# Python 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.

1. **Creating numpy:**

# 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)

1. **Find shape,size and dimension of numpy:**

import numpy as np

# Creating array object

arr = np.array( [[ 1, 2, 3],

                 [ 4, 2, 5]] )

# Printing type of arr object

print("Array is of type: ", type(arr))

# Printing array dimensions (axes)

print("No. of dimensions: ", arr.ndim)

# Printing shape of array

print("Shape of array: ", arr.shape)

# Printing size (total number of elements) of array

print("Size of array: ", arr.size)

# Printing type of elements in array

print("Array stores elements of type: ", arr.dtype)

1. **Array creation:**There are various ways to create arrays in NumPy.

# Python program to demonstratearray creation techniques

import numpy as np

# Creating array from list with type float

a = np.array([[1, 2, 4], [5, 8, 7]], dtype = 'float')

print ("Array created using passed list:\n", a)

# Creating array from tuple

b = np.array((1 , 3, 2))

print ("\nArray created using passed tuple:\n", b)

# Creating a 3X4 array with all zeros

c = np.zeros((3, 4))

print ("\nAn array initialized with all zeros:\n", c)

# Create a constant value array of complex type

d = np.full((3, 3), 6, dtype = 'complex')

print ("\nAn array initialized with all 6s.Array type is complex:\n", d)

# Create an array with random values

e = np.random.random((2, 2))

print ("\nA random array:\n", e)

# Create a sequence of integers

# from 0 to 30 with steps of 5

f = np.arange(0, 30, 5)

print ("\nA sequential array with steps of 5:\n", f)

# Create a sequence of 10 values in range 0 to 5

g = np.linspace(0, 5, 10)

print ("\nA sequential array with 10 values between 0 and 5:\n", g)

# Reshaping 3X4 array to 2X2X3 array

arr = np.array([[1, 2, 3, 4],

                [5, 2, 4, 2],

                [1, 2, 0, 1]])

newarr = arr.reshape(2, 2, 3)

print ("\nOriginal array:\n", arr)

print ("Reshaped array:\n", newarr)

# Flatten array

arr = np.array([[1, 2, 3], [4, 5, 6]])

flarr = arr.flatten()

print ("\nOriginal array:\n", arr)

print ("Fattened array:\n", flarr)

# 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)

**2.**