ENGR 1330: Computational Thinking with Data Science

Lesson 6: NumPy In Python

Dinesh S. Devarajan
Whitacre College of Engineering
Texas Tech University

Topic Outline

- NumPy library

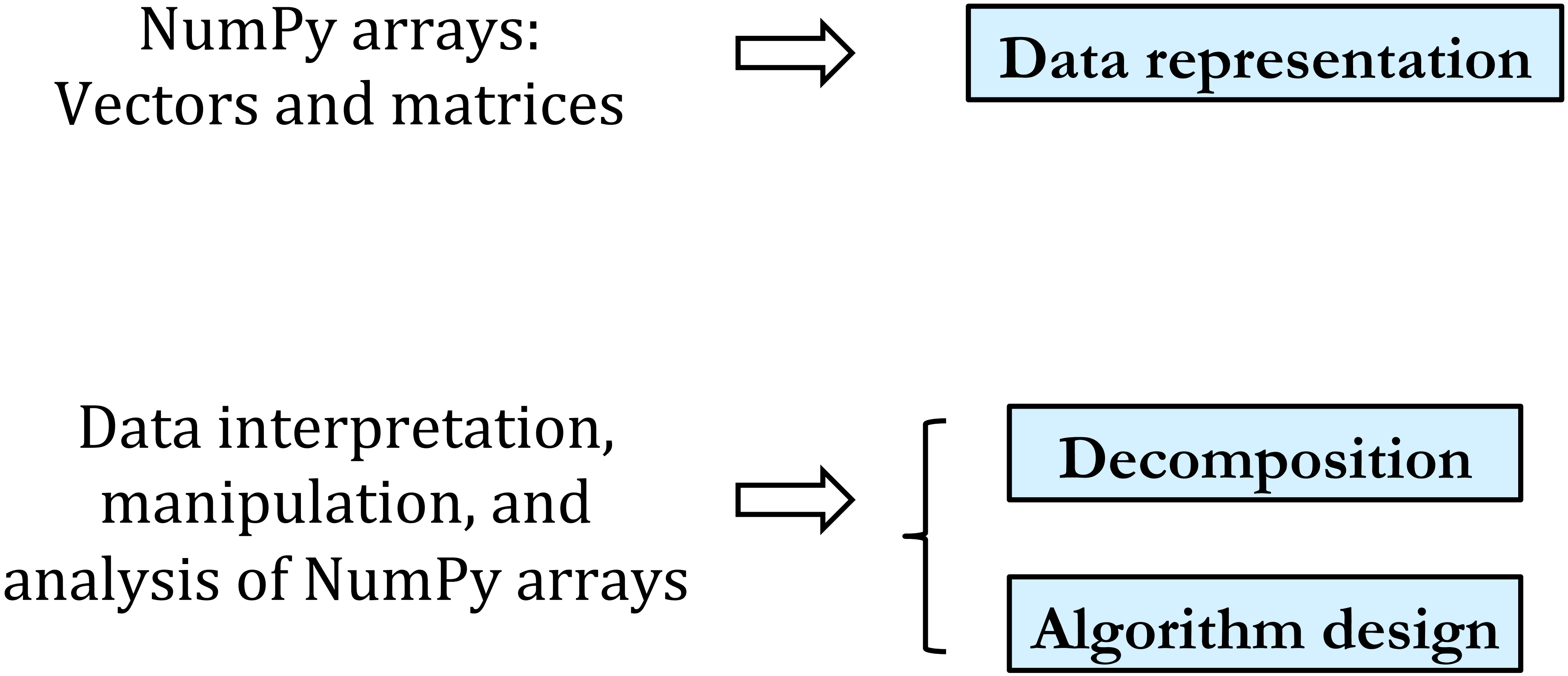
  ✓ Data representation: Arrays - vectors and matrices

  ✓ Data operations: Mathematical operations, indexing, selection, and copying

Objectives

- To be able to represent data in different forms via the NumPy library
- To be able to access data within a NumPy array
- To be able to perform basic mathematical functions on the NumPy arrays

