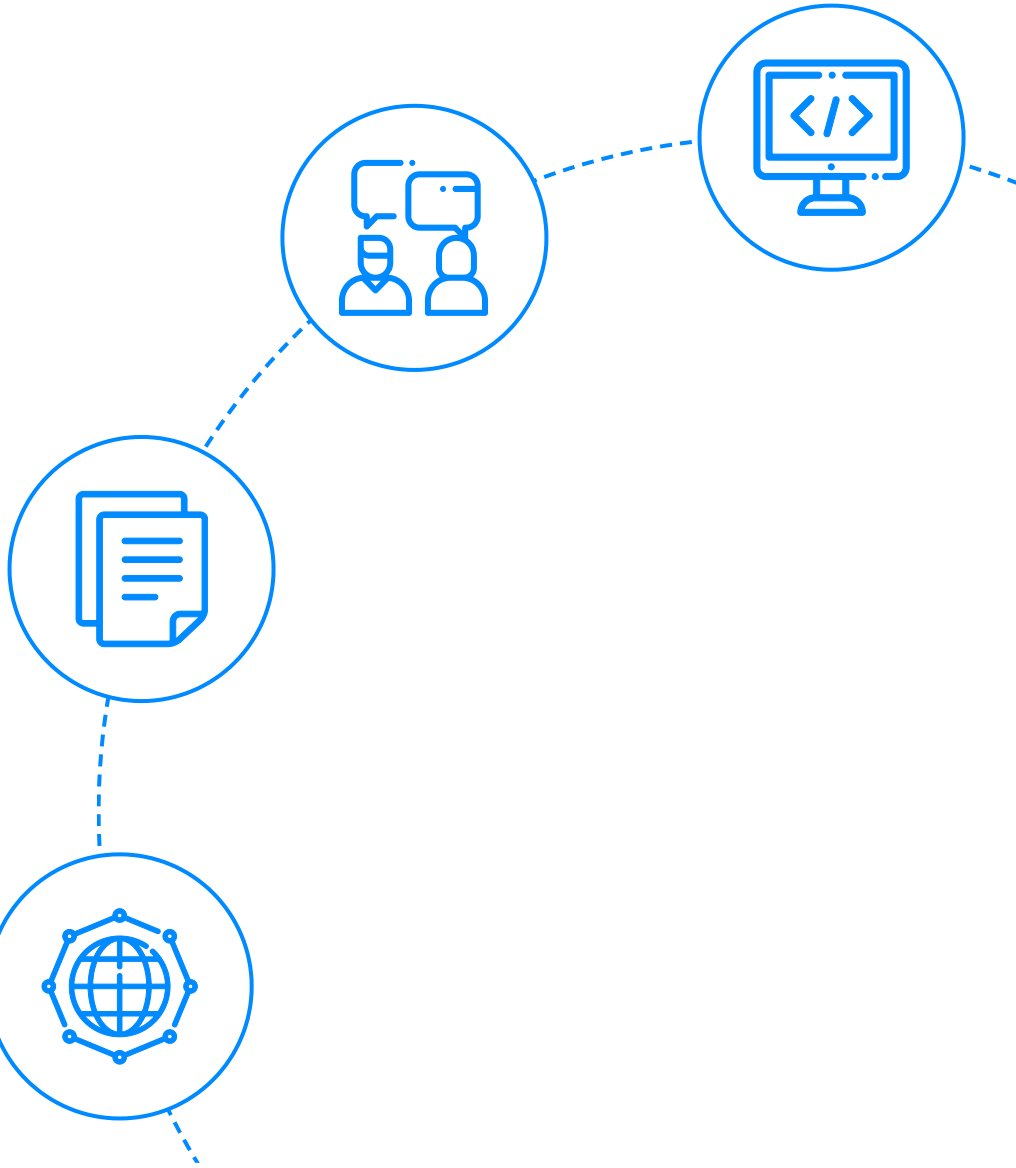
InterviewBit

NumPY Cheat Sheet



To view the live version of the page, click here.

Contents

Numpy Tutorial: Basic to Advanced

Let's get Started

Introduction

Jim Hugunin originally developed Numeric, the predecessor to NumPy, with assistance from a number of other programmers. Travis Oliphant developed it in 2005 by heavily altering Numeric to incorporate features of the rival Numarray. The multidimensional array objects and the collection of operations for handling those arrays are part of an open-source library. Its sturdy n-dimensional array speeds up data processing. It offers the ability to easily interact with other Python packages and other programming languages like C, C++, etc. The foundational library for Python's scientific computing is this one. It offers a multidimensional array object with outstanding speed as well as capabilities for interacting with these arrays. The library is what enables Python's quick data manipulation.

