -1])
```

```
Last element from 2nd dim:  10
```

NumPy Array Slicing

Slicing arrays

In [15]:

```
import numpy as np

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

print(arr[1:3])
```

```
[2 3]
```

Negative Slicing

Negative Slicing

In [16]:

```
import numpy as np

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

print(arr[-3:-2])
```

```
[5]
```

STEP

In [17]:

```
import numpy as np

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

print(arr[1:6:2])
```

```
[2 4 6]
```

In [18]:

```
import numpy as np

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

print(arr[:5:2])
```

```
[1 3 5]
```

Slicing 2-D Arrays

In [19]:

```
import numpy as np

arr = np.array([[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]])

print(arr[0, 1:4])
```

```
[2 3 4]
```

In [20]:

```
import numpy as np

arr = np.array([[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]])

print(arr[1,0:2])
```

```
[6 7]
```

In [21]:

```
import numpy as np

arr = np.array([[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]])

print(arr[0:2,3])
```

```
[4 9]
```

In [22]:

```
import numpy as np

arr = np.array([[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]])

print(arr[0:2, 1:4])
```

```
[[2 3 4]
 [7 8 9]]
```

NumPy Array Shape

In [23]:

```
import numpy as np

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

print(arr.shape)
```

```
(2, 4)
```

Reshaping arrays

Reshape From 1-D to 2-D

In [24]:

```
import numpy as np

arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])

newarr = arr.reshape(4, 3)

print(newarr)
```

```
[[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]]
```

Reshape From 1-D to 3-D

In [25]:

```
import numpy as np

arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])

newarr = arr.reshape(3,4)

print(newarr)
```

```
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
```