Computational Thinking Concepts



NumPy arrays:
Vectors and matrices

⇨ Data representation

Data interpretation,
manipulation, and
analysis of NumPy arrays

⇨ Decomposition

Algorithm design

Library Setup

- Built-in with CoCalc

- You do not have to do any extra steps to install the library in Python

NumPy in Python

NumPy

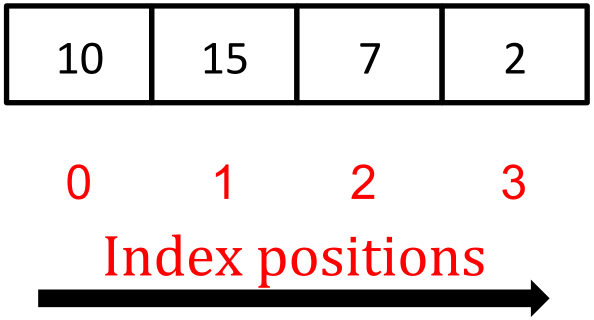
- NumPy: Numerical Python
- Foundational library for scientific computing
- All data science libraries rely on NumPy as one of their building blocks

Features of NumPy

- Features:
  - ✓ Provides a fast and efficient multi-dimensional array object called ‘ndarray’ (n-dimensional array) – NumPy arrays
  - ✓ Functions for performing computations with arrays and mathematical operations between arrays
  - ✓ Linear algebra operations and random number generation

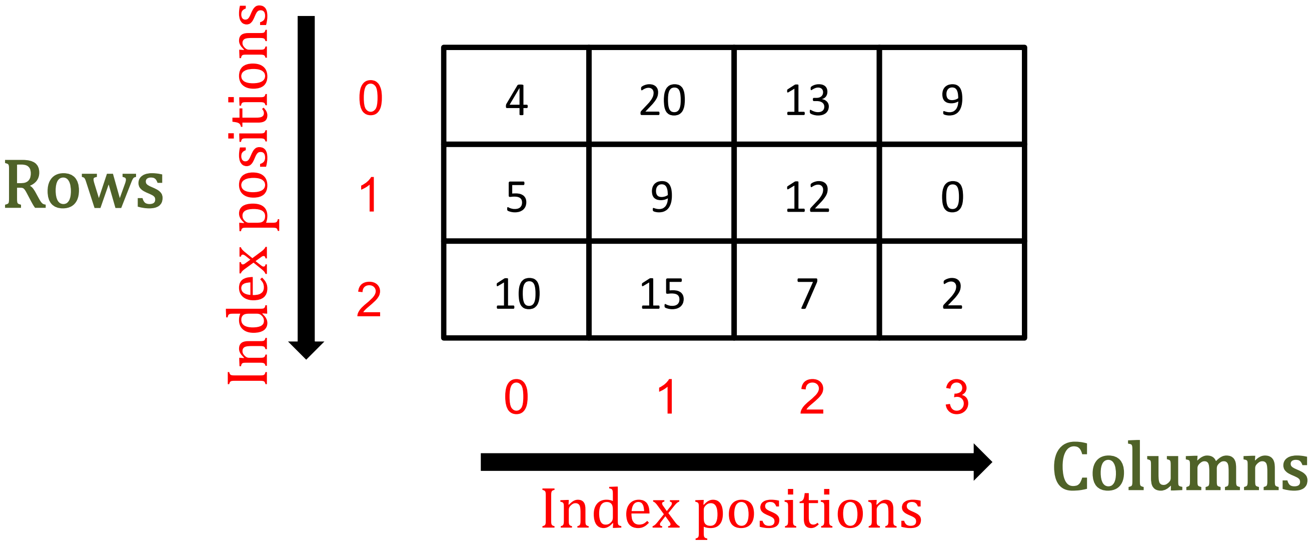
Multi-dimensional Array

- 1D array



10	15	7	2
0	1	2	3

- 2D array

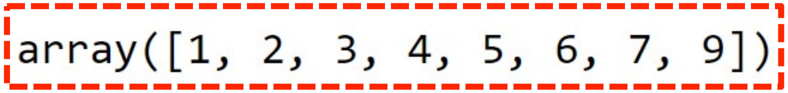


	0	1	2	3
0	4	20	13	9
1	5	9	12	0
2	10	15	7	2

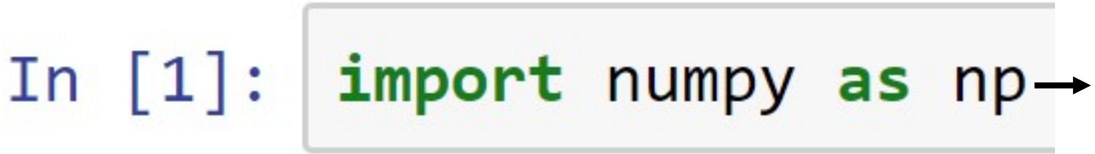
NumPy Arrays

- NumPy arrays can be 1-dimensional (1D) or 2-dimensional (2D)

