Numpy Axes Explained

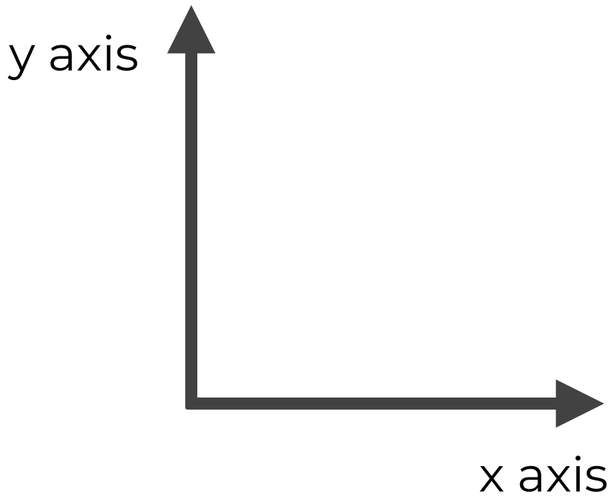
SHARP SIGHT

What you'll learn

- What array axes are
- How to think about array axes
- How axes are different for 1D and 2D arrays
- Why axes are important

How to think about Numpy array axes

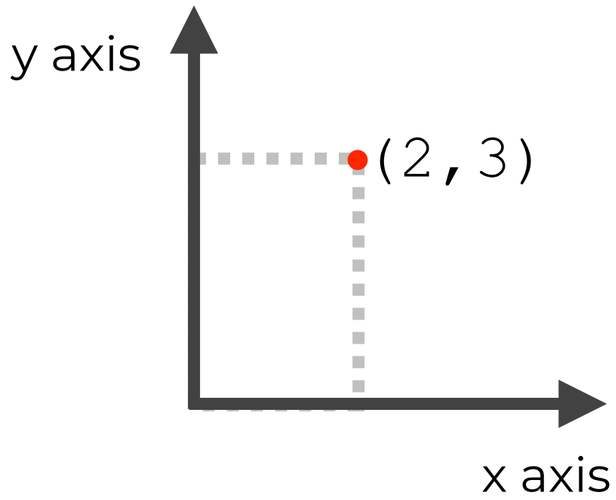
Array axes are like axes in a coordinate system



For example, a Cartesian coordinate system has an x axis and y axis

These axes are like directions in space

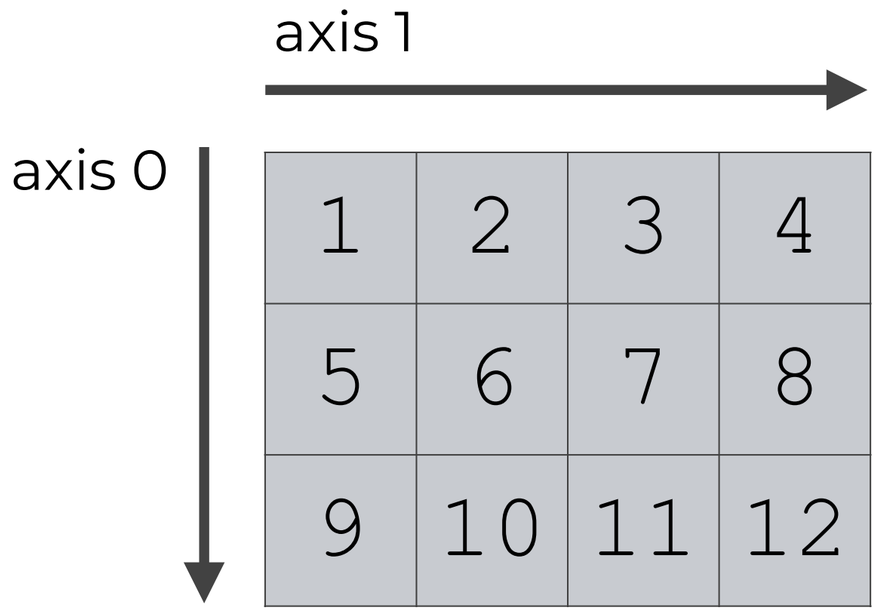
In a coordinate system, points can be defined by values along the axes



