GENERAL ASSEMBLY

ARRAYS AND FUNCTIONS

Haley Boyan

Data Science Instructional Associate, General Assembly DC

LEARNING OBJECTIVES

- Describe a matrix and an array
- Explain how to add and subtract arrays
- Explain the dot product

ARRAYS AND FUNCTIONS

PRE-WORK

PRE-WORK REVIEW

- https://www.khanacademy.org/computing/computer-programming/programming/arrays/p/intro-to-arrays

WHY USE ARRAYS

WHY USE ARRAYS?

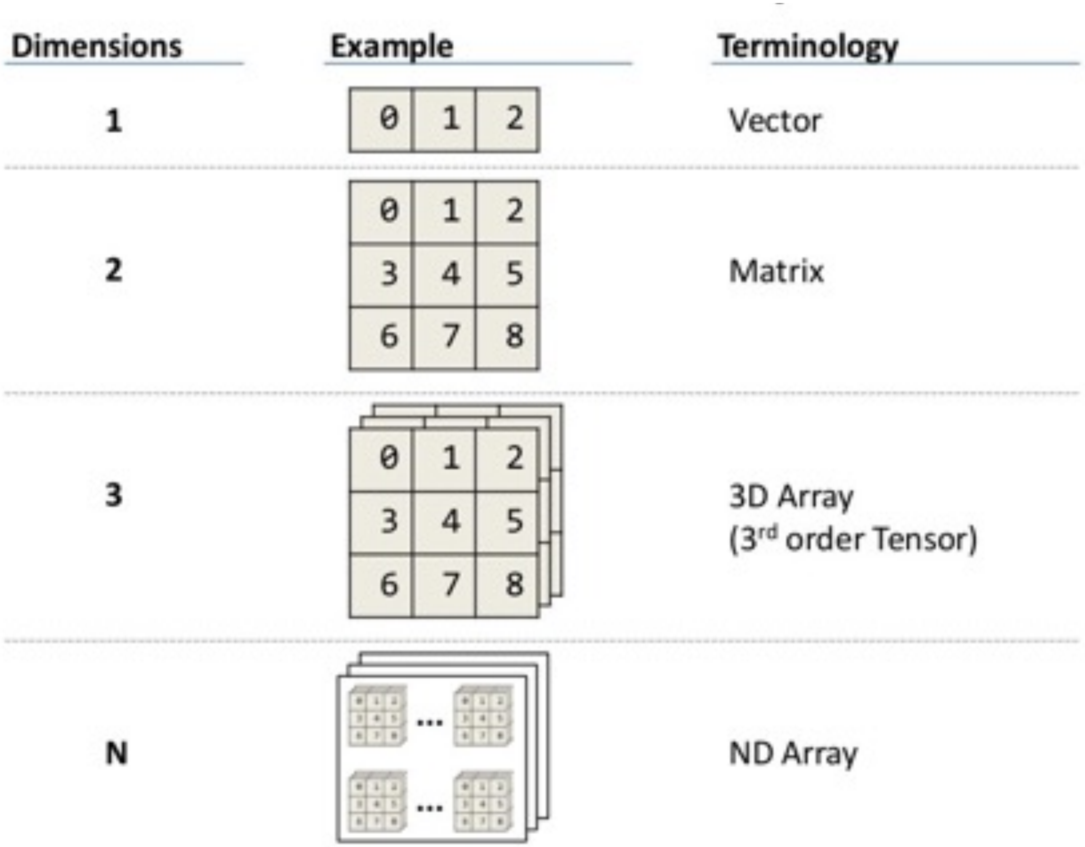
- An array is a collection of related data items, called elements, associated with a single variable name.
- Arrays can:
  - Ease programming and offer improved performance.
  - Simplify the task of naming and referencing items in a collection
  - Boost the performance of your application
  - Let you manipulate an entire collection of data items with a single statement.

ARRAYS AND FUNCTIONS

HIP HIP ARRAY

WHAT IS AN ARRAY?

