Introduction to NumPy: Takeaways

Syntax

ACCESSING AND UPDATING VALUES IN AN NDARRAY:

- Access and update values in a 1-dimensional array:

```
x = np.array([5, 10, 15, 20])
print(x[1]) # Will print 10
x[2] = 42   # Will update the value 15 to 42
```

- Access and update values in a 2-dimensional array:

```
x = np.array([
    [1, 2, 3, 4],
    [5, 6, 7, 8]
])
print(x[1, 1]) # Will print 6
x[0, 2] = 42   # Will update the value 3 to 42
```

SLICING AN NDARRAY

- Slice a 1-dimensional array:

```
x[start:end:step]
```

- Examples:

```
x[2:5]    # Select indexes 2, 3, and 4
x[2:8:2]  # Select indexes 2, 4 and 6
x[8:2:-2] # Select indexes 8, 6 and 4
x[:5]     # Select indexes 0, 1, 2, 3 and 4
x[5:]     # Select indexes 5, 6, 7, ..., len(x) - 1
x[:]      # Select all indexes
x[::-1]   # Select all indexes in reverse order
```

- Slice a 2-dimensional array:

```
x[row_start:row_end:row_step, col_start:col_end:col_step]
```

- Examples:

```
x[1:4,3:6]    # Select rows 1,2,3, cols 3,4,5
x[1:4,3:6:2]  # Select rows 1,2,3, cols 3,5
x[1:4:2,3:6]  # Select rows 1,3, cols 3,4,5
```

- Select rows and columns with lists:

```
x[[0, 4],:] # Select rows 0 and 4
x[:,[2, 5]] # Select columns 2 and 5
```

COPYING AN NDARRAY

- To get a view slice an array. If we want a copy we can do it like so:

```
x_copy = x.copy()
```

Concepts

- NumPy uses ndarrays to store data. The name stands from n-dimensional array. An array is a rigid data structure whose size is fixed at creation and cannot be altered.
- We can obtain parts of an ndarray through slicing. When slicing an ndarray, we do not get a brand new copy. Instead, we get a view into that array. This means that any value modification is reflected in both the original array and the slide.

Resources

- NumPy ndarray documentation