Slicing in NumPy arrays is similar to how it works with a Python list

In [1]:

```
import numpy as np
arr = np.array([[1,2,3,4,5], [4,5,6,7,8], [9,8,7,6,5]])
print("The NumPy Array is:")
print(arr)
arr1 = arr[1:3, :3] # row 1 to 3 (not inclusive) and first 3 columns
print("The Sliced NumPy Array is:")
print(arr1)
```

```
The NumPy Array is:
[[1 2 3 4 5]
 [4 5 6 7 8]
 [9 8 7 6 5]]
The Sliced NumPy Array is:
[[4 5 6]
 [9 8 7]]
```

Slicing in python means taking elements from one given index to another given index.

We pass slice instead of index like this: [start: end].

We can also define the step, like this: [start: end: step].

If we don't pass start its considered 0.

If we don't pass end it's considered length of array in that dimension.

If we don't pass step it's considered 1.

Program 2: Slice elements from index 1 to index 5 from the following array.

In [2]:

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5, 6, 7])
print(arr[1:5])
#Note: The result includes the start index, but excludes the end index.
```

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

Program 3: Slice elements from index 4 to the end of the array

In [3]:

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5, 6, 7])
print(arr[4:])
```

```
[5 6 7]
```

Program 4: Slice elements from the beginning to index 4 (not included).

In [4]:

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5, 6, 7])
print(arr[:4])
```

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

Negative Slicing: Use the minus operator to refer to an index from the end.

Program 5: Slice from the index 3 from the end to index 1 from the end.

In [5]:

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5, 6, 7])
print(arr[-3:-1])
```

```
[5 6]
```

STEP: Use the step value to determine the step of the slicing.

Program 6: Return every other element from index 1 to index 5.

In [6]:

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5, 6, 7])
print(arr[1:5:2])
```

```
[2 4]
```

Program 7: Return every other element from the entire array.

In [7]:

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5, 6, 7])
print(arr[::2])
```

```
[1 3 5 7]
```

Slicing 2-D Array's

Program 8: From the second element, slice elements from index 1 to index 4 (not included).

In [8]:

```
import numpy as np
arr = np.array([[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]])
print(arr[1, 1:4]) # from second element, so only 1,
# they didn't say from second till end and stuff like that

# NOTICE THAT THE RESULT IS A ONE DIMENSTIONAL ARRAY
```

```
[7 8 9]
```

Program 9: From both elements, return index 2.

In [9]:

```
import numpy as np
arr = np.array([[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]])
print(arr[0:2, 2]) # from both elements - both rows

# NOTICE THAT THE RESULT IS A ONE DIMENSTIONAL ARRAY
```

```
[3 8]
```

Program 10: From both elements, slice index 1 to index 4 (not included), this will return a 2-D array

In [10]:

```
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]]
```

In [ ]:

```

```