- A data structure that allows you to group variables under a single name
- Variables in an array can be numeric or string variables, but not both
- Specific types of arrays:
  - Vector: 1-Dimensional
  - Matrix: 2-Dimensional
- Arrays can have any number of dimensions



Dimensions	Example	Terminology
1	0 1 2	Vector

2	0 1 2
	3 4 5
	6 7 8
		Matrix

3	0 1 2
	3 4 5
	6 7 8
		3D Array
		(3rd order Tensor)

N	...	ND Array

ADDING AND SUBTRACTING ARRAYS

- Arrays can be added or subtracted by performing the operation on the corresponding elements
- Arrays must be the same shape for NumPy to do this

$$\begin{bmatrix} 4 & -2 \\ 7 & 0 \end{bmatrix} - \begin{bmatrix} 3 & 2 \\ 5 & -3 \end{bmatrix} = \begin{bmatrix} 1 & -4 \\ 2 & 3 \end{bmatrix}$$

DOT PRODUCT

- Allows comparison of two vectors into a single value
- Results in a scalar product that is coordinated with the angle between two vectors

$$\begin{bmatrix} a & b \end{bmatrix} \bullet \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax + by \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \bullet \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax + by \\ cx + dy \end{bmatrix}$$

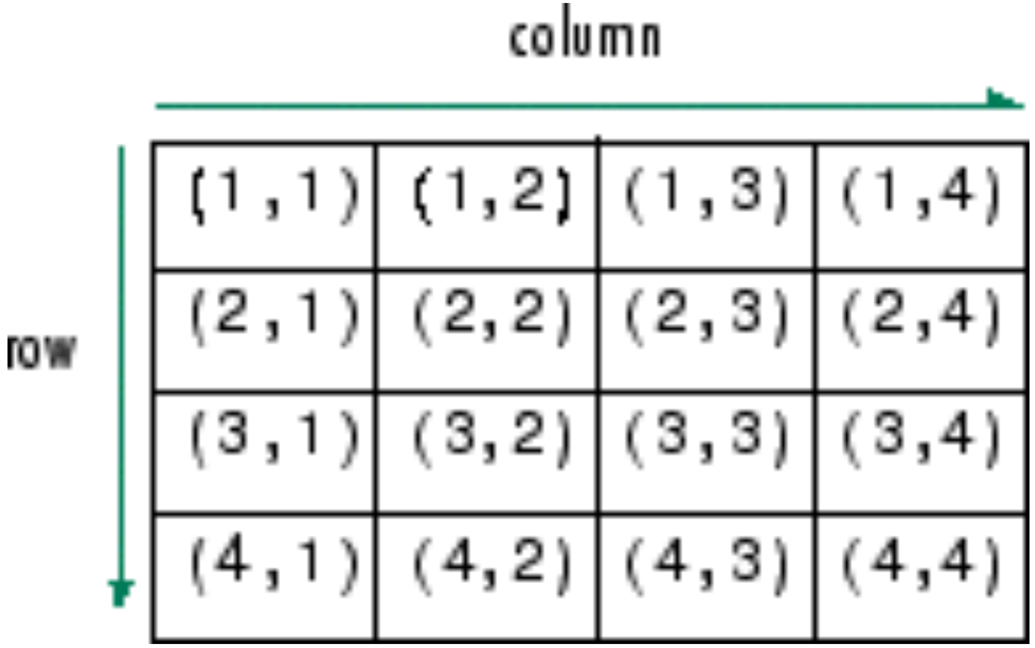
$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \bullet \begin{bmatrix} w & x \\ y & z \end{bmatrix} = \begin{bmatrix} aw + by & ax + bz \\ cw + dy & cx + dz \end{bmatrix}$$

ARRAY OBJECTS

- Arrays are an object in Python, of type ndarray
- Create a random array with:
  - `np.arange(start, stop, step)`
    - can add `.reshape(dimensions)` to specify size
  - `np.ones(shape)`
  - `np.zeros(shape)`