Here, the point lies at
2 units along the x axis,
and 3 units along the y axis

NumPy array axes

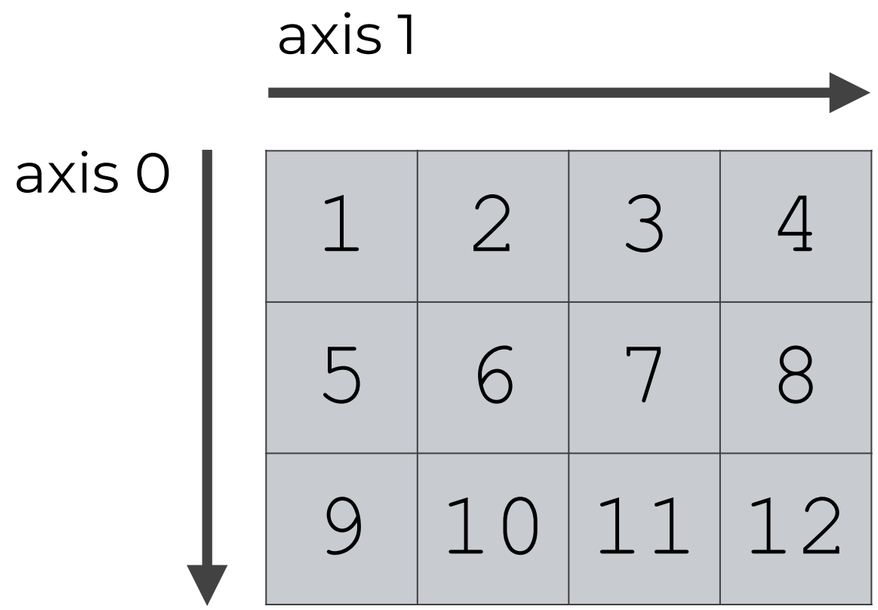
NumPy arrays have *axes*



Array axes are very similar to axes in coordinate systems

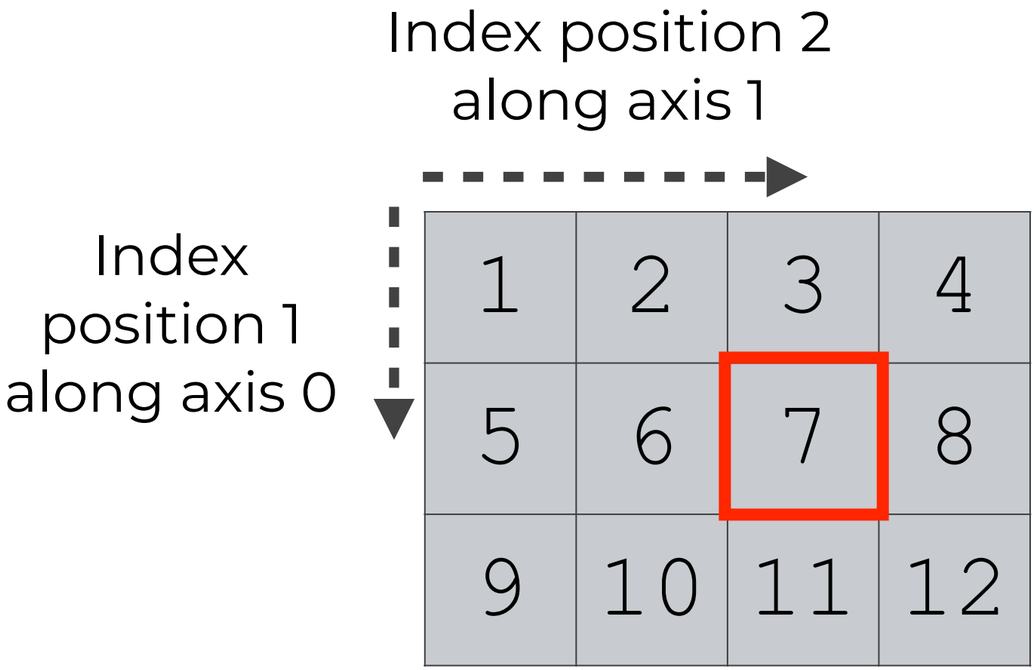
Axes are like *directions* along a NumPy array

