Installation:

* Pip install numpy
* Pip install jupytor
* To run jupyton on cmd:

>>jupytor notebook

**What is numpy?**

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python.  
Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data.

**Why use numpy?**

* Provides efficient storage
* Provide better way of handling data for processing
* Fast
* Easy to learn
* Use less memory to store data

Pointers are not in numpy

**How to create numpy array from python list?**

**Creating a Numpy Array**

# Python program for Creation of Arrays

import numpy as np

# Creating a rank 1 Array

arr = np.array([1, 2, 3])

print("Array with Rank 1: \n",arr)

 # Creating a rank 2 Array

arr = np.array([[1, 2, 3],

                [4, 5, 6]])

print("Array with Rank 2: \n", arr)

# Creating an array from tuple

arr = np.array((1, 3, 2))

print("\nArray created using "

      "passed tuple:\n", arr)

**Output:**

Array with Rank 1:

[1 2 3]

Array with Rank 2:

[[1 2 3]

[4 5 6]]

Array created using passed tuple:

[1 3 2]

import numpy as np

myarr = np.array([4,5,2,8])

myarr

Output: array([4, 5, 2, 8])

## What is an Int64 in Python?

## You will often see the data type Int64 in Python which stands for 64 bit integer. The 64 refers to the memory allocated to store data in each cell which effectively relates to how many digits it can store in each “cell”.

* To find the shape and dtype of array:

myarr.shape, return row and column of array

myarr.dtype, return the type of array

# Array creation

There are 6 general mechanisms for creating arrays:

1. Conversion from other Python structures (i.e. lists and tuples)
2. Intrinsic NumPy array creation functions (e.g. arange, ones, zeros, etc.)
3. Replicating, joining, or mutating existing arrays
4. Reading arrays from disk, either from standard or custom formats
5. Creating arrays from raw bytes through the use of strings or buffers
6. Use of special library functions (e.g., random)
7. **Conversion from other Python structures (i.e. lists and tuples)**

NumPy arrays can be defined using Python sequences such as lists and tuples. Lists and tuples are defined using [...] and (...), respectively. Lists and tuples can define ndarray creation:

* a list of numbers will create a 1D array,
* a list of lists will create a 2D array,
* further nested lists will create higher-dimensional arrays. In general, any array object is called an **ndarray** in NumPy.

>>> a1D **=** np**.**array**([1,** **2,** **3,** **4])**

>>> a2D **=** np**.**array**([[1,** **2],** **[3,** **4]])**

>>> a3D **=** np**.**array**([[[1,** **2],** **[3,** **4]],** **[[5,** **6],** **[7,** **8]]])**

## Intrinsic NumPy array creation functions

These functions can be split into roughly three categories, based on the dimension of the array they create:

1. 1D arrays
2. 2D arrays
3. ndarrays

**Numpy Axis:**

**Numpy array attributes and methods:**

<https://docs.scipy.org/doc/numpy-1.6.0/reference/generated/numpy.ndarray.html>

**argmax():** givs max value localtion according to its index value