MAPPING ARRAYS

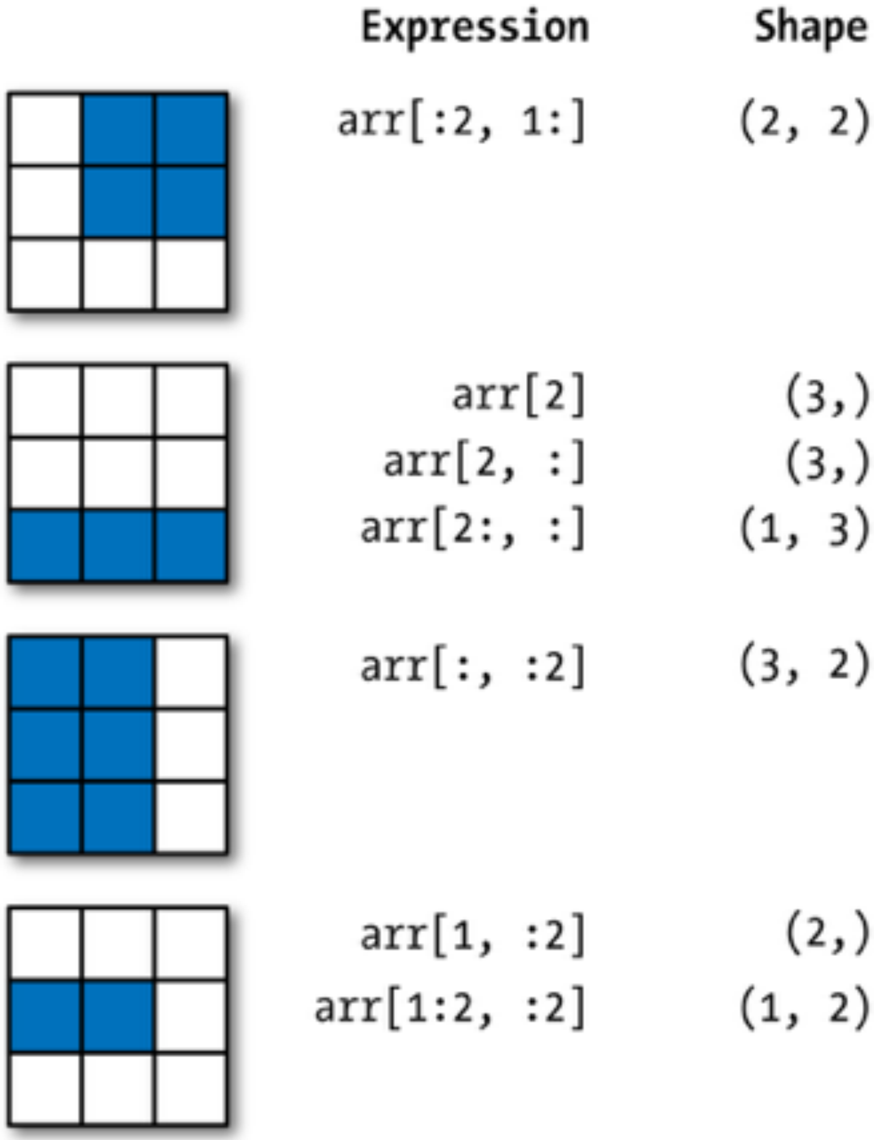
- Arrays are structured according to an index
- An element’s place within an array is written as array[d1, d2, d3…dn]
- In a matrix, this would look like array[row,column]
- You can also refer to columns with the argument (array, axis = n) in operations like sum or mean



(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	(4,2)	(4,3)	(4,4)

