Queen Mary
University of London

QHP4701
Introduction to Data Science Programming

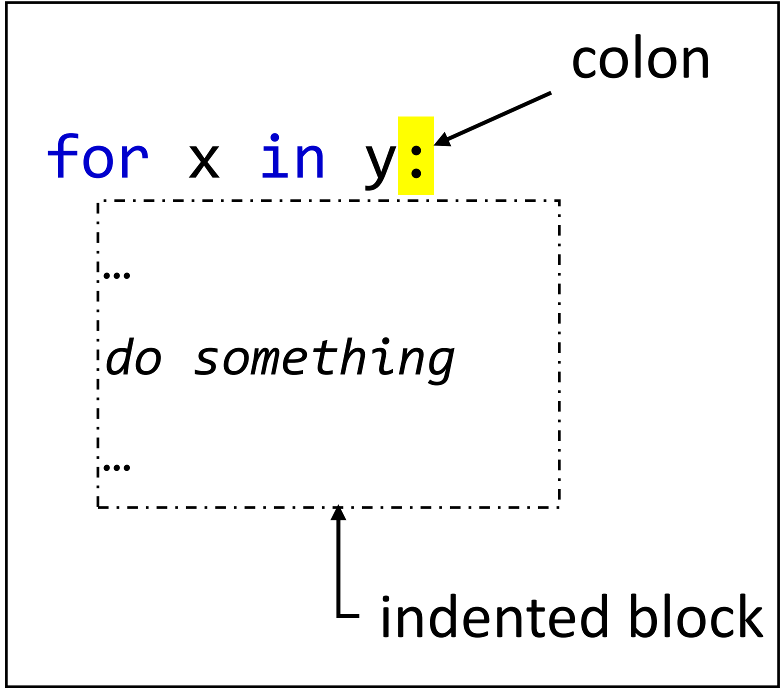
Vectors, Matrices, and Numpy Arrays

Lecturer: Nikesh Bajaj, PhD
School of Physical and Chemical Sciences
http://nikeshbajaj.in

Lecture Outline

- For-loop and Linear Algebra in Python
- Numpy Arrays: Vectors, Matrices
  - Creating, indexing, manipulating, methods
- Linear Algebra with Numpy
  - Dot product, Matrix multiplication, rank, inverse,
- Data in Numpy Arrays
  - Image, Audio, Speech, CSV
- Numpy Methods

Loops: for-loop



colon

```
for x in y:
    ...
    do something
    ...
```

indented block

- **For-loop:** In programming languages, for repeating operation(s), a loop is used, which iterate over a sequence (goes over each element of a sequence)

```python
fruits = ["apple", "banana", "cherry"]
for fruit in fruits:
  print(fruit)
```

```python
numbers = [1,2,3,4,5]
squares = []
for num in numbers:
    squares.append(num ** 2)

print(squares)
```

```python
for i in range(6):
  print(i)
```

Linear Algebra in Python

- In Python, a vector can be created as a list of numbers
  - a = [1, 2, 3, 4, 5]
  - b = [5, 6, 7, 8, 9]

$$a = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix} \quad b = \begin{bmatrix} 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{bmatrix}$$

How to compute?

Multiplication of scalar with vector

- 2a

```
a_2 = []
for x in a:
    a_2.append(x*2)
```

Linear Algebra in Python

- Addition of two vectors

c = a + b

```python
c = []
for i in range(len(a)):
    c.append(a[i]+b[i])
```

- a = [1, 2, 3, 4, 5]
- b = [5, 6, 7, 8, 9]

Linear Algebra in Python

- Dot-product

c = a·b

$$c = \sum a_i b_i$$

- a = [1, 2, 3, 4, 5]
- b = [5, 6, 7, 8, 9]

```
c = 0
for i in range(len(a)):
    c = c + a[i]*b[i]
```

Linear Algebra in Python

- Magnitude of a vector |a|

- a = [1, 2, 3, 4, 5]

$$c = |a| = \sqrt{\sum a_i^2}$$

```
c = 0
for i in range(len(a)):
    c = c + a[i]*2
c = c**(1/2)
```

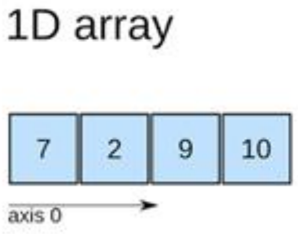
Is there a mistake in this computation?

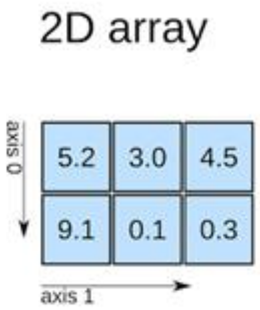
Lecture Outline

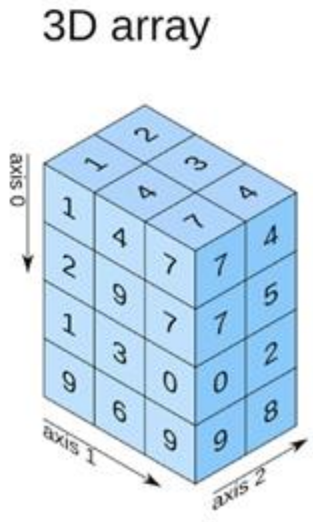
- For-loop and Linear Algebra in Python
- Numpy Arrays: Vectors, Matrices
    - Creating, indexing, manipulating, methods
- Linear Algebra with Numpy
    - Dot product, Matrix multiplication, rank, inverse,
- Data in Numpy Arrays
    - Image, Audio, Speech, CSV
- Numpy Methods

NumPy in Python

- NumPy in Python is a very basic library used for scientific computations. It is heavily used in Data Science.
- It provides arrays (vectors, matrices) as objects that supports linear algebra, sorting, selecting and manipulating data







...

NumPy in Python

- Anaconda comes with NumPy and several other libraries used for Data Science
- To use NumPy, first you have to load the library

```
import numpy
```

Or

```
import numpy as np
```

NumPy : Creating Arrays

Creating Arrays from sequences

- x1D = np.array([1, 2, 3, 4])
- x2D = np.array([[1, 2],

  [3, 4]])
- x3D = np.array([[[1, 2], [3, 4]],

  [[5, 6], [7, 8]]])

Check type and share and size

- type(x1D)
- x1D.shape
- x2D.shape
- x3D.shape
- x3D.size

NumPy : Creating Arrays

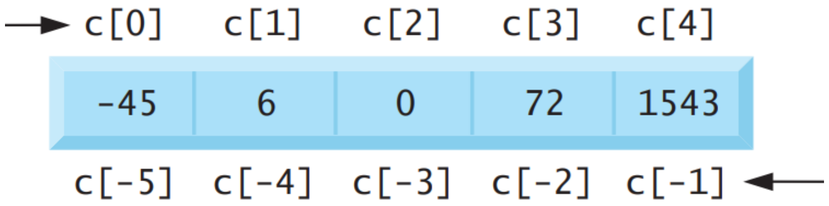
Creating Arrays from functions

x = np.arange(10)	Range from 0 to 9
x = np.arange(2, 10, 2) # can accept float	Range from 2 to 10 with step 2
x = np.linspace(0,10,10)	10 values from 0 to 10 including 0 and 10 (evenly spaced
x = np.eye(3)	Identity matrix of 3x3
x = np.diag([1,2,3])	Diagonal matrix
x = np.zeros([2,3])	Matrix of zeros
x = np.ones([2,3])	Matrix of ones
x = np.random.rand(10)	Random 1d matrix of 10 values
x = np.random.randn(10)	
x = np.random.randint(0, 5, 10)	

NumPy : Indexing

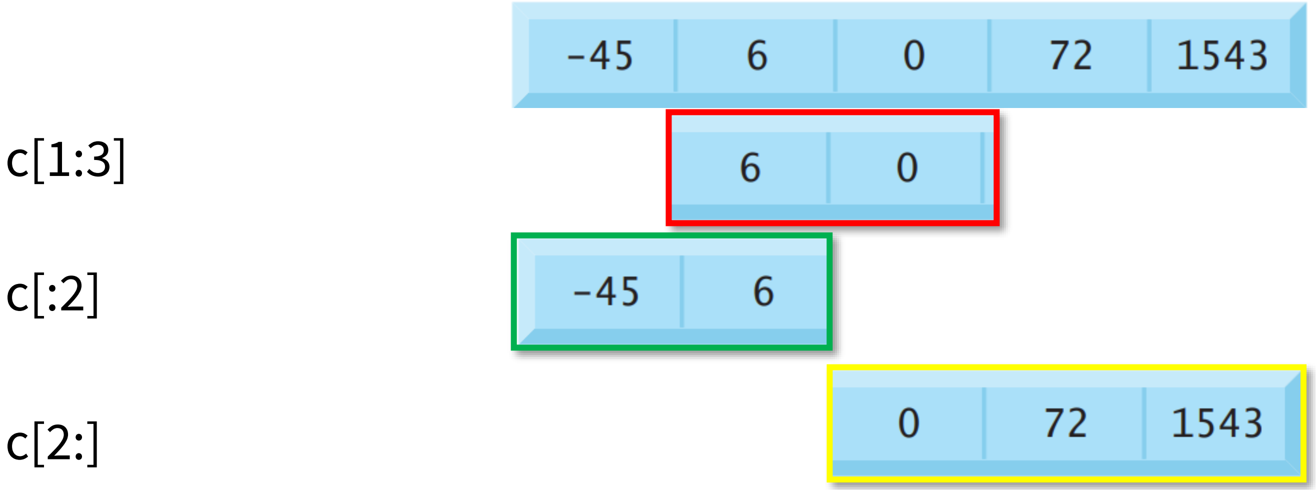
Indexing in Numpy Array is ***similar to list***

- c = np.array([-45, 6, 0, 72, 1543])



Selecting element(s):

- Slicing



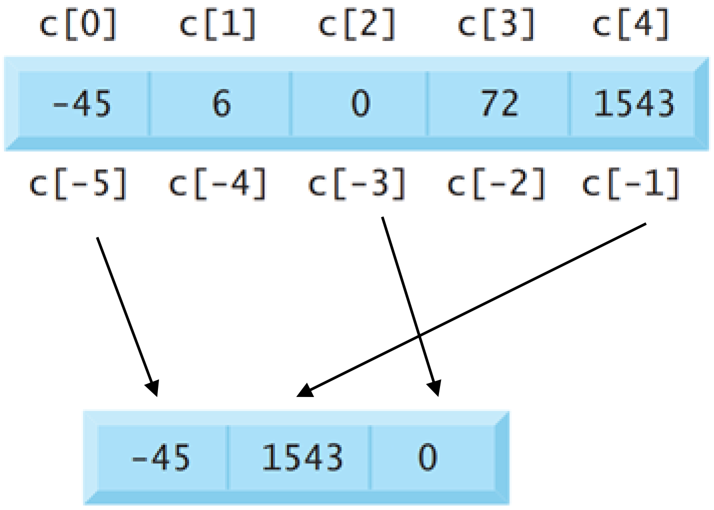
Excluding

c[starts: end]
c[start:end:step]
c[::step]

NumPy : Indexing

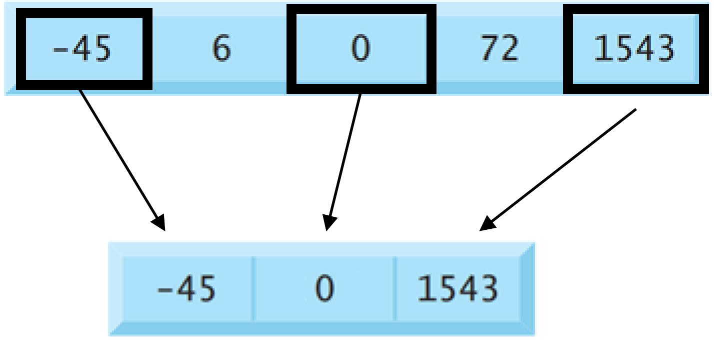
NumPy support more ways to select elements

- c = np.array([-45, 6, 0, 72, 1543])



Integer array indexing

- Idx = [0,4, 2]
- c[idx]
- array([-45, 1543,   0])

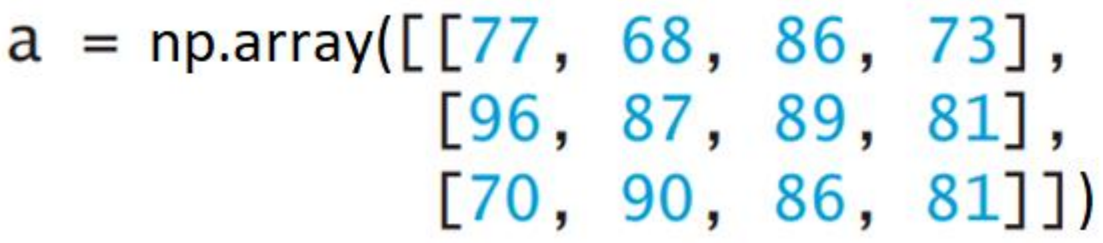


Boolean array indexing

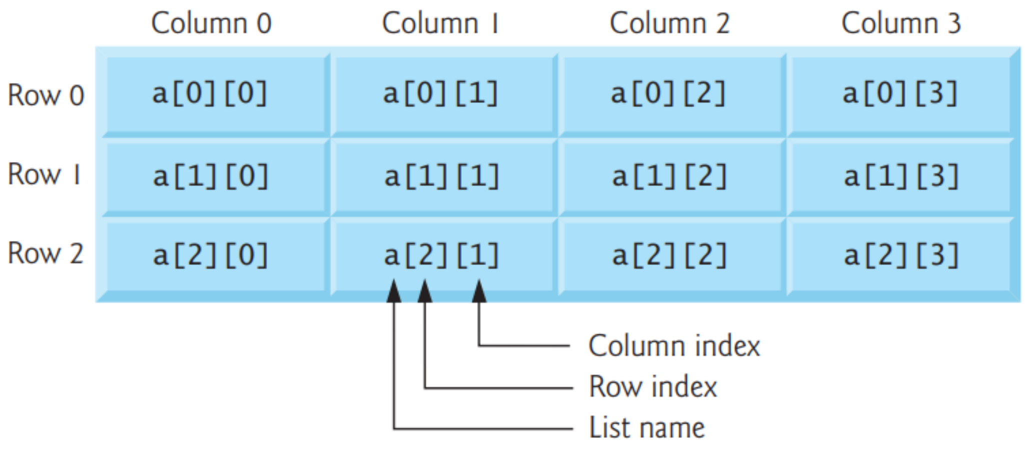
- idx = [True, False, True , False, True]
- c[idx]
- array([-45,   0, 1543])

NumPy : Indexing

- Similar to List of List, with additional use



```
a = np.array([[77, 68, 86, 73],
              [96, 87, 89, 81],
              [70, 90, 86, 81]])
```



	Column 0	Column 1	Column 2	Column 3
Row 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Row 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

- a[1,1]
- a[0]   -- 0th raw
- a[:, 2]   -- 2nd column
- a[:2, :2]
- a[:, 1:3]

Lecture Outline

- For-loop and Linear Algebra in Python
- Numpy Arrays: Vectors, Matrices
  - Creating, indexing, manipulating, methods
- Linear Algebra with Numpy
  - Dot product, Matrix multiplication, rank, inverse,
- Data in Numpy Arrays
  - Image, Audio, Speech, CSV
- Numpy Methods

NumPy : Linear Algebra

- Recall the Linear Algebra operations with List, NumPy make is very easy
- a = np.array([1,2,3,4,5])
- b = np.array([5,6,7,8,9])

$$a = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix} \qquad b = \begin{bmatrix} 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{bmatrix}$$

- Addition: c = a+b
- Scalar multiplication c = 2*a
- Dot product c = a.dot(b)
- Squared c = a**2
- Magnitude c = sum(a**2)**(1/2) c = np.linalg.norm(a)

NumPy : Linear Algebra
Matrix Multiplication

- A = [[1, 2, 3],
  [2, 0, 1],
  [1, 0, 1]]

- B = [[2, 2, 3],
  [1, 1, 0],
  [2, 1, 0]]

compute

- C = A × B

With NumPy:

- C = A@B
- C = A.dot(B)

Lecture Outline

- For-loop and Linear Algebra in Python
- Numpy Arrays: Vectors, Matrices
  - Creating, indexing, manipulating, methods
- Linear Algebra with Numpy
  - Dot product, Matrix multiplication, rank, inverse,
- Data in Numpy Arrays
  - Image, Audio, Speech, CSV
- Numpy Methods

Visualise with matplotlib

- In Python Matplotlib is used to visualise and plot different kind of data.
- Matplotlib will be covered in future session with great details.
- For now, to visualise arrays, we will use a few functionalities of matplotlib

```
import matplotlib.pyplot as plt
```

Additional Libraries to load data

- For read different types of data, we need some additional libraries

```
import pandas as pd
import matplotlib.pyplot as plt
import scipy
```

Audio/Speech	.wav	Scipy	*scipy.io.wavfile.read*
Image	.png, .bmp, .jpg	Matplotlib	*plt.imread*
CSV	.csv	Pandas	*pd.read_csv*

Data in Numpy Arrays

- Audio File
  - To read an audio file, use *fs, x = scipy.io.wavfile.read(file_name)*
  - *Check type of x* *type(x)*

- Image File
  - To read image file, use *I = plt.imread(file_name)*
  - *Check type of I* *type(I)*

- CSV File
  - To read csv file of numbers, use *data = pd.read_csv(file_name)*
  - *Check type of data* *type(data)*

Data in Numpy Arrays

Use NumPy operations on audio, Image, and CSV columns

Lecture Outline

- For-loop and Linear Algebra in Python
- Numpy Arrays: Vectors, Matrices
  - Creating, indexing, manipulating, methods
- Linear Algebra with Numpy
  - Dot product, Matrix multiplication, rank, inverse,
- Data in Numpy Arrays
  - Image, Audio, Speech, CSV
- Numpy Methods

NumPy : Methods

- a = np.array([1,2,3,4,5,6])

$$a = \begin{bmatrix}1\\2\\3\\4\\5\\6\end{bmatrix}$$

• Reshape:	a.reshape(2,3)
• Transpose	a.T
• Sum, max, min, mean	a.sum(), a.max(), a.min(), a.mean()
• Comparison	a ==3, a>3
• Unique	
• Len	

- Next !!!

  - 3.1: More on NumPy
  - 3.2: Lab with Numpy Array

Queen Mary
University of London