NumPy (Numerical Python)

- NumPy is an **open-source Python library.**
- NumPy is **mostly used for numerical computing in Python.**
- NumPy library **contains multidimensional array data structures.**
- It contains high-level mathematical functions that operate on these arrays and matrices.

Features of NumPy

1. **Array-oriented computing** – (ndarray) an efficient multidimensional array object, with methods to efficiently operate on it.
2. **Complex computations** – operations on entire arrays without the need for Python for loops. Example: Common algorithms - sorting, unique, and set operations.
3. **Mathematical functions** - Linear algebra, Random number generation, and Fourier transform capabilities.
4. **Statistical Computation** - Efficient descriptive statistics, aggregating or summarizing data.
5. **Data Handling** - Group-wise data categorization and transformation.
6. **Data Manipulation** - Data alignment for merging and joining together heterogeneous datasets.
7. **Data Preprocessing**- allows data cleaning, filtering and removal of duplicate data.
8. **Minimal Code** - Expressing conditional logic as array expressions instead of loops with if-elif-else branches
9. **Faster** - NumPy-based algorithms are generally 10 to 100 times faster (or more) than their pure Python counterparts and use significantly less memory.

Why NumPy?

- In Python we have l**ists that serve the purpose of arrays, but they are slow to process.**
- **NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.**
- The array object in NumPy is called **ndarray, it provides a lot of supporting functions that make working with ndarray very easy.**
- Arrays are very frequently used in data science, where speed and resources are very important.

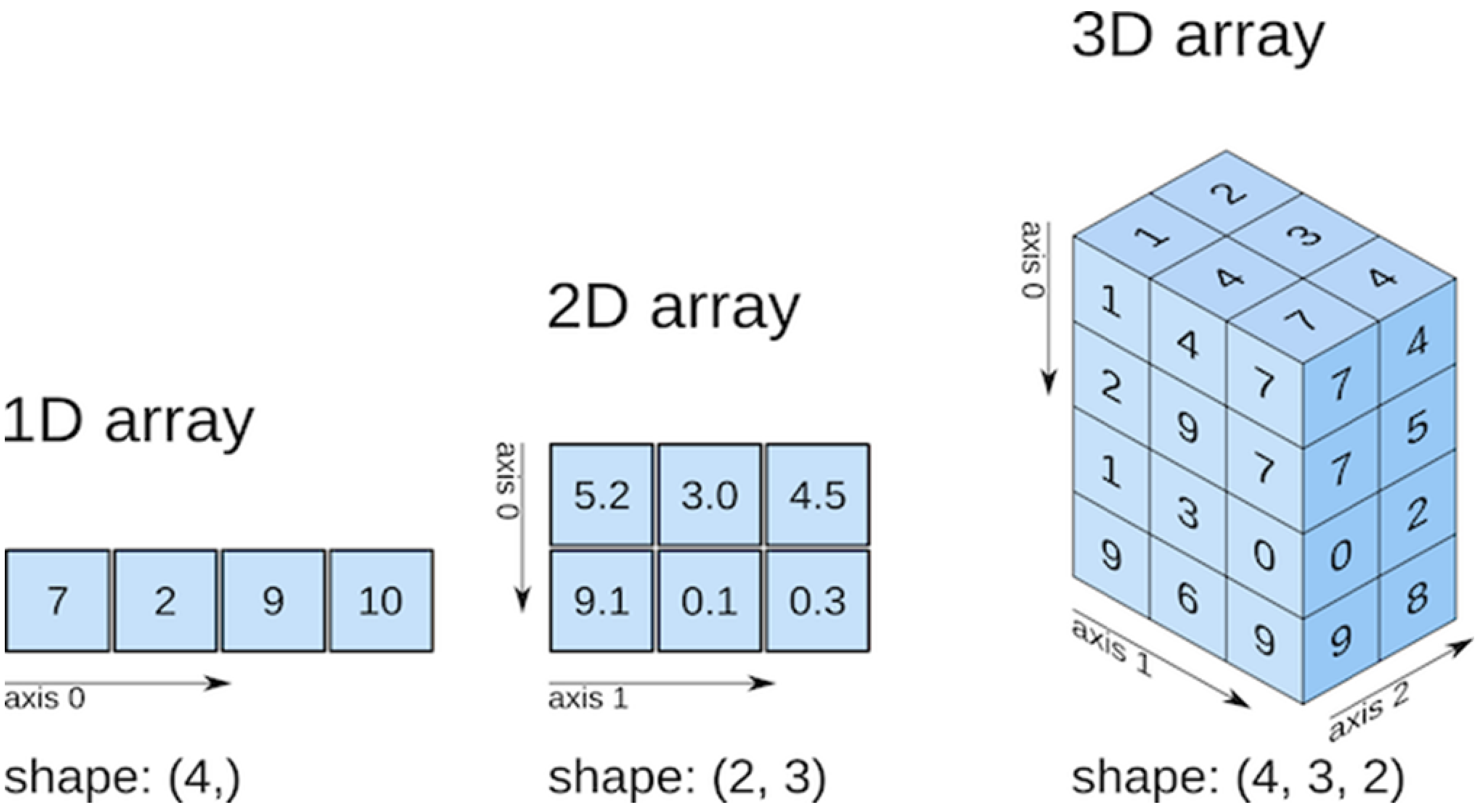
Note:

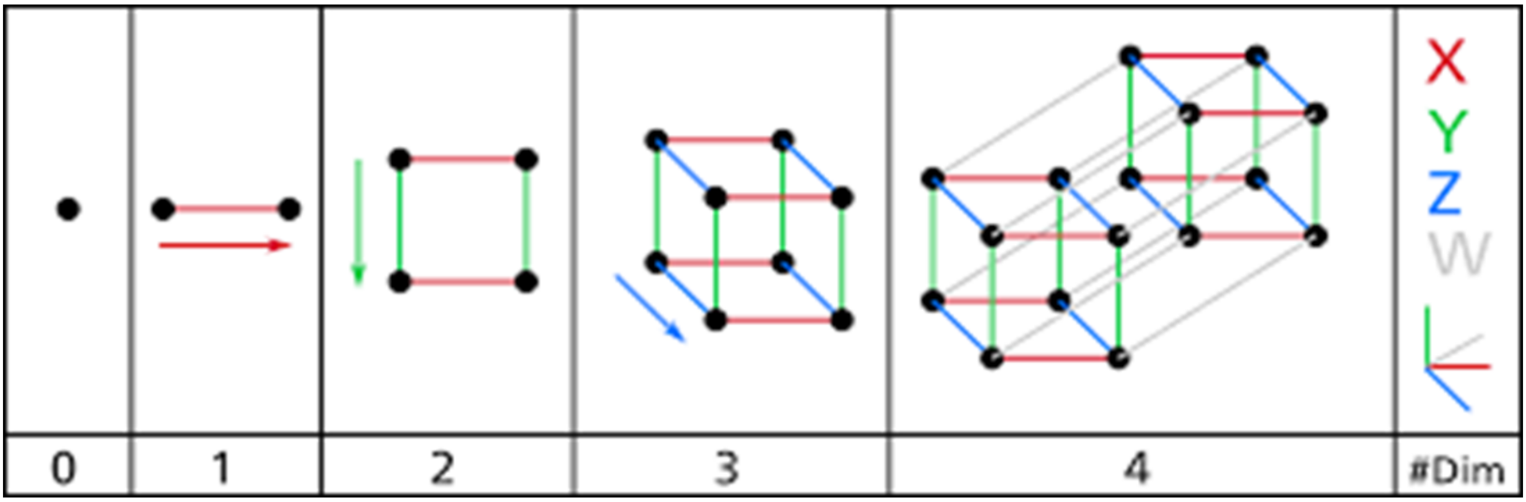
- The multi-dimensional NumPy arrays are called **Tensors.**
- Tensors are the basic data structure in *_machine learning and deep learning models._ *
- In neural networks, data (text, images, videos) are represented by using tensors.
- The term **“tensor”** is a technical term in the context of deep learning. The deep learning library **“TensorFlow”** was named keeping tensors in mind.

