Why NumPy?

NumPy is an open-source numerical Python library used for working with arrays. It aims to provide an **array object that is upto 50x faster than traditional python list takes significantly less amount of memory as compared to python lists**.

How to Install Numpy

```
pip install numpy
or
conda install numpy
```

Importing Library

```
import numpy as np
```

Attributes of ndarray

ndarray.shape	Tuple of array shape
ndarray.ndim	Number of array dimensions as interger
ndarray.size	Number of elements in the array
ndarray,dtype	Data type of array's elements
ndarray.base	To check if object has its own memory

SLICING

arr[0]	Returns the element at index 0
arr[1,2]	Returns array element on index [1][2]
arr[0:3]	Returns the elements at indices on outer dimension
arr[0:3,2]	Returns the elements on rows 0,1,2 at column 2
arr<n	Returns an array with boolean values
~arr	Returns an array with boolean values

Statistics

np.mean(arr,axis=*0/1*)	Compute the arithmetic mean along the specified axis.
arr.sum()	Sum of array elements over a given axis
arr.min()	Return the minimum along a given axis
arr.max()	Return the maximum along a given axis
np.var(arr)	Compute the variance along the specified axis
np.std(arr)	Compute the standard deviation along the specified axis.
arr.corrcoef()	Return Pearson product-moment correlation coefficients

Creating Arrays

np.array(object)	Creates an array
np.array([1,2,3])	1D array
np.array([(1,2,3),(4,5,6)])	2D array
np.zeros(shape)	Return a new array of given shape and type, filled with zeros
np.ones(shape)	Return a new array of given shape and type, filled with ones
np.eye(no. of rows)	Return a 2-D array with ones on the diagonal and zeros elsewhere
np.arange(start,stop,step)	Return evenly spaced values within a given interval.
np.random.rand(shape)	Return array of random floats between 0–1 of fiven shape
np.random.randint(low,high)	Return random integers from low (inclusive) to high (exclusive)
np.linspace(start, stop, n)	Returns n evenly spaced numbers over a specified interval

commonly used methods

np.sort(arr)	Returns a sorted copy of the array
np.argsort(arr)	Returns the indices that would sort an array
np.resize(a, new_shape)	Return a new array with the specified shape
np.dot(arr1, arr2)	Dot product of two arrays
arr.copy()	Returns a copy of the array
arr.view()	New view of array with the same data
arr.flatten()	Return a copy of the array collapsed into 1D
arr.reshape(new-_shape)	Returns an array containing the same data with a new shape

Math operators

np.add(arr_1, arr_2)	Add arguments element-wise
np.subtract(arr_1, arr_2)	Subtract arguments, element-wise
np.multiply(arr_1, arr_2)	Multiply arguments, element-wise
np.divide(arr_1, arr_2)	Divide arguments, element-wise

Math operators (cont)	
np.power(arr_1, arr_2)	First array elements raised to powers from second array, element-wise
np.sqrt(arr)	Return the non-negative square-root of an array, element-wise
np.log(arr)	Natural logarithm, element-wise
np.ceil(arr)	Rounds up to the nearest int , element-wise
np.floor(arr)	Rounds down to the nearest int ,element-wise
np.abs(arr)	Absolute value of each element in the array
np.round(arr)	Rounds to the nearest int

Useful links
NumPy Official documenataion
w3schools NumPy Tutorial
NumPy Illustrated: The Visual Guide to NumPy
NumPy: creating and manipulating numerical data

By **Lavanya-M22**
cheatography.com/lavanya-m22/

Published 11th September, 2021.
Last updated 12th September, 2021.