Python Numpy

Last Updated : 15 Jul, 2024

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.

Arrays in Numpy

Array in Numpy is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers. In Numpy, number of dimensions of the array is called rank of the array.A tuple of integers giving the size of the array along each dimension is known as shape of the array. An array class in Numpy is called as **ndarray**. Elements in Numpy arrays are accessed by using square brackets and can be initialized by using nested Python Lists.

Creating a Numpy Array

Arrays in Numpy can be created by multiple ways, with various number of Ranks, defining the size of the Array. Arrays can also be created with the use of various data types such as lists, tuples, etc. The type of the resultant array is deduced from the type of the elements in the sequences.

Note: Type of array can be explicitly defined while creating the array.

Python

```python
# 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]
```

Accessing the array Index

In a numpy array, indexing or accessing the array index can be done in multiple ways. To print a range of an array, slicing is done. Slicing of an array is defining a range in a new array which is used to print a range of elements from the original array. Since, sliced array holds a range of elements of the original array, modifying content with the help of sliced array modifies the original array content.

Python

```
# Python program to demonstrate
# indexing in numpy array
import numpy as np

# Initial Array
arr = np.array([[-1, 2, 0, 4],
                [4, -0.5, 6, 0],
                [2.6, 0, 7, 8],
```

```
                [3, -7, 4, 2.0]])
print("Initial Array: ")
print(arr)

# Printing a range of Array
# with the use of slicing method
sliced_arr = arr[:2, ::2]
print ("Array with first 2 rows and"
    " alternate columns(0 and 2):\n", sliced_arr)

# Printing elements at
# specific Indices
Index_arr = arr[[1, 1, 0, 3],
                [3, 2, 1, 0]]
print ("\nElements at indices (1, 3), "
    "(1, 2), (0, 1), (3, 0):\n", Index_arr)
```

Output

```
Initial Array:
[[-1.   2.   0.   4. ]
 [ 4.  -0.5  6.   0. ]
 [ 2.6  0.   7.   8. ]
 [ 3.  -7.   4.   2. ]]
Array with first 2 rows and alternate columns(0 and 2):
 [[-1.  0.]
 [ 4.  6.]]

Elements a...
```

Basic Array Operations

In numpy, arrays allow a wide range of operations which can be performed on a particular array or a combination of Arrays. These operation include some basic Mathematical operation as well as Unary and Binary operations.

Python

```
# Python program to demonstrate
# basic operations on single array
import numpy as np
```

```
# Defining Array 1
a = np.array([[1, 2],
              [3, 4]])

# Defining Array 2
b = np.array([[4, 3],
              [2, 1]])

# Adding 1 to every element
print ("Adding 1 to every element:", a + 1)

# Subtracting 2 from each element
print ("\nSubtracting 2 from each element:", b - 2)

# sum of array elements
# Performing Unary operations
print ("\nSum of all array "
       "elements: ", a.sum())

# Adding two arrays
# Performing Binary operations
print ("\nArray sum:\n", a + b)
```

Output

```
Adding 1 to every element: [[2 3]
 [4 5]]

Subtracting 2 from each element: [[ 2  1]
 [ 0 -1]]

Sum of all array elements:  10

Array sum:
 [[5 5]
 [5 5]]
```

More on Numpy Arrays

- Basic Array Operations in Numpy

- Advanced Array Operations in Numpy
- Basic Slicing and Advanced Indexing in NumPy Python

Data Types in Numpy

Every Numpy array is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers. Every ndarray has an associated data type (dtype) object. This data type object (dtype) provides information about the layout of the array. The values of an ndarray are stored in a buffer which can be thought of as a contiguous block of memory bytes which can be interpreted by the dtype object. Numpy provides a large set of numeric datatypes that can be used to construct arrays. At the time of Array creation, Numpy tries to guess a datatype, but functions that construct arrays usually also include an optional argument to explicitly specify the datatype.