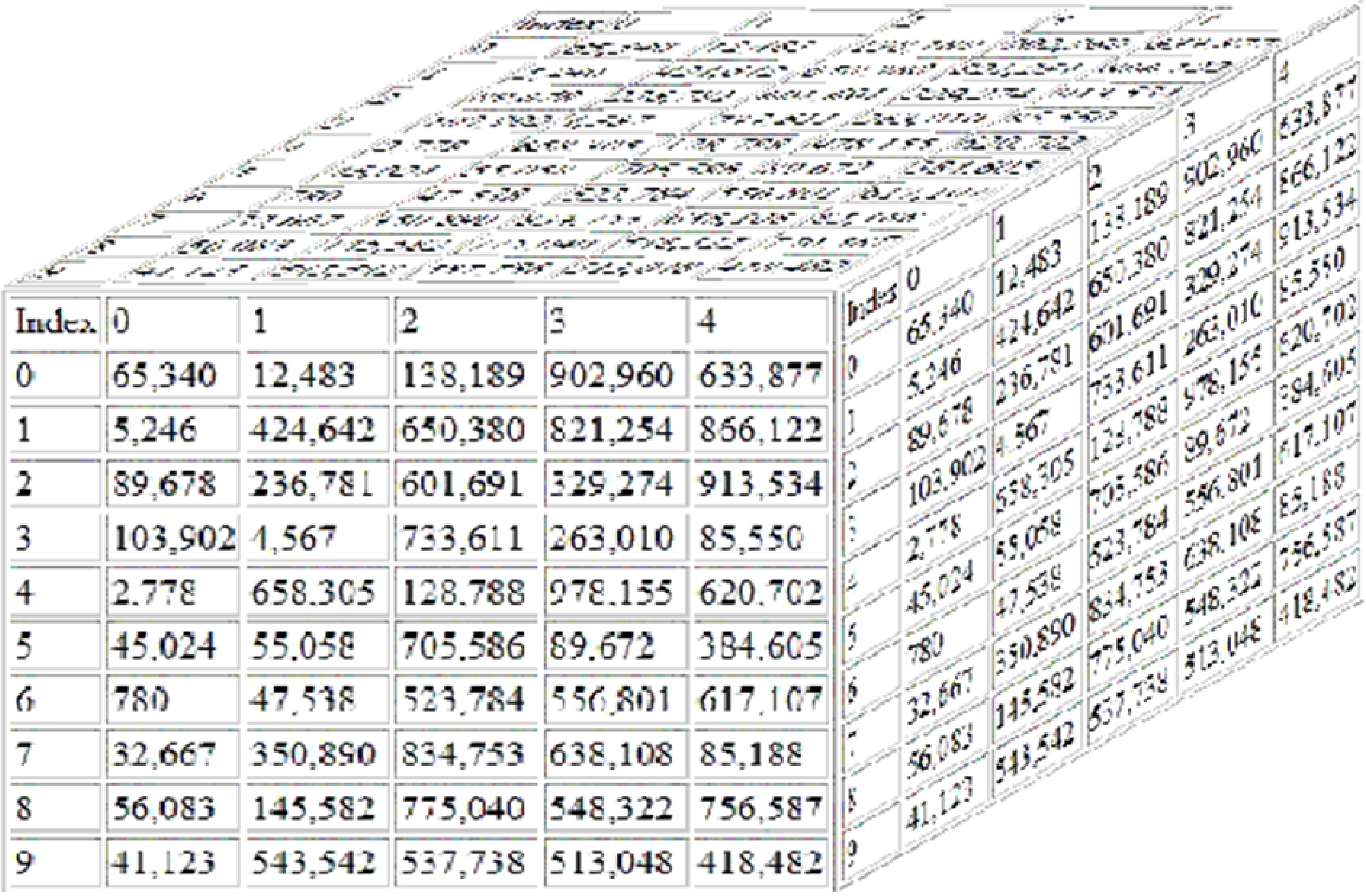
Types of NumPy Array

1. One Dimensional Array
2. Multi-Dimensional Array (nD array)

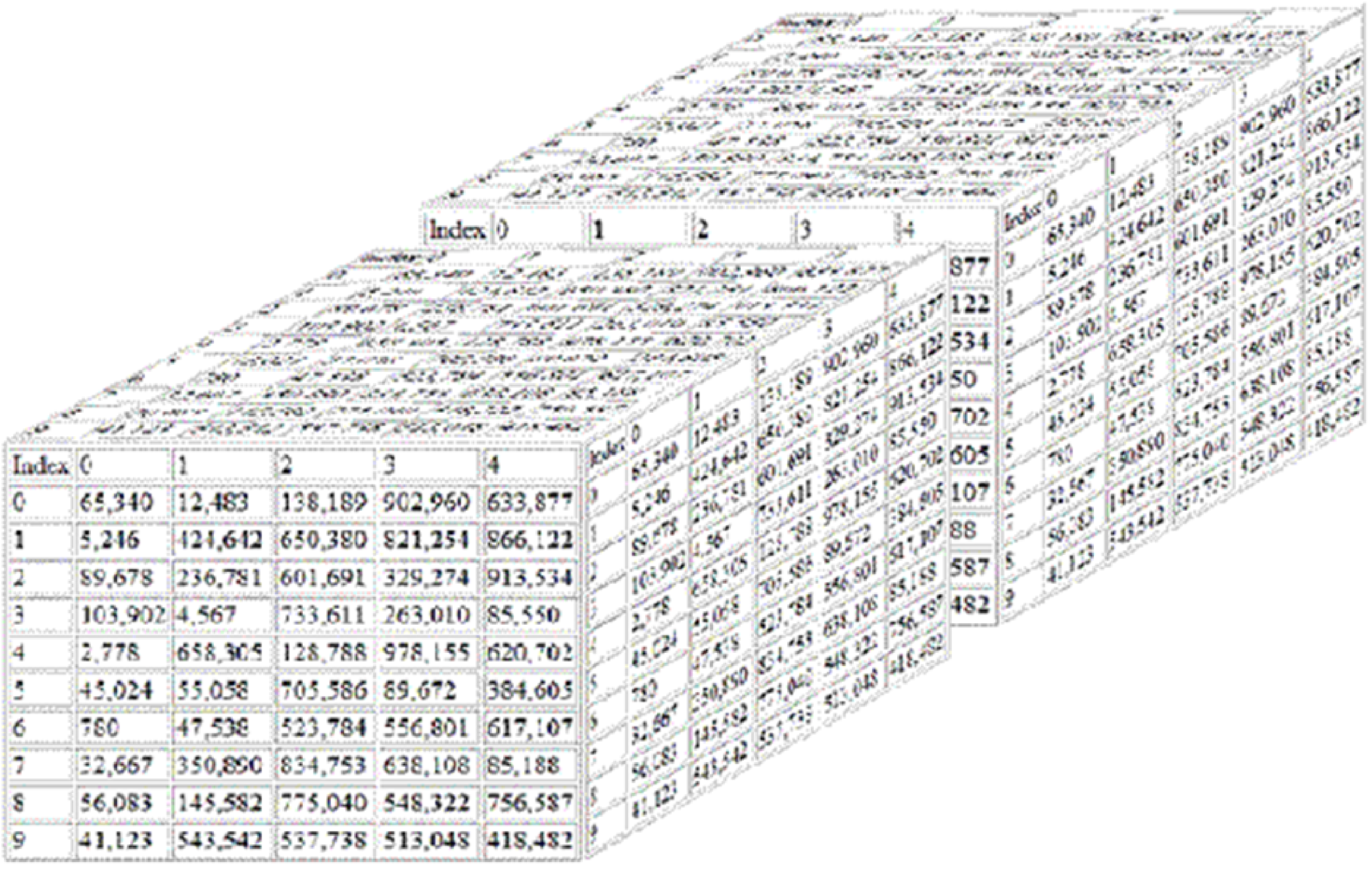
Diagrammatic Representations



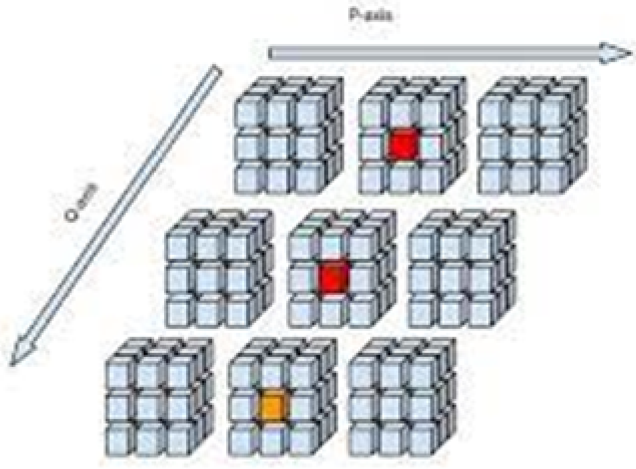