Axis-1 is the direction that runs horizontally across the columns



Axis-0 is the direction that runs downward down the rows

Any value in an array can be identified by its position along the axes



This cell is at location [1, 2]

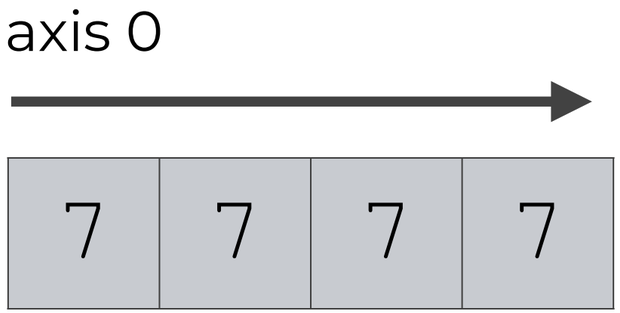
Index 1 along axis 0

Index 2 along axis 1

Remember: indexes in Python start at 0!

Axes for 1D and 2D arrays

In a 1-dimensional array, the first axis is axis-0

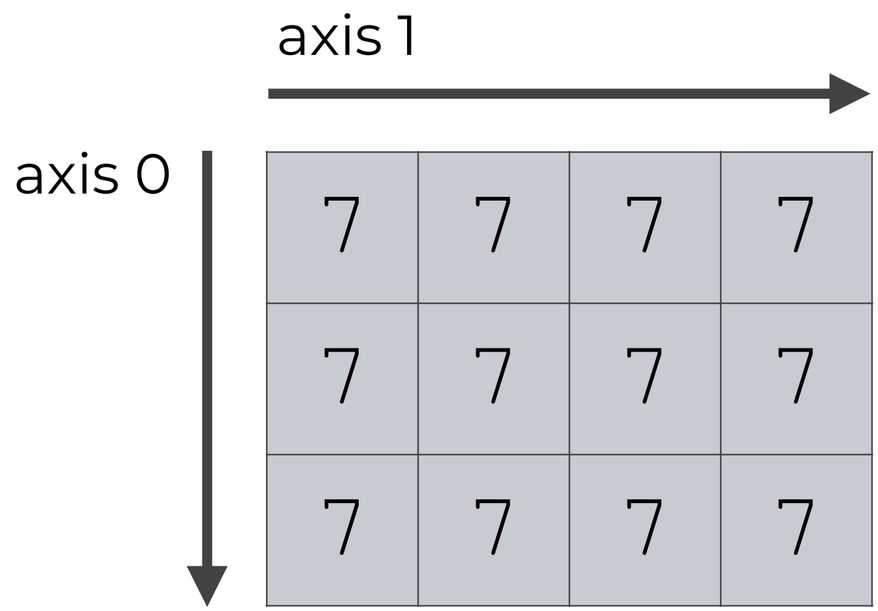


Just remember, 1-dimensional arrays are a little different

In a 2-dimensional array, axis-0 is down and axis-1 is across

Axis-1 is the direction that runs horizontally across the columns

Axis-0 is the direction that runs downward down the rows



Why you need to know axes

NumPy axes are important!

- We will use axes when we use many functions
  - `np.sum()`
  - `np.mean()`
  - `np.concatenate()`
  - `np.sort()`
  - etc

- We commonly use axes when we need to aggregate, sort, or manipulate

- Make sure you understand them!
  - It's best to memorize them

Recap

Recap of what we learned

- NumPy arrays have "axes"
- Axes are directions along an array
- Array axes are like axes in a Cartesian plotting system
- Array axes are different for 1D and 2D arrays
- Axes are important, so memorize them