Constructing a Datatype Object

In Numpy, datatypes of Arrays need not to be defined unless a specific datatype is required. Numpy tries to guess the datatype for Arrays which are not predefined in the constructor function.

Python

```python
# Python Program to create
# a data type object
import numpy as np

# Integer datatype
# guessed by Numpy
x = np.array([1, 2])
print("Integer Datatype: ")
print(x.dtype)

# Float datatype
# guessed by Numpy
x = np.array([1.0, 2.0])
print("\nFloat Datatype: ")
print(x.dtype)

# Forced Datatype
x = np.array([1, 2], dtype = np.int64)
```

```
print("\nForcing a Datatype: ")
print(x.dtype)
```

Output

```
Integer Datatype:
int64

Float Datatype:
float64

Forcing a Datatype:
int64
```

Math Operations on DataType array

In Numpy arrays, basic mathematical operations are performed element-wise on the array. These operations are applied both as operator overloads and as functions. Many useful functions are provided in Numpy for performing computations on Arrays such as **sum**: for addition of Array elements, **T**: for Transpose of elements, etc.

Python

```
# Python Program to create
# a data type object
import numpy as np

# First Array
arr1 = np.array([[4, 7], [2, 6]],
                dtype = np.float64)

# Second Array
arr2 = np.array([[3, 6], [2, 8]],
                dtype = np.float64)

# Addition of two Arrays
Sum = np.add(arr1, arr2)
print("Addition of Two Arrays: ")
print(Sum)
```

```
17
18 # Addition of all Array elements
19 # using predefined sum method
20 Sum1 = np.sum(arr1)
21 print("\nAddition of Array elements: ")
22 print(Sum1)
23
24 # Square root of Array
25 Sqrt = np.sqrt(arr1)
26 print("\nSquare root of Array1 elements: ")
27 print(Sqrt)
28
29 # Transpose of Array
30 # using In-built function 'T'
31 Trans_arr = arr1.T
32 print("\nTranspose of Array: ")
33 print(Trans_arr)
```

Output

```
Addition of Two Arrays:
[[ 7. 13.]
 [ 4. 14.]]

Addition of Array elements:
19.0

Square root of Array1 elements:
[[2.         2.64575131]
 [1.41421356 2.44948974]]

Transpose of Array:
[[4. 2.]
 ...
```

More on Numpy Data Type

- Data type Object (dtype) in NumPy

Methods in Numpy:

all()	diag()	hypot()	ones_like()
any()	diagflat()	absolute()	full_like()
take()	diag_indices()	ceil()	sin()
put()	asmatrix()	floor()	cos()
apply_along_axis()	bmat()	degrees()	tan()
apply_over_axes()	eye()	radians()	sinh()
argmin()	roll()	npv()	cosh()
argmax()	identity()	fv()	tanh()
nanargmin()	arange()	pv()	arcsin()
nanargmax()	place()	power()	arccos()
amax()	extract()	float_power()	exp()
amin()	compress()	log()	exp2()
insert()	rot90()	log1()	fix()
delete()	tile()	log2()	logical_or()
append()	reshape()	log10()	logical_and()
around()	ravel()	dot()	logical_not()
flip()	isinf()	vdot()	logical_xor()
fliplr()	isrealobj()	trunc()	array_equal()

flipud()	isscalar()	divide()	array_equiv()
triu()	isneginf()	floor_divide()	arctan2()
tril()	isposinf()	true_divide()	equal()
tri()	iscomplex()	random.rand()	not_equal()
empty()	isnan()	random.randn()	less()
empty_like()	iscomplexobj()	ndarray.flat()	less_equal()
zeros()	isreal()	expm1()	greater()
zeros_like()	isfinite()	bincount()	greater_equal()
ones()	isfortran()	rint()	prod()
	arctan()	cbrt()	square()

Programs on Numpy

- Python | Check whether a list is empty or not
- Python | Get unique values from a list
- Python | Multiply all numbers in the list (3 different ways)
- Transpose a matrix in Single line in Python
- Multiplication of two Matrices in Single line using Numpy in Python
- Python program to print checkerboard pattern of nxn using numpy
- Graph Plotting in Python | Set 1, Set 2, Set 3