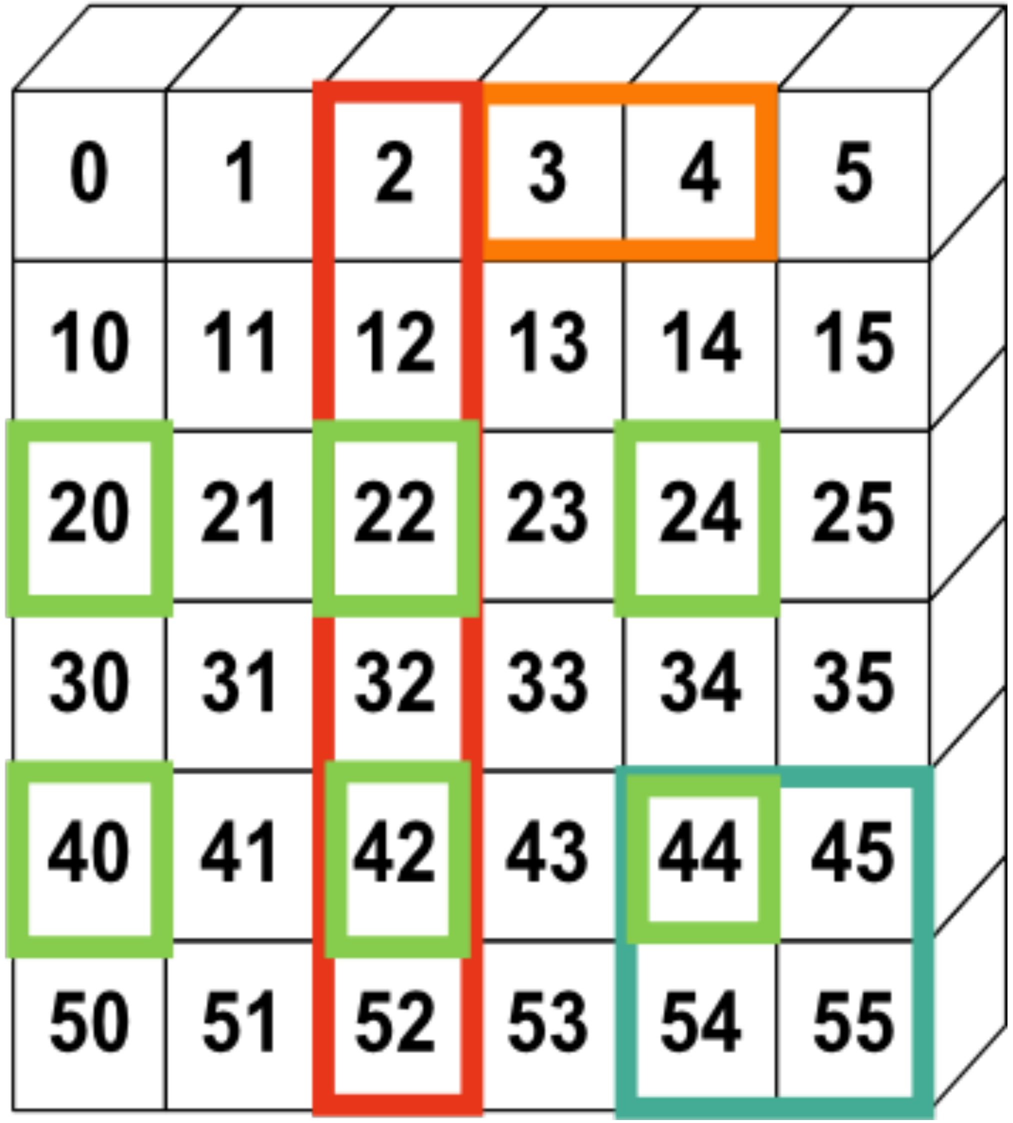
SLICING ARRAYS

- By giving ranges within an index, you can take “slices” of an array
- Dimensions are divided by commas
- min and max of a slice divided by a colon (adding a third colon represents “step”)
- Any information left blank defaults to “all”



Expression	Shape
`arr[:2, 1:]`	(2, 2)
`arr[2]`	(3,)
`arr[2, :]`	(3,)
`arr[2:, :]`	(1, 3)
`arr[:, :2]`	(3, 2)
`arr[1, :2]`	(2,)
`arr[1:2, :2]`	(1, 2)

PRACTICE SLICING ARRAYS



0	1	2	3	4	5
10	11	12	13	14	15
20	21	22	23	24	25
30	31	32	33	34	35
40	41	42	43	44	45
50	51	52	53	54	55

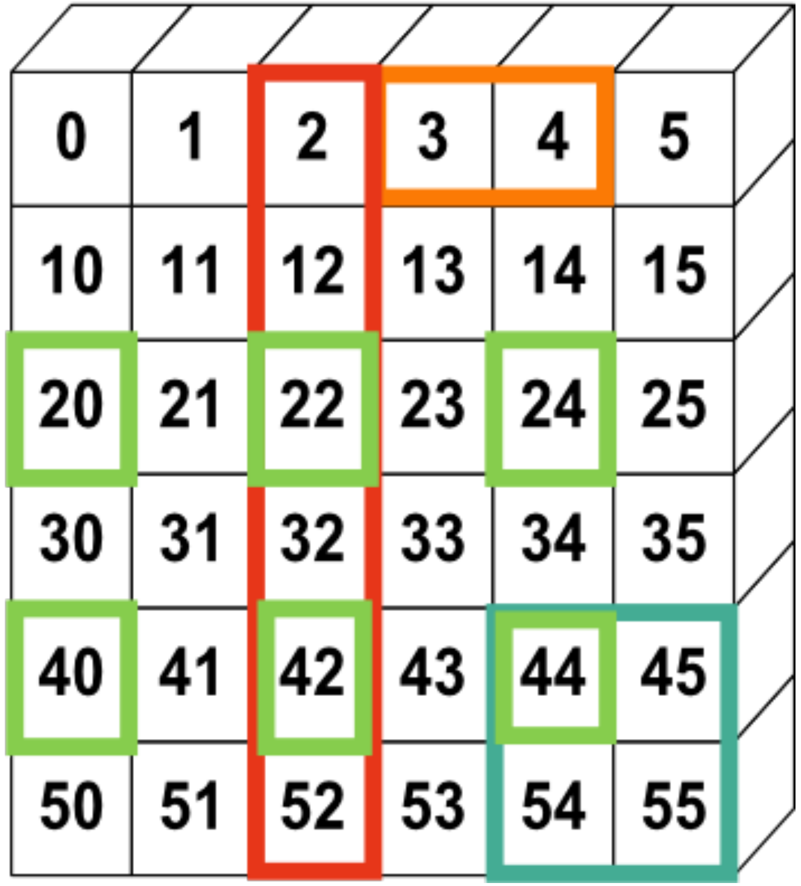
PRACTICE SLICING ARRAYS

```
>>> a[0,3:5]
array([3,4])
```

```
>>> a[4:,4:]
array([[44, 45],
       [54, 55]])
```

```
>>> a[:,2]
array([2,12,22,32,42,52])
```

```
>>> a[2::2,::2]
array([[20,22,24]
       [40,42,44]])
```



