

Numpy

Basics of Numpy -

Quick Recap: Lists

Below are some of the properties of lists:

List

```
distance=[10,15,17,26]
```

```
time=[.30,.47,.55,1.20]
```

Collection of values

Multiple types (heterogeneous)

Add, remove, update

Limitations of Lists

You can change individual values in a list, but you cannot apply a mathematical operation over the entire list.

```
distance=[10,15,17,26]
time=[.30,.47,.55,1.20]
```

```
speed=distance/time
```

← Mathematical operation over the entire "distance" and "time" lists

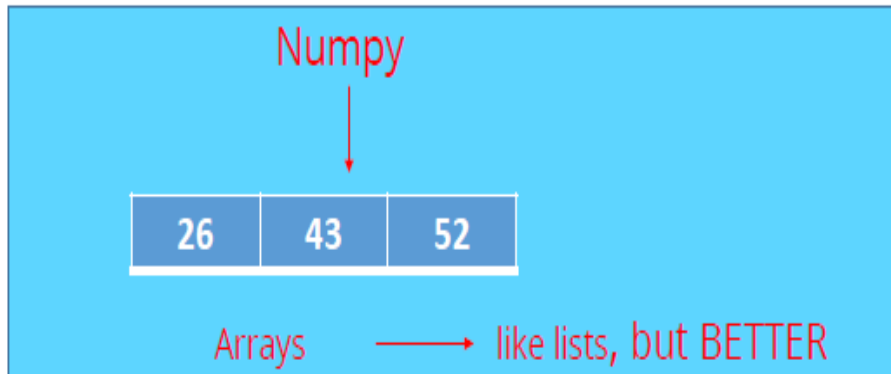
```
-----
TypeError                                 Traceback (most recent call last)
<ipython-input-37-b779bad68500> in <module>()
----> 1 speed=distance/time
```

```
TypeError: unsupported operand type(s) for /: 'list' and 'list'
```

← Error

Why NumPy

Numerical Python (NumPy) supports multidimensional arrays over which you can easily apply mathematical operations.



```
distance=[10,15,17,26]  
time=[.30,.47,.55,1.20]
```

```
import numpy as np
```

← Import NumPy

```
np_distance = np.array(distance)  
np_time=np.array(time)  
speed=np_distance/np_time
```

} Create "distance" and "time" NumPy arrays
← Mathematical function applied over the entire "distance" and "time" arrays

```
speed
```

