**EX.NO: 5**

**DATE:**

**Create NumPy arrays from Python Data Structures, Intrinsic NumPy objects and Random Functions.**

**AIM:**

To create NumPy arrays using Python data structures, intrinsic NumPy objects, and random functions.

**PROCEDURE:**

**Import NumPy:**

Import the NumPy library to work with arrays and perform mathematical operations.

**Creating NumPy Arrays:**

**From Python Data Structures (Lists, Tuples):** Convert lists and tuples into NumPy arrays.

**Using Intrinsic NumPy