0	1	2	3	4	5
10	11	12	13	14	15
20	21	22	23	24	25
30	31	32	33	34	35
40	41	42	43	44	45
50	51	52	53	54	55

SHAPING ARRAYS

- `.shape`
  - Tells you dimensions of an array

- `.reshape(new shape as tuple)`
  - Allows you to change array dimensions

- `numpy.transpose(array)`
  - Transposes the array

- `numpy.ravel(array)`
  - Flattens the array

MATRICES VS ARRAYS IN NUMPY

- Numpy matrices are strictly 2-dimensional, while numpy arrays (ndarrays) are N-dimensional
- Matrix objects are a subclass of ndarray, so they inherit all the attributes and methods of ndarrays.
- Matrices provide a convenient notation for matrix multiplication: if a and b are matrices, then a*b is their matrix product
- Both matrix objects and ndarrays have .T to return the transpose, but matrix objects also have .H for the conjugate transpose, and .I for the inverse.
- By default, use arrays, since they behave more consistently

CHECK-IN

- What is the difference between a vector, a matrix, and an array?
- Is a vector an array?
- Is an array a vector?
- Is a matrix an array?
- Is an array a matrix?
- Can arrays have both numeric and string values?

TITLE

DEMO: TOPIC

ARRAYS WITH NUMPY

- See Jupyter Notebook

ARRAYS AND NUMPY

GUIDED PRACTICE: TOPIC

ACTIVITY: TITLE OF ACTIVITY

EXERCISE

DIRECTIONS

1. Create an array
2. Reshape the array
3. Transpose the array
4. Create another array in the same shape
5. Add the two arrays together
6. Subtract one array from the other
7. Find the dot product of the arrays

INDEPENDENT PRACTICE

ARRAY STATS

ACTIVITY: TITLE OF ACTIVITY

EXERCISE

DIRECTIONS

1. Create an array of random numbers with 10 rows and 10 columns
2. Write a function that iterates through the rows and reports the largest sum, mean and standard deviation
3. Repeat step 2, but iterate through the columns
4. Create a second 10x10 array and calculate the sum, difference, and dot product of the two arrays

DELIVERABLE

A .py file or python notebook submitted as a pull request to this repository (make sure the file has your name or initials to make it unique)

GO ARRAY

WRAP UP

- What do arrays represent?
- What data can be stored in arrays?
- What math operations can you perform on arrays?
- How can you select a certain portion of an array?

TITLE

BEFORE NEXT CLASS

BEFORE NEXT CLASS

DUE DATE

- Homework: Complete as many exercises from the following website as you can (Choose at least 8) and submit your code (including great comments) as a pull request to this repo

- https://github.com/rougier/numpy-100/blob/master/100%20Numpy%20exercises%20no%20solution.md

TITLE

CREDITS

TITLE

CITATIONS

- https://docs.oracle.com/cd/A57673_01/DOC/api/doc/PC_22/ch10.htm
- http://www.truebasic.com/node/1038
- http://www.slideshare.net/mikeranderson/2013-1114-enter-thematrix
- http://stackoverflow.com/questions/12024820/danger-of-mixing-numpy-matrix-and-array
- http://www.maplesoft.com/support/help/maple/view.aspx?path=SignalProcessing%2FDotProduct
- http://tutorial.math.lamar.edu/Classes/CalcII/DotProduct.aspx

TITLE

Q & A