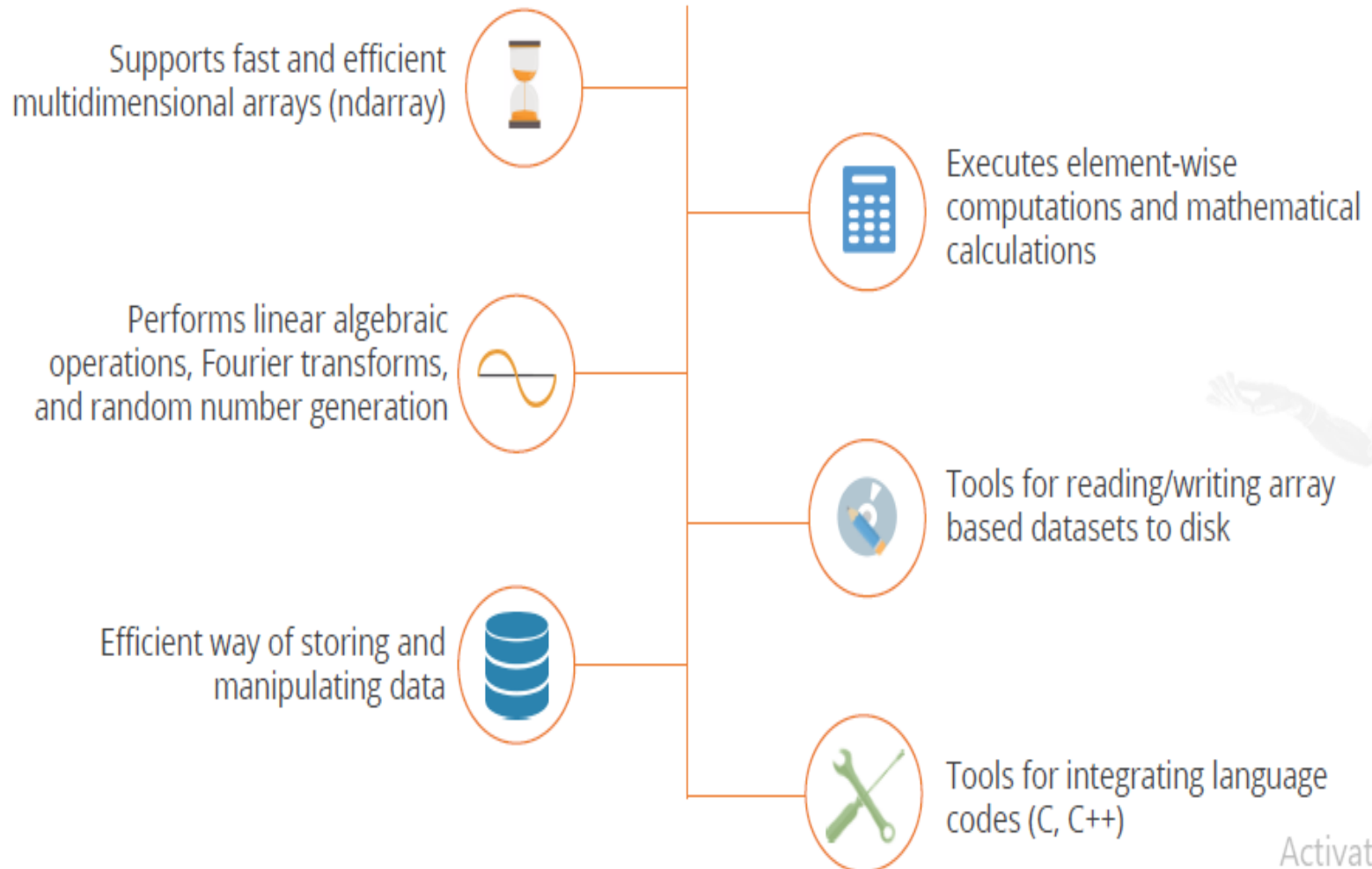
```
array([ 33.33333333,  31.91489362,  30.90909091,  21.66666667])
```

← Output

NumPy Overview