NumPy targets the non-optimizing CPython bytecode interpreter, which is the Python reference implementation. Due to the lack of compiler optimization, mathematical algorithms created for this version of Python frequently execute considerably slower than their compiled counterparts. Multidimensional arrays, efficient array-based functions, and operators are some of the ways that NumPy addresses the slowness issue. Using them necessitates rewriting some code, primarily inner loops, in NumPy.

What is NumPy?

The core Python library for scientific computing is called NumPy. It includes multidimensional array objects, various derived objects (masked arrays, matrices, etc.), and a variety of quick manipulations of arrays such as math, logic, shape manipulation, sorting, selection, I / O, and discrete. A Python library that provides a variety of routines. Fourier transform, basic linear algebra, basic statistical operations, random simulation, etc.

The ndarray object is the core of the NumPy package. This contains n-dimensional arrays of uniform data types, with many operations carried out in compiled code for speed. **View Important Interview Questions on NumPY**.

Why Use NumPy?

Some of the key features that contribute to the popularity of NumPy are:

- It is a powerful N-dimensional array object
- It is a sophisticated broadcasting functions
- It is a tool for integrating C/C++ and Fortran code
- It is useful for linear algebra, Fourier transform, and random number capabilities

Limitations Of NumPy

An array cannot be added to or subtracted from in the same manner as a list in Python. When extending an array, the np.pad(...) function actually creates new arrays with the required shape and padding values, copies the existing array into the new one, and then returns the new array.

The np.concatenate([a1,a2]) action in NumPy returns a new array that is filled with the sequential items from the two given arrays rather than actually linking the two arrays together. Only when the array's element count stays constant can an array's dimensionality be changed using the np.reshape(...) method. These conditions result from the need that arrays in NumPy to be viewed on contiguous memory buffers. Blaze, a substitute package, makes an effort to get around this restriction.

Numerous contemporary large-scale scientific computing applications have needs that go beyond what NumPy arrays can handle. For instance, NumPy arrays are frequently loaded into a computer's RAM, which may not have enough space to accommodate huge dataset processing. Furthermore, a single CPU is used to do NumPy computations. However, by running them on groups of CPUs or specialized hardware, such as GPUs and TPUs, which are used in many deep learning applications, many linear algebra operations may be made faster.

Numpy Tutorial: Basic to Advanced

1. Importing/Exporting Commands

> Note: Importing numpy as n for all codes.

Command	Explanation
`n.save('my_array' , a)`	This command saves the array on disk.
`n.load( 'my_array.npy')`	This command loads an array on the CSV file disk.
`n.loadtxt("myfile.txt")`	This command loads text files on the disk.
`n.savetxt('New_file.txt',arr,delimiter=' ')`	This command writes your array to a Text file on disk.
`n.savetxt('New_file.txt',arr,delimiter=' ')`	These command commands write your array to CSV file on the disk

2. Creating Arrays Commands

Command	Explanation
`a = n.array()`	This command is used to create an empty array.
`a = n.array([10,20,30])`	This command is used to create an array with values.
`a = n.array([10,20,30], [11,22,33])`	This command is used to create a 2D array.
`n.zeros(p)`	This command 1 D arrays with all zeros in it with p number of elements. The goal of this article is to introduce you to NumPy, familiarize you with them, and provide you with a cheat sheet. It takes some effort to read through the extensive online documentation for NumPy to find what you need. We all want to use new tools as quickly and efficiently as possible. This article should have been able to help you with that.
`n.ones((a,b))`	This command creates a 2D array of Size a*b with all ones in it.
`n.ones((a,b))`	This command is used to make a 1 D array of values from 0 to b.
`y = n.arange(0,n,p)`	This command is used to make a 1 D array of values from 0 to n with p steps.

3. Inspecting Properties Commands

Command	Explanation
`array.size`	This command tells about the count of elements in the array.
`array.shape`	This command tells about rows and columns of an array.
`array.dtype`	This command tells about the element data type in the array.
`array.astype(dtype)`	This command converts elements of an Array to a given data type.
`array.tolist()`	This command converts arrays to python lists.
`n.info(n.eye)`	This command searches for documentation of np.eye.

4. Copying, Sorting and Reshaping Commands

Command	Explanation
`n.copy(array)`	This command copies the array to a new memory space.
`array.view(dtype)`	This command view of array elements with given data type2-dimensional.
`array.sort()`	This command sorts the array.
`np.sort(array, axis=0)`	This command is used to sort along the first axis of the array.
`two_d_array.flatten()`	This command flattens a 2-dimensional array to a 1-dimensional array.
`array.T`	This command transposes the Array.
`array.reshape(m,n)`	This command reshapes the given array in the m * n dimension.
`array.resize((m,n))`	This command resizes the given array in m * n size.