- Creating a 1D array: Vector

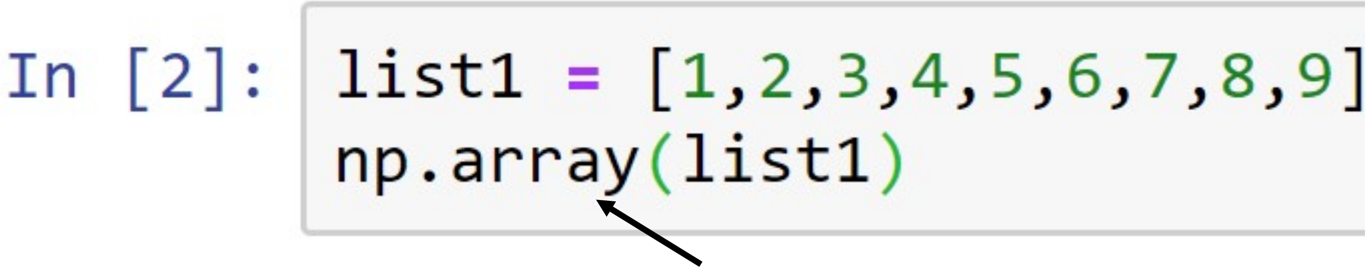


```
array([1, 2, 3, 4, 5, 6, 7, 9])
```



```
In [1]: import numpy as np
```

Importing NumPy library



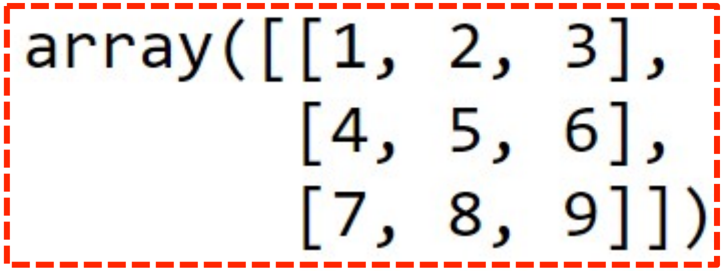
```
In [2]: list1 = [1,2,3,4,5,6,7,8,9]
        np.array(list1)
```

(Demo)

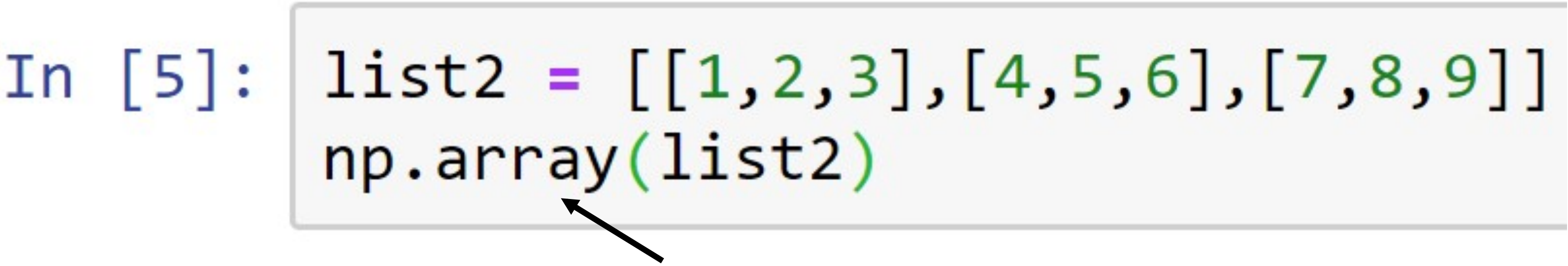
Function to create a NumPy array

NumPy Arrays

- Creating a 2D array: Matrix



```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```