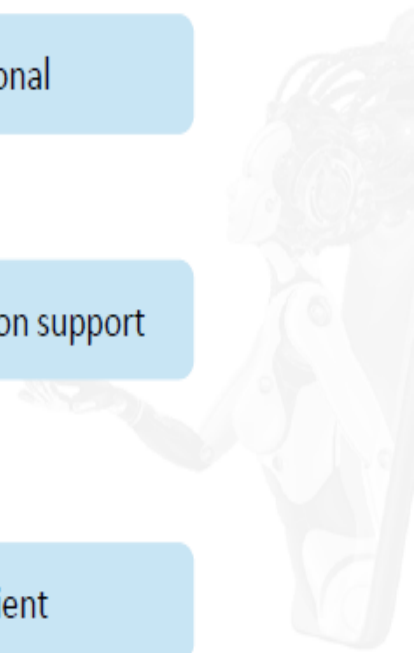
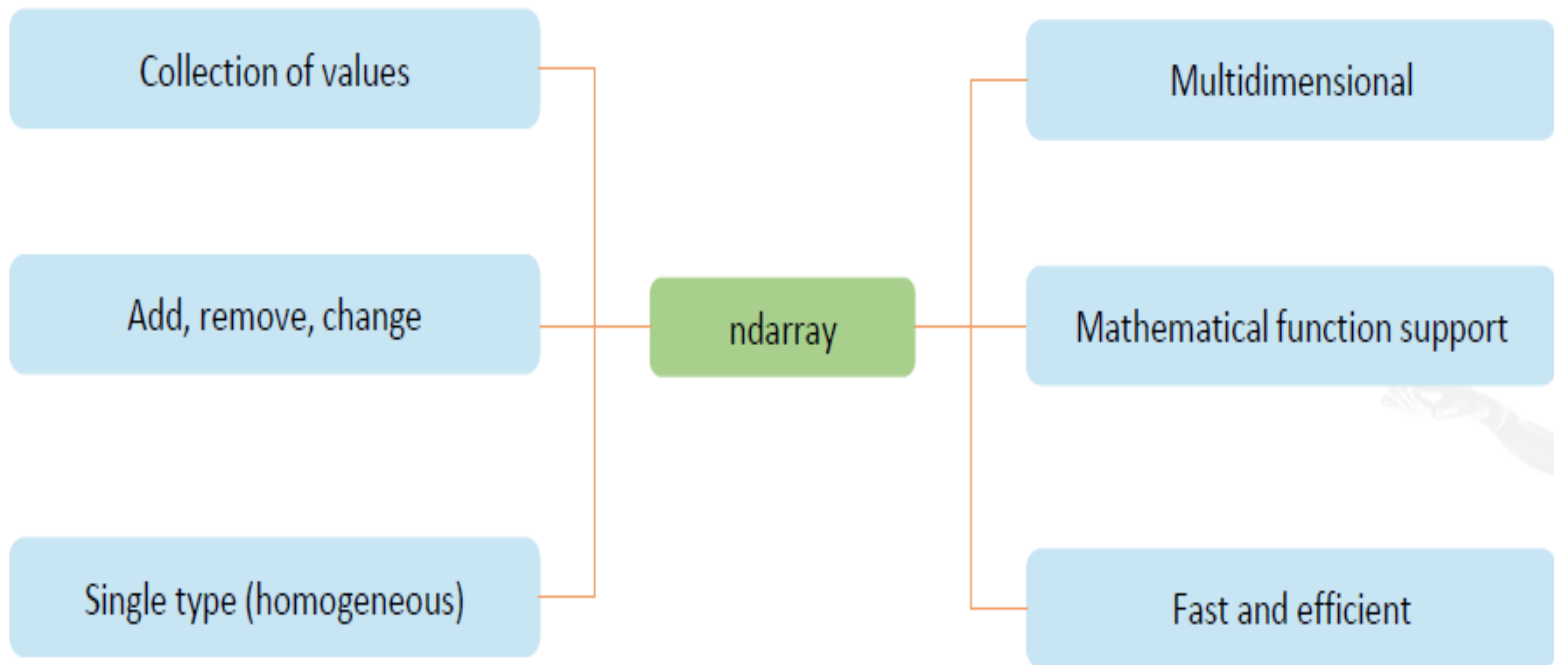
NumPy is the foundational package for mathematical computing in Python.

