DATA FOLKZ
CATAPULT DATA LEADERS

What is NumPy?

DATA FOLKZ
CATAPULT DATA LEADERS

- Python is a fabulous language
  - Easy to extend
  - Great syntax which encourages easy to write and maintain code
  - Incredibly large standard-library and third-party tools
- No built-in multi-dimensional array (but it supports the needed syntax for extracting elements from one)
- NumPy provides a **fast** built-in object (ndarray) which is a multi-dimensional array of a homogeneous data-type.

DATA FOLKZ
CATAPULT DATA LEADERS

Overview of NumPy

N-D ARRAY (NDARRAY)

- N-dimensional array of rectangular data
- Element of the array can be C-structure or simple data-type.
- Fast algorithms on machine data-types (int, float, etc.)
- Arrays are used to store multiple values in one single variable.Python does not have built-in support for Arrays, but Python lists can be used instead .

NumPy Array

- A NumPy array is an N-dimensional homogeneous collection of "items" of the same "kind". The kind can be any arbitrary structure and is specified using the data-type.

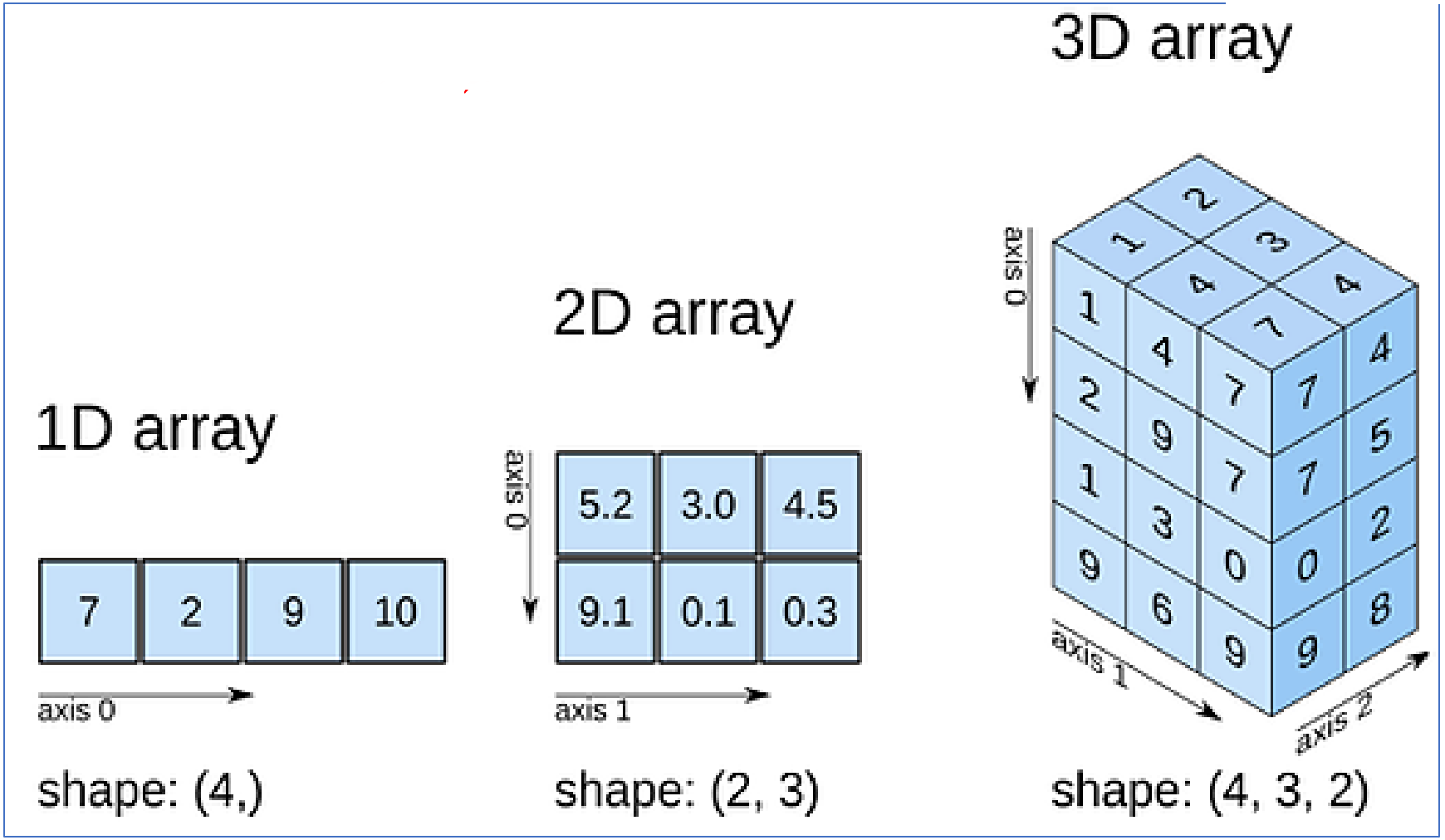
Command

```
np.array([1,2,3])
```

NumPy Array

1
2
3

Dimensions in Arrays



Dimensions in Arrays

DATA FOLKZ
CATAPULT DATA LEADERS

- **0-D Arrays** : 0-D arrays, or Scalars, are the elements in an array. Each value in an array is a 0-D array.

  Input : arr = np.array(42)

  Output : 42

- **1-D Arrays** : An array that has 0-D arrays as its elements is called uni-dimensional or 1-D array.