```
In [5]: list2 = [[1,2,3],[4,5,6],[7,8,9]]
        np.array(list2)
```

Function to create a NumPy array

- What will be the shape of the above 2D NumPy array?

(Demo)

NumPy Arrays

- Other functions to create NumPy arrays easily

  ✓ arange( ): Returns evenly spaced array elements

  ✓ linspace( ): Returns evenly spaced array elements

  ✓ zeros( ): Returns an array of zeros

  ✓ ones( ): Returns an array of ones

  ✓ eye( ): Returns an identity matrix

  ✓ random.randint( ): Returns random integers

(Demo)

Arrays: Basic Operations

- Functions to do basic operations on NumPy arrays

  ✓ min( ): Returns minimum value in an array

  ✓ max( ): Returns maximum value in an array

  ✓ argmin( ): Returns minimum value position in an array

  ✓ argmax( ): Returns maximum value position in an array

  ✓ reshape( ): Reshaping an array to a specific shape

  ✓ sort( ): Sorting an array in ascending order

  ✓ sum( ): Summing the array elements

(Demo)

Arrays: Mathematical Operations

- Functions to do mathematical operations on NumPy arrays
  - ✓ sqrt( ): Returns square root of array elements
  - ✓ exp( ): Returns exponential of array elements
  - ✓ sin( ): Returns trigonometric sine of array elements
  - ✓ cos( ): Returns trigonometric cosine of array elements
  - ✓ log( ): Returns natural logarithm of array elements
  - ✓ log10( ): Returns base 10 logarithm of array elements

(Demo)

Arrays: Indexing, Selection, & Copying

- Indexing: An important step in manipulating and analyzing arrays

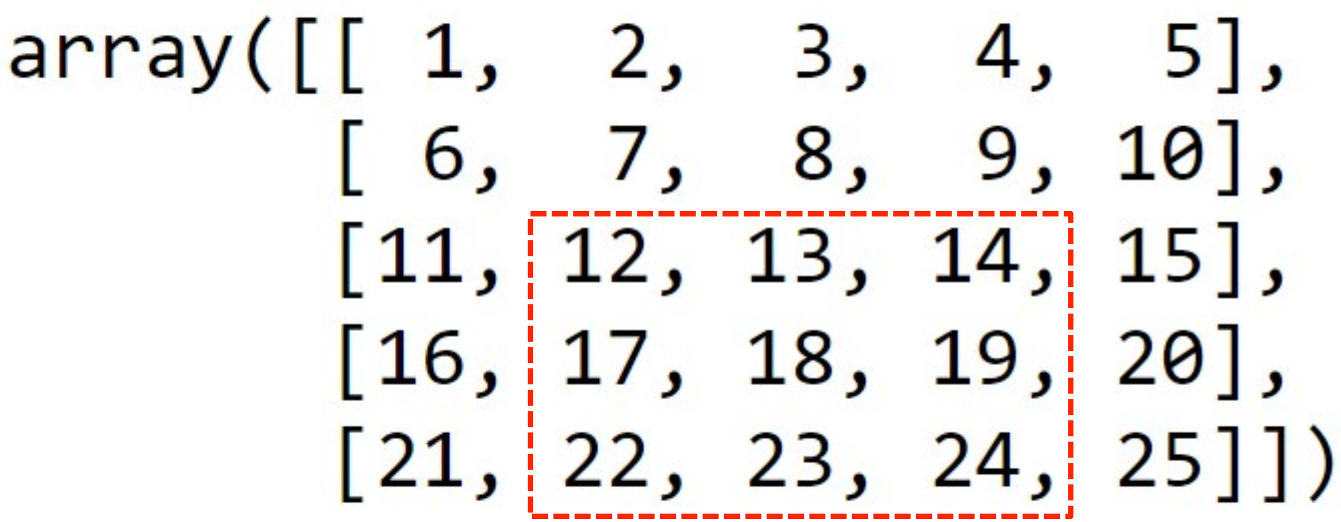
- Conditional selection: Selecting array elements based on specific conditions typed using conditional operators

- Copying: Always use the copy( ) function to copy arrays and to preserve the original array

(Demo)

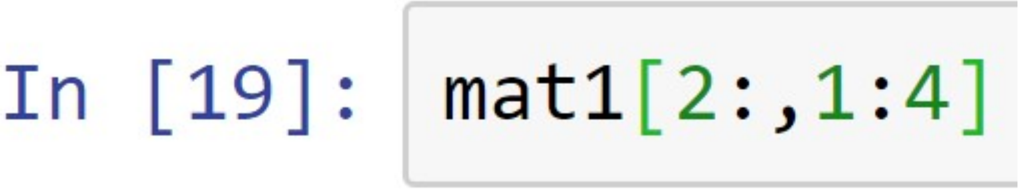
Discussion Exercise

mat1 =



```
array([[ 1,  2,  3,  4,  5],
       [ 6,  7,  8,  9, 10],
       [11, 12, 13, 14, 15],
       [16, 17, 18, 19, 20],
       [21, 22, 23, 24, 25]])
```

- How would you index and slice the elements within the red-dashed box above from the matrix named ‘mat1’?



```
In [19]: mat1[2:,1:4]
```

(Demo)

Summary

- Concepts of representing data in the form of NumPy arrays are covered

- Concepts of interpreting, manipulating, and analyzing data within NumPy arrays are covered