It has the following properties:



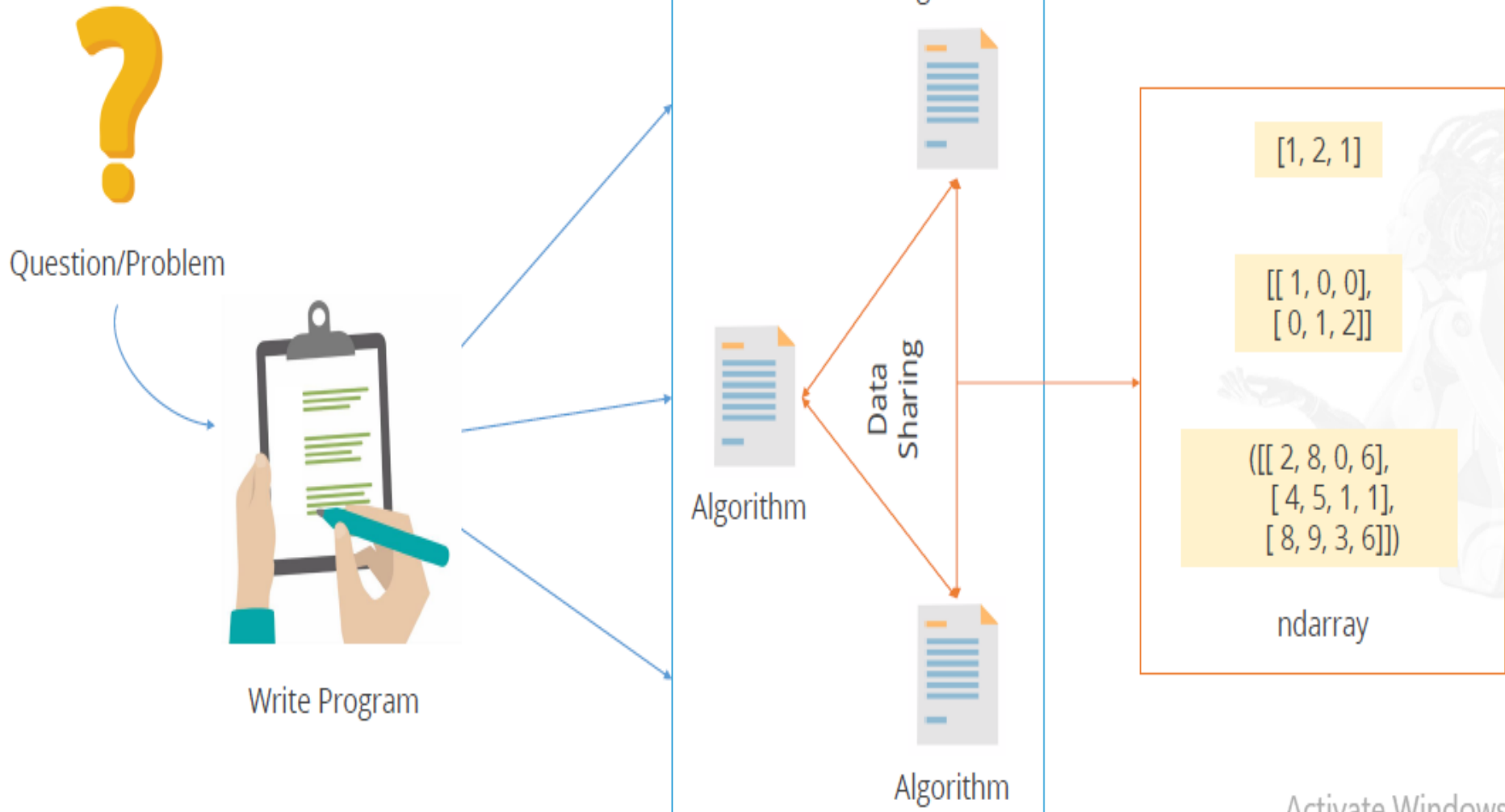
Properties of ndarray

An array in NumPy has the following properties:



Purpose of ndarray

The ndarray in Python is used as the primary container to exchange data between algorithms.



Types of Arrays

Arrays can be one-dimensional, two-dimensional, three-dimensional, or multi-dimensional.

One-Dimensional Array

Printed as rows

`array([5, 7, 9])` ← 1 axis rank 1

Length = 3

5	7	9
---	---	---

0 1 2
x axis

Two-Dimensional Array

Printed as matrices (2x3)

`array([[0, 1, 2],
[5, 6, 7]])` ← 2 axes rank 2

Length = 3

y axis ↑	0	1	2
	(0,0)	(0,1)	(0,2)
	5	6	7
	(1,0)	(1,1)	(1,2)
	x axis →		

Three-Dimensional Array

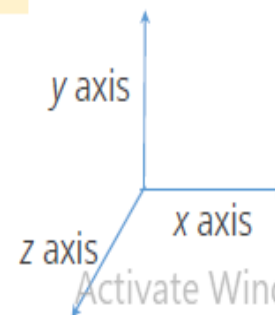
Printed as list of matrices (3x3x3)

`array([[[0, 1, 2],
[3, 4, 5],
[6, 7, 8]],`

`[[9, 10, 11],
[12, 13, 14],
[15, 16, 17]],` ← 3 axes rank 3

`[[18, 19, 20],
[21, 22, 23],
[24, 25, 26]]])`

Length = 3



Activity: Sequence it Right!

The code here is buggy. You must correct its sequence to debug it.

1

```
distance=[10,15,17,26]  
time=[.30,.47,.55,1.20]
```

2

```
np_distance = np.array(distance)  
np_time=np.array(time)
```

3

```
import numpy as np
```

4

```
speed=np_distance/np_time  
  
speed  
array([ 33.33333333,  31.91489362,  30.90909091,  21.66666667])
```

Activity: Sequence it Right!

1

```
distance=[10,15,17,26]  
time=[.30,.47,.55,1.20]
```

2

```
import numpy as np
```

3

```
np_distance = np.array(distance)  
np_time=np.array(time)
```

4

```
speed=np_distance/np_time  
  
speed  
array([ 33.33333333,  31.91489362,  30.90909091,  21.66666667])
```

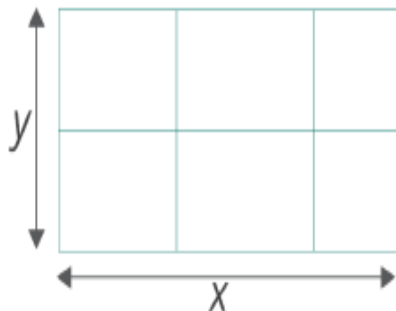
Classes and Attributes of ndarray: .ndim