  Input : arr = np.array([1, 2, 3, 4, 5])

  Output: [1 2 3 4 5]

Dimensions in Arrays

DATA FOLKZ
CATAPULT DATA LEADERS

- **2-D Arrays :** An array that has 1-D arrays as its elements is called a 2-D array. These are often used to represent matrix or 2nd order tensors.

Input : arr = np.array([[1, 2, 3], [4, 5, 6]])

Output :
```
[[1 2 3]
 [4 5 6]]
```

- **3-D arrays :** An array that has 2-D arrays (matrices) as its elements is called 3-D array.These are often used to represent a 3rd order tensor.

Input : arr = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])

Output :
```
[[[1 2 3]
  [4 5 6]]
 [[1 2 3]
  [4 5 6]]]
```

Multi-Dimensional Arrays

An array can have any number of dimensions.
When the array is created, you can define the number of dimensions by using the ndmin argument.

MULTI-DIMENSIONAL ARRAYS

Input : a = array([[0, 1, 2, 3],
[10,11,12,13]])

Output : array([[0, 1, 2, 3],
[10,11,12,13]])

(ROWS,COLUMNS)

Input : a.shape
Output : (2, 4)

NUMBER OF DIMENSIONS

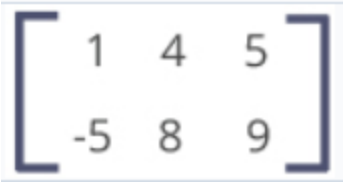
Input : a.ndims
Output :2

Matrix In Python

- A Python matrix is a specialized two-dimensional rectangular array of data stored in rows and columns. The data in a matrix can be numbers, strings, expressions, symbols, etc. Matrix is one of the important data structures that can be used in mathematical and scientific calculations.

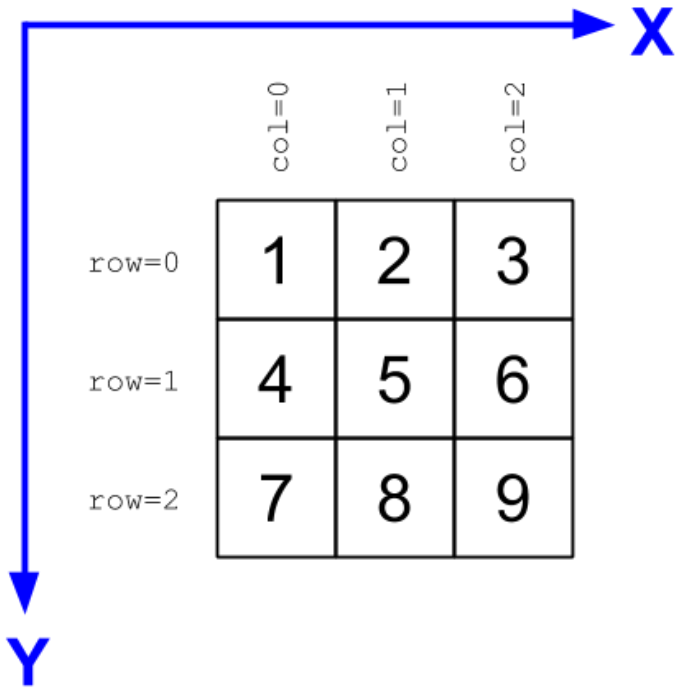
```
A = [[1, 4, 5],
     [-5, 8, 9]]
```