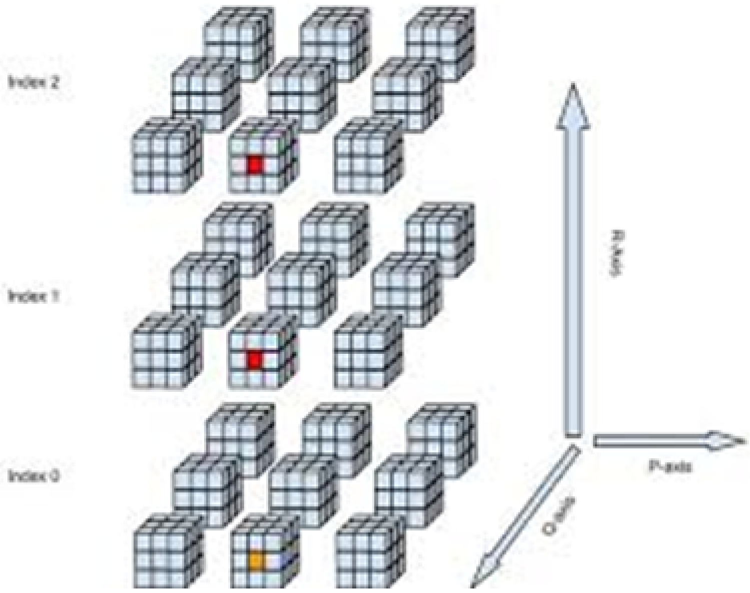


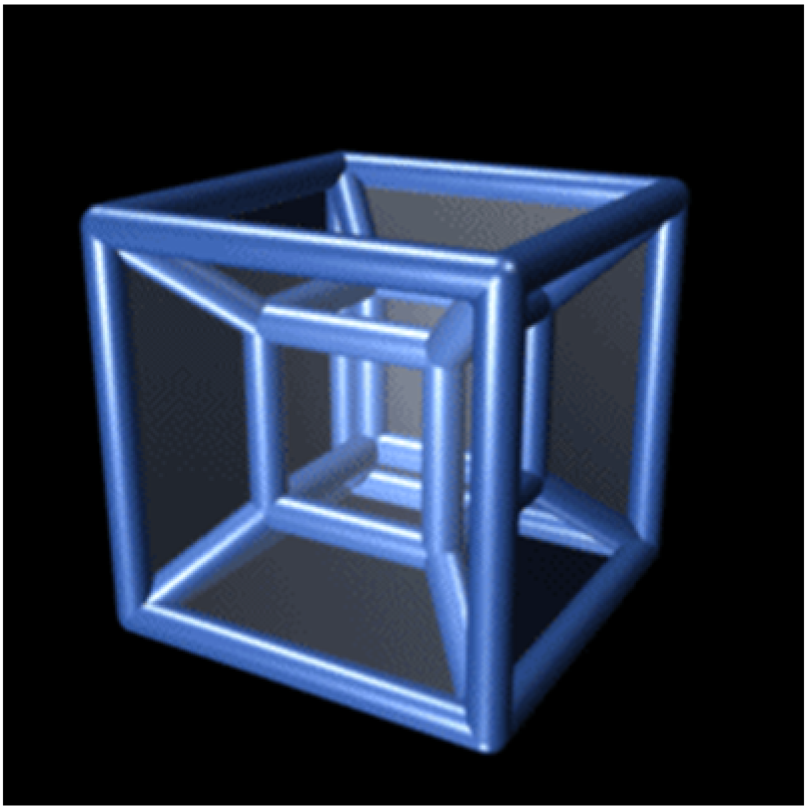
Index	0	1	2	3	4
0	65,340	12,483	138,189	902,960	633,877
1	5,246	424,642	650,380	821,254	866,122
2	89,678	236,781	601,691	329,274	913,534
3	103,902	4,567	733,611	263,010	85,550
4	2,778	658,305	128,788	978,155	620,702
5	45,024	55,058	705,586	89,672	384,605
6	780	47,538	523,784	556,801	617,107
7	32,667	350,890	834,753	638,108	85,188
8	56,083	145,582	775,040	548,322	756,587
9	41,123	543,542	537,738	513,048	418,482

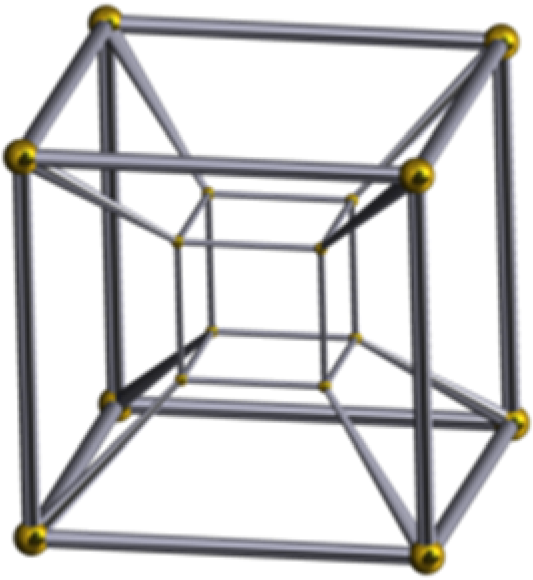


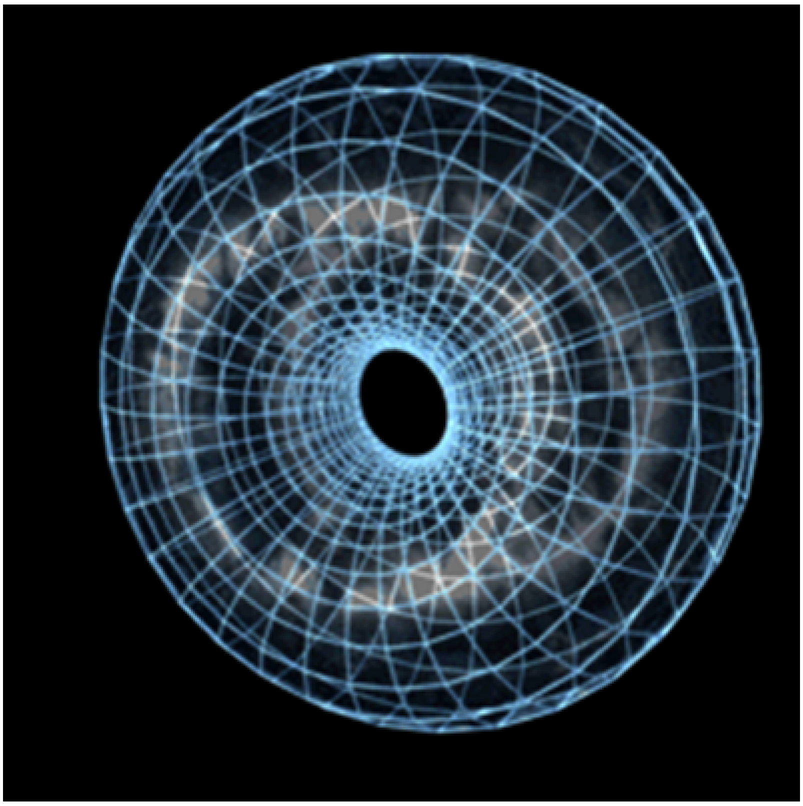
Index	0	1	2	3	4
0	65,340	12,483	138,189	902,960	633,877
1	5,246	424,642	650,380	821,254	866,122
2	89,678	236,781	601,691	329,274	913,534
3	103,902	4,567	733,611	263,010	85,550
4	2,778	658,305	128,788	978,155	620,702
5	45,024	55,058	705,586	89,672	384,605
6	780	47,538	523,784	556,801	617,107
7	32,667	350,890	834,753	638,108	85,188
8	56,083	145,582	775,040	548,322	756,587
9	41,123	543,542	537,738	513,048	418,482


P-axis
Q-axis


Index 2
Index 1
Index 0
R-axis
P-axis
Q-axis







How to use NumPy?

To access NumPy and its functions it is imported in the Python code as:

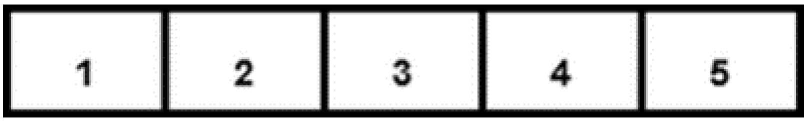
> **import numpy as np**

- The numpy is renamed to np for better readability of code using NumPy.
- This is a widely adopted convention that is followed so that anyone working with your code can easily understand it.

Basic example to learn creation of a Numpy array using array() is given below:

A) Creation of One-Dimensional Array:

A one-dimensional array is a type of linear array.

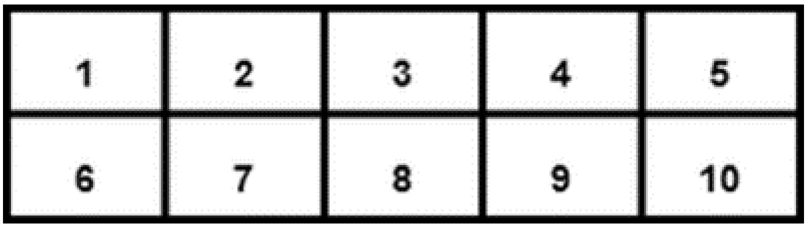


```
#Creation of One-Dimensional Array:

# importing numpy module
import numpy as np

# creating list
L = [1, 2, 3, 4]
print("List in python : ", L)
print(type(L))



# creating numpy array
a = np.array(L)
print("Numpy Array in python :", a)
print(type(a))
```

```
List in python :  [1, 2, 3, 4]
<class 'list'>
Numpy Array in python : [1 2 3 4]
<class 'numpy.ndarray'>
```

B) Creation of Multi-Dimensional Array:

Data in multidimensional arrays are stored in tabular form.



1	2	3	4	5
6	7	8	9	10

```
# importing numpy module
import numpy as np

# creating list
L1 = [1, 2, 3, 4]
L2 = [5, 6, 7, 8]
L3 = [9, 10, 11, 12]

# creating numpy array
a = np.array([L1, L2, L3])
print("Numpy multi dimensional array in python is: \n", a)
```

```
Numpy multi dimensional array in python is:
 [[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
```

✓ 0s completed at 13:14 ● ✕