Numpy's array class is **ndarray**, also referred to as **numpy.ndarray**. The attributes of ndarray are:

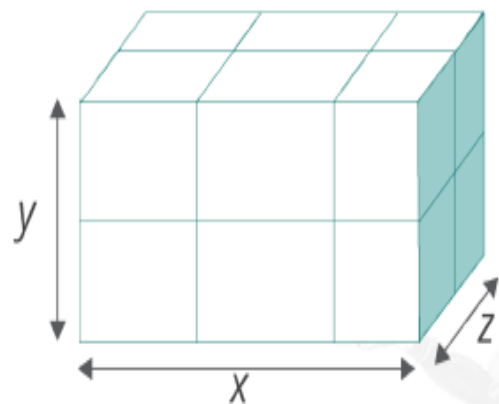
ndarray.ndim

This refers to the number of axes (dimensions) of the array. It is also called the rank of the array.

ndarray.shape



Two axes or 2D array



Three axes or 3D array

ndarray.dtype

Concept

Example

Classes and Attributes of ndarray: .ndim

ndarray.ndim

The array `np_city` is one-dimensional, while the array `np_city_with_state` is two-dimensional.

ndarray.shape

```
In [108]: np_city = np.array(['NYC', 'LA', 'Miami', 'Houston'])
```

```
In [109]: np_city.ndim
```

```
Out[109]: 1
```

ndarray.size

```
In [110]: np_city_with_state = np.array([[ 'NYC', 'LA', 'Miami', 'Houston'], [ 'NY', 'CA', 'FL', 'TX']])
```

```
In [111]: np_city_with_state.ndim
```

```
Out[111]: 2
```

ndarray.dtype

Concept

Example

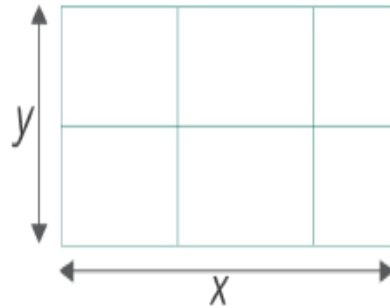
Classes and Attributes of ndarray: .shape

Numpy's array class is **ndarray**, also referred to as **numpy.ndarray**. The attributes of ndarray are:

ndarray.ndim

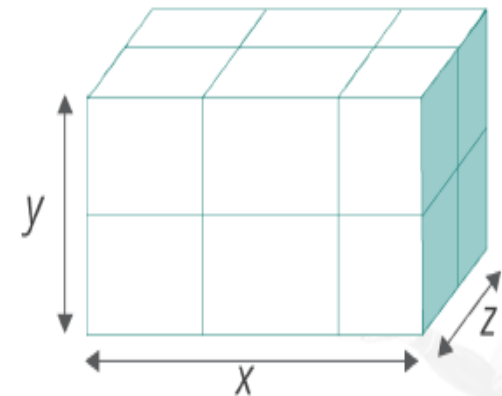
This consists of a tuple of integers showing the size of the array in each dimension. The length of the **shape tuple** is the rank or ndim.

ndarray.shape



2 rows, 3 columns

Shape: (2, 3)



2 rows, 3 columns, 2 ranks

Shape: (2, 3, 2)

ndarray.dtype

Concept

Example

Classes and Attributes of ndarray: .shape

Numpy's array class is **ndarray**, also referred to as **numpy.ndarray**. The attributes of ndarray are:

ndarray.ndim

The shape tuple of both the arrays indicate their size along each dimension.

ndarray.shape

```
In [108]: np_city = np.array(['NYC', 'LA', 'Miami', 'Houston'])
```

```
In [110]: np_city_with_state = np.array(['NYC', 'LA', 'Miami', 'Houston'], ['NY', 'CA', 'FL', 'TX'])
```

```
In [112]: np_city.shape
```

```
Out[112]: (4L,)
```

ndarray.size

```
In [113]: np_city_with_state.shape
```

```
Out[113]: (2L, 4L)
```

ndarray.dtype

Concept

Example

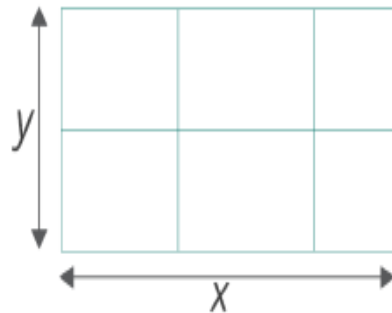
Classes and Attributes of ndarray: .size