Useful Numpy Articles

- Matrix manipulation in Python
- Basic Slicing and Advanced Indexing in NumPy Python
- Differences between Flatten() and Ravel()
- rand vs normal in Numpy.random in Python

Are you passionate about data and looking to make one giant leap into your career? Our **Data Science Course** will help you change your game and, most importantly, allow students, professionals, and working adults to tide over into the data science immersion. Master state-of-the-art methodologies, powerful tools, and industry best practices, hands-on projects, and real-world applications. Become the executive head of industries related to **Data Analysis**, **Machine Learning**, and **Data Visualization** with these growing skills. Ready to Transform Your Future? ***Enroll Now to Be a Data Science Expert!***

kuma...

2

Previous Article

NumPy Introduction & Guide for Beginners

Next Article

NumPy Array in Python

Similar Reads

Python | Numpy numpy.transpose()

With the help of Numpy numpy.transpose(), We can perform the simple function of transpose within one line by using numpy.transpose() method of Numpy. It ca...

2 min read

Python | Numpy numpy.ndarray.__lt__()

With the help of numpy.ndarray.__lt__() method of Numpy, We can find that which element in an array is less than the value which is provided in the...

1 min read

Python | Numpy numpy.ndarray.__gt__()

With the help of numpy.ndarray.__gt__() method of Numpy, We can find that which element in an array is greater then the value which is provided in the...

1 min read

Python | Numpy numpy.ndarray.__le__()

With the help of numpy.ndarray.__le__() method of Numpy, We can find that which element in an array is less than or equal to the value which is provided in...

1 min read

Python | Numpy numpy.ndarray.__ge__()

With the help of numpy.ndarray.__ge__() method of Numpy, We can find that which element in an array is greater then or equal to the value which is provided...

1 min read

Python | Numpy numpy.ndarray.__ne__()

With the help of numpy.ndarray.__ne__() method of Numpy, We can find that which element in an array is not equal to the value which is provided in the...

1 min read

Python | Numpy numpy.ndarray.__neg__()

With the help of numpy.ndarray.__neg__() method of Numpy, one can multiply each and every element of an array with -1. Hence, the resultant array having...

1 min read

Python | Numpy numpy.ndarray.__pos__()

With the help of numpy.ndarray.__pos__() method of Numpy, one can multiply each and every element of an array with 1. Hence, the resultant array having...

1 min read

Python | Numpy numpy.ndarray.__sub__()

With the help of Numpy numpy.ndarray.__sub__(), We can subtract a particular value that is provided as a parameter in the ndarray.__sub__() method. Value wil...

1 min read

Python | Numpy numpy.ndarray.__add__()

With the help of Numpy numpy.ndarray.__add__(), we can add a particular value that is provided as a parameter in the ndarray.__add__() method. Value will be...

1 min read

Article Tags : AI-ML-DS Python-numpy

Corporate & Communications Address:- A-143, 9th Floor, Sovereign Corporate Tower, Sector- 136, Noida, Uttar Pradesh (201305) | Registered Address:- K 061, Tower K, Gulshan Vivante Apartment, Sector 137, Noida, Gautam Buddh Nagar, Uttar Pradesh, 201305

GET IT ON Google Play
Download on the App Store

Company
About Us
Legal
In Media
Contact Us
Advertise with us
GFG Corporate Solution
Placement Training Program
GeeksforGeeks Community

Languages
Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial
Tutorials Archive

DSA
Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Data Science & ML
Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Web Technologies
HTML
CSS
JavaScript
TypeScript
ReactJS

Python Tutorial
Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial

NextJS
Bootstrap
Web Design

Python Interview Question
Django

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

Inteview Preparation

Competitive Programming
Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation
Puzzles

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar
Commerce
World GK

GeeksforGeeks Videos

DSA
Python
Java
C++
Web Development
Data Science
CS Subjects

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved