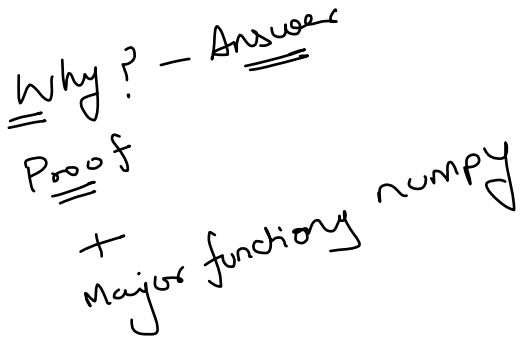
upGrad

#LifeKoKaroLift



Introduction to NumPy

Course : Data Science

Lecture On : NumPy

Instructor : Sumit Shukla

upGrad

What we will cover in this session?

1. Introduction to NumPy library
2. Why we NumPy
3. How NumPy is more fast than Python Data Structures like list
4. Python Demonstration

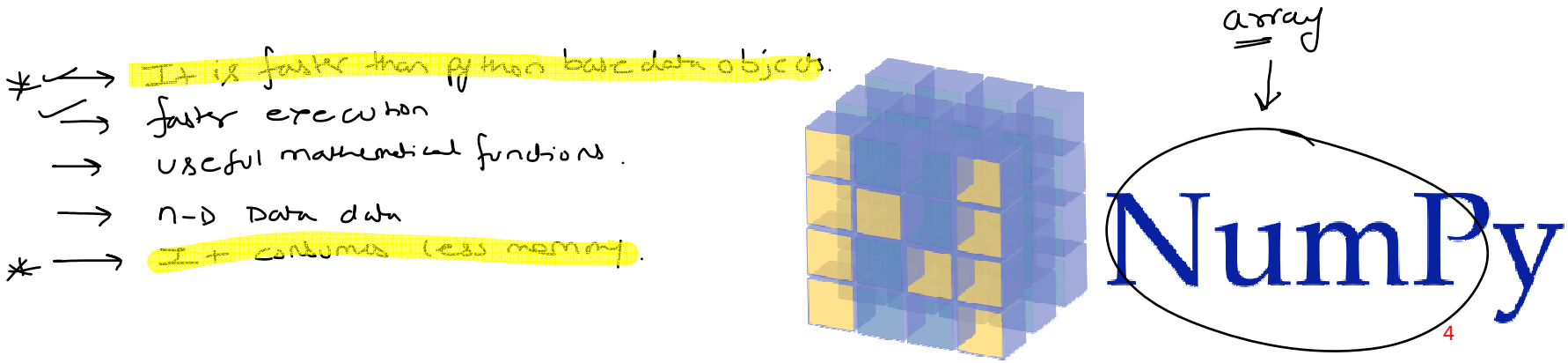
Introduction to NumPy

upGrad

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- A powerful N-dimensional array object
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.



Why we need NumPy

upGrad

Constraints

→ It can hold data of one kind (homogenous)
→ You need to define the length, data type if you wanted to create an empty numpy array.
→ Immutable

- Performance - they are faster than lists
- Need - Multidimensional array
- Functionality - NumPy have optimized functions such as linear algebra operations built in.

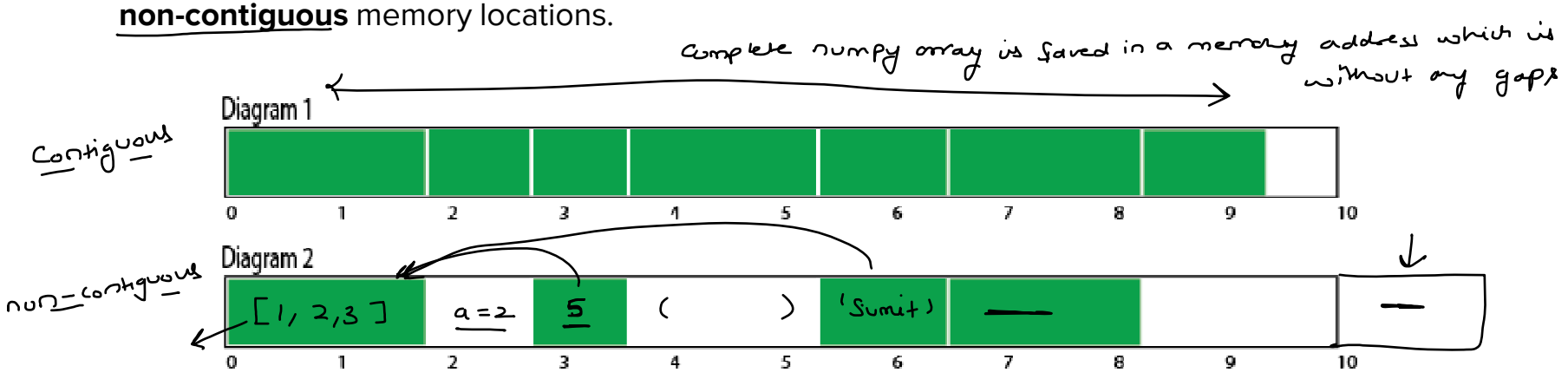
NumPy arrays are made to be created as homogeneous arrays, considering the mathematical operations that can be performed on them. It would not be possible with heterogeneous data sets.

How NumPy is faster than other Data Structures?

upGrad

memory a = np.array( )

1. An array is a collection of homogeneous data-types which are stored in **contiguous memory** locations, on the other hand, a list in Python is collection of heterogeneous data types stored in **non-contiguous** memory locations.



l = [1, 2, 3] a=2

l.append(5)

lists → mutuable.

→ as we will add new elements to the list, the next available memory block will be assigned to the new element

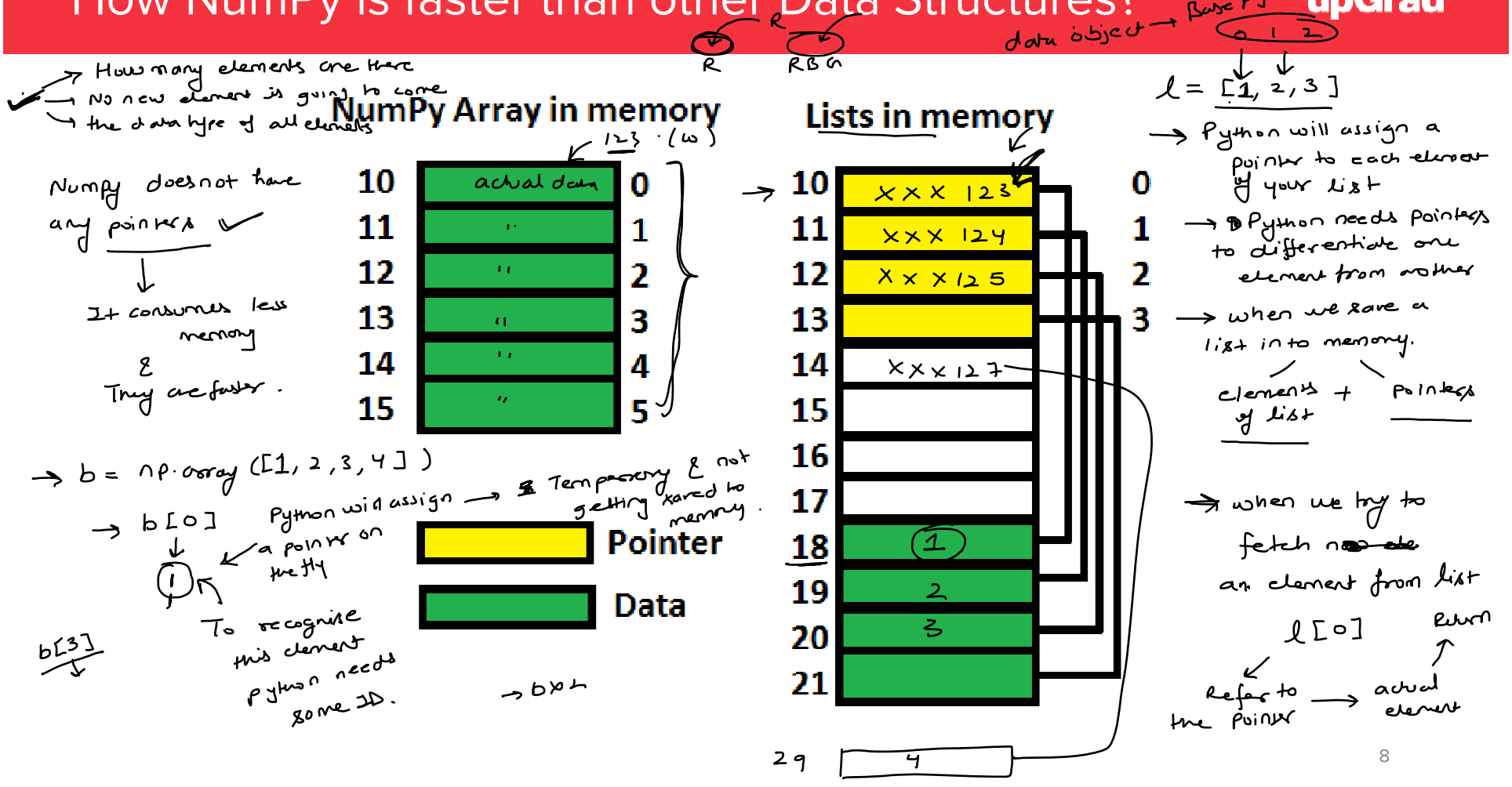
How NumPy is faster than other Data Structures?

upGrad

1. While declaring a NumPy array, it is necessary to describe the shape and the dtypes.
2. But this is not true in the case of the lists. While creating a list, you need not to specify how many elements you will append to your list nor do you have to define a data type. This is because the data in a lists is stored in a form of a pointers which describes the reference to the memory location. So, once you add a new data to Python lists, the lists will first create it's pointer and then it will save the data into the available memory location.
3. When you retrieve the first element in your list, python is taking two steps: First, retrieve the pointer. Second, go to the memory location of the pointer to finally get the object you want.

How NumPy is faster than other Data Structures?

upGrad



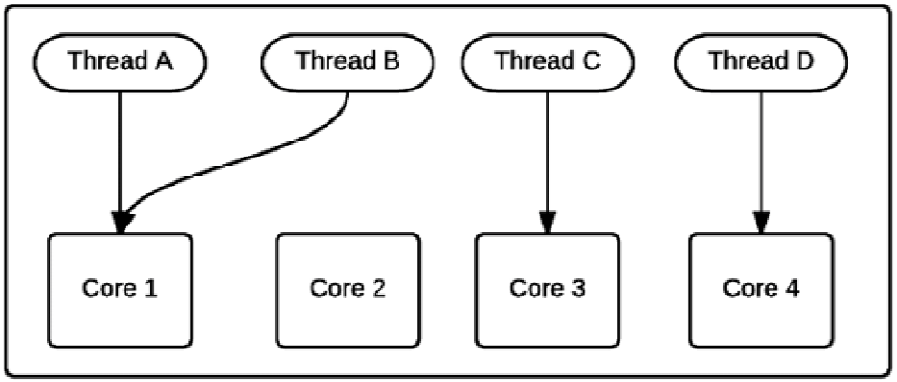
How NumPy is faster than other Data Structures?

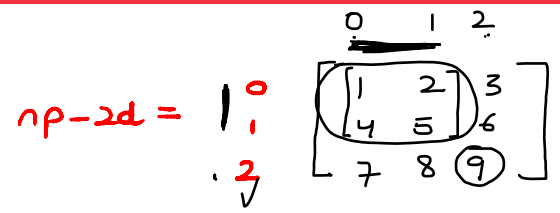
upGrad

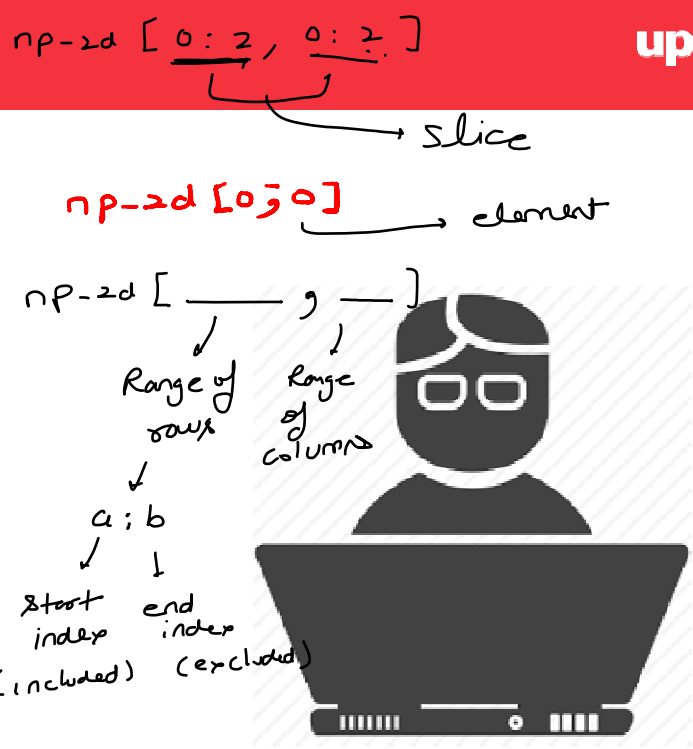
Apart from a least time and space complexity, we have more advantages using NumPy arrays:

2. The NumPy package breakdowns a task into multiple fragments, and then processes all the fragments parallelly.

3. The NumPy package integrates C, C++ and Fortran codes in Python, these programming languages have very less execution time as compared to python.







Let's Code

np-2d [0:1 , 0:1]

↓

np-2d [0, 0]

↓

1

upGrad

#LifeKoKaroLift

Thank You!