Numpy's array class is **ndarray**, also referred to as **numpy.ndarray**. The attributes of ndarray are:

ndarray.ndim

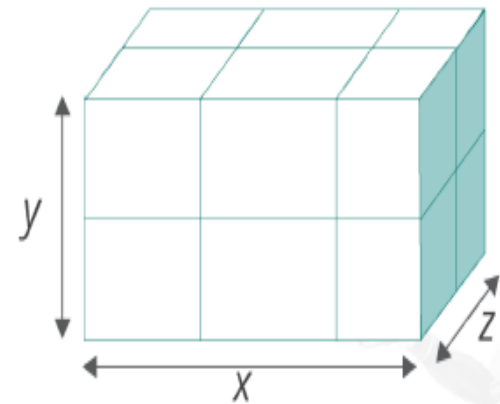
It gives the total number of elements in the array. It is equal to the product of the elements of the shape tuple.

ndarray.shape



Array contains 6 elements

Array a = (2, 3)
Size = 6



Array contains 12 elements

Array b = (2, 3, 2)
Size = 12

ndarray.size

ndarray.dtype

Concept

Example

Activate Windows

Go to Settings to activate W

Classes and Attributes of ndarray: .size

Numpy's array class is **ndarray**, also referred to as **numpy.ndarray**. The attributes of ndarray are:

ndarray.ndim

Look at the examples to see how the shape tuples of the arrays are used to calculate their size.

```
In [112]: np_city.shape
```

```
Out[112]: (4L,)
```

```
In [113]: np_city_with_state.shape
```

```
Out[113]: (2L, 4L)
```

ndarray.shape

```
In [114]: np_city.size
```

```
Out[114]: 4
```

ndarray.size

```
In [115]: np_city_with_state.size
```

```
Out[115]: 8
```

ndarray.dtype

Concept

Example

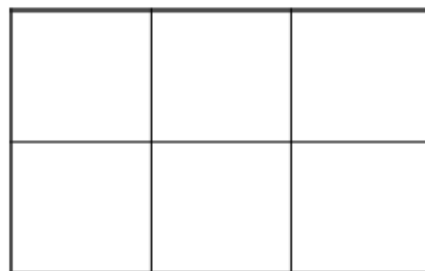
Classes and Attributes of ndarray: .dtype

Numpy's array class is **ndarray**, also referred to as **numpy.ndarray**. The attributes of ndarray are:

ndarray.ndim

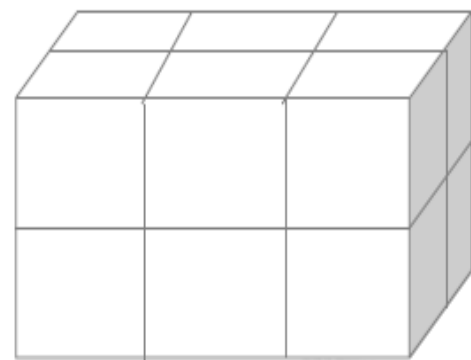
It's an object that describes the type of the elements in the array. It can be created or specified using Python.

ndarray.shape



Array contains integers

```
Array a = [3, 7, 4]
          [2, 1, 0]
```



Array contains floats

```
Array b = [1.3, 5.2, 6.7]
          [0.2, 8.1, 9.4]

          [2.6, 4.2, 3.9]
          [7.8, 3.4, 0.8]
```

ndarray.dtype

Concept

Example

Classes and Attributes of ndarray: .dtype

Numpy's array class is **ndarray**, also referred to as **numpy.ndarray**. The attributes of ndarray are:

ndarray.ndim

Both the arrays are of **string** data type (dtype) and the longest string is of length 7, which is **Houston**.

```
In [116]: np_city
```

```
Out[116]: array(['NYC', 'LA', 'Miami', 'Houston'],  
              dtype='<S7')</pre>
```

ndarray.shape

```
In [117]: np_city_with_state
```

```
Out[117]: array([[ 'NYC', 'LA', 'Miami', 'Houston'],  
                [ 'NY', 'CA', 'FL', 'TX']],  
              dtype='<S7')</pre>
```

ndarray.size

```
In [118]: np_city_with_state.dtype
```

```
Out[118]: dtype('<S7')</pre>
```

ndarray.dtype

Concept

Example



THANK YOU!