5. Adding and Removing Commands

Command	Explanation
`n.append(array,values)`	This command is used to append values to the end of the array.
`array.reshape(m,n)`	This command is used to insert values into an array before index n.
`n.delete(array,n,axis=0)`	This command is used to delete rows on index n of the array.
`n.delete(array,n,axis=1)`	This command is used to delete columns on index n of the array

6. Combining and Splitting Commands

Command	Explanation
`n.concatenate((array1,array2),axis=0)`	This command is used to add arr2 as rows to the end of arr1.
`n.concatenate((array1,array2),axis=1)`	This command is used to add arr2 as columns to the end of arr1.
`array1 np.split(array,n)`	This command splits array arr into n sub-arrays.
`n.hsplit(array,n)`	This command splits array arr horizontally on the nth index.

7. Indexing, slicing and subsetting

Command	Explanation
`array[m]`	This command will return the value of the element at index m.
`array[a,b]`	This command will return the value of the 2D array element on index [a][b].
`array[a]=b`	This command assigns value to an array element on index a the value b.
`array[a,b]=p`	This command assigns value to an array element on index [a][b] the value p.
`array[a:b]`	This command will return values of the elements at indices n to m (On a 2D array: returns rows a to b)
`array[0:a,b]`	This command will return values of the elements on rows 0 to a at column b
`array[:m]`	This command will return values of the elements at indices 0,m-1 (On a 2D array: returns rows 0,m-1)
`array[:,m]`	This command will return values of the elements at index m on all rows
`array<p`	This command will return values of an array with boolean values.
`(array1<p) & (array2>q)`	This command will return values of an array with boolean values.
`~array`	This command will invert values of a boolean array.

8. Scalar Math Commands

Command	Explanation
`n.add(array,a)`	This command will add a to each array element value.
`n.subtract(array,b)`	This command will subtract b from each array element value.
`n.multiply(array,c)`	This command will multiply each array element value by c.
`n.divide(array,d)`	This command will divide each array element value by d (returns np.nan for division by zero).
`n.power(array,e)`	This command will raise each array element value to the power of e.

9. Vector Math Commands

Command	Explanation
`n.add(array1,array2)`	This command will add values of arguments arr1 and arr2 element-wise.
`n.subtract(array1,array2)`	This command will subtract the values of arguments arr1 and arr2 element-wise.
`n.multiply(array1,array2)`	This command will multiply the values of arguments arr1 and arr2 element-wise.
`n.divide(array1,array2)`	This command will add values of arguments arr1 and arr2 element-wise.
`n.power(array1,array2)`	This command will raise arr1 values to exponents in arr2 element-wise.
`n.array_equal(array1,array2)`	This command will return true if the arrays have the same elements and shape.
`n.sqrt(array)`	This command will perform the square root of each element value in the array.
`n.sin(array)`	This command will perform sine to the value of each element in the array.
`n.log(array)`	This command will perform a natural log of each element value in the array

10. Statistics Commands

Command	Explanation
`n.mean(array,axis=0)`	This command will return mean along a specific axis of the array values
`array.sum()`	This command will return the sum of the array of elements
`array.min()`	This command will return the minimum value of array elements
`array.max(axis=0)`	This command will return the maximum value of the specific axis of array elements
`n.var(array)`	This command will return the variance of array elements
`n.std(array,axis=1)`	This command will return the standard deviation of the specific axis for array elements
`array.corrcoef()`	This command will return the correlation coefficient of array elements

Conclusion

This article aims to introduce you to NumPy, familiarize you with them, and provide you with a cheat sheet. It takes some effort to read through the extensive online documentation for NumPy to find what you need. We all want to use new tools as quickly and efficiently as possible. This article should have been able to help you with that.

NumPy arrays can be used for math operations, data loading and storing, and array indexing. We covered array manipulation, import/export, and statistical techniques that are crucial.

Important Resources

- NumPy Documentation
- NumPy Mathematical Functions
- Creating a Numpy DataType
- Pandas Vs NumPy: What's The Difference?
- Numpy Arrays - Problems

Links to More Interview Questions

C Interview Questions	Php Interview Questions	C Sharp Interview Questions
Web Api Interview Questions	Hibernate Interview Questions	Node Js Interview Questions
Cpp Interview Questions	Oops Interview Questions	Devops Interview Questions
Machine Learning Interview Questions	Docker Interview Questions	Mysql Interview Questions
Css Interview Questions	Laravel Interview Questions	Asp Net Interview Questions
Django Interview Questions	Dot Net Interview Questions	Kubernetes Interview Questions
Operating System Interview Questions	React Native Interview Questions	Aws Interview Questions
Git Interview Questions	Java 8 Interview Questions	Mongodb Interview Questions
Dbms Interview Questions	Spring Boot Interview Questions	Power Bi Interview Questions
Pl Sql Interview Questions	Tableau Interview Questions	Linux Interview Questions
Ansible Interview Questions	Java Interview Questions	Jenkins Interview Questions