- Output :



Matrix Operation using Num py.Array()

- **The matrix operation that can be done is addition, subtraction, multiplication, transpose, reading the rows, columns of a matrix, slicing the matrix, etc.**



Array Slicing

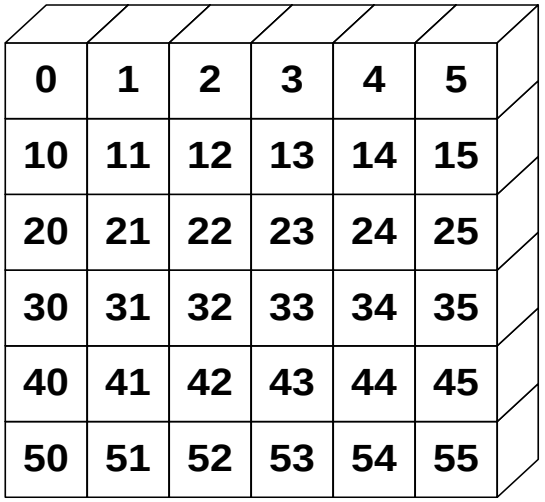
DATA FOLKZ
CATAPULT DATA LEADERS

Slicing in python means taking elements from one given index to another given index. We pass slice instead of index like this: [start:end]

SLICING WORKS MUCH LIKE STANDARD PYTHON SLICING :-

Input : a[0,3:5]
Output : array([3, 4])

Input : a[:,2]
Output :
array([2,12,22,32,42,52])



NumPy dtypes

Basic Type	Available NumPy types	Comments
Boolean	bool	Elements are 1 byte in size
Integer	int8, int16, int32, int64, int128, int	int defaults to the size of int in C for the platform
Unsigned Integer	uint8, uint16, uint32, uint64, uint128, uint	uint defaults to the size of unsigned int in C for the platform
Float	float32, float64, float, longfloat,	Float is always a double precision floating point value (64 bits). longfloat represents large precision floats. Its size is platform dependent.
Complex	complex64, complex128, complex	The real and complex elements of a complex64 are each represented by a single precision (32 bit) value for a total size of 64 bits.
Strings	str, unicode	Unicode is always UTF32 (UCS4)
Object	object	Represent items in array as Python objects.
Records	void	Used for arbitrary data structures in record arrays.

Click to add text

Statistics Array Methods

MEAN

Gives mean value of each column

```
a = array([[1,2,3],
           [4,5,6]], float)
```

Input : mean(a, axis=0)

Output : array([2.5,3.5,4.5])

STANDARD DEV./VARIANCE

Standard Deviation

Input : a.std(axis=0)

Output : array([1.5, 1.5, 1.5])

Variance :

Input :a.var(axis=0)

Output : array([2.25, 2.25, 2.25])

Summary of (most) array attributes/methods

BASIC ATTRIBUTES

- **a.dtype – Numerical type of array elements. float32, uint8, etc.**
- **a.shape – Shape of the array. (m,n,o,...)**
- **a.size – Number of elements in entire array.**
- **a.itemsize – Number of bytes used by a single element in the array.**
- **a.nbytes – Number of bytes used by entire array (data only).**
- **a.ndim – Number of dimensions in the array.**
- **a.T – Shorthand for a.transpose()**

Summary of (most) array attributes/methods

REDUCTION METHODS

All the following methods "reduce" the size of the array by 1 dimension by carrying out an operation along the specified axis. If axis is None, the operation is carried out across the entire array.

`a.sum(axis=None)` – **Sum up values along axis.**
`a.prod(axis=None)` – **Find the product of all values along axis.**
`a.min(axis=None)`– **Find the minimum value along axis.**
`a.max(axis=None)` – **Find the maximum value along axis.**
`a.mean(axis=None)` – **Find the mean (average) value along axis.**
`a.std(axis=None)` – **Find the standard deviation along axis.**
`a.var(axis=None)` – **Find the variance along axis.**
`a.any(axis=None)` – **True if any value along axis is non-zero. (or)**
`a.all(axis=None)` – **True if all values along axis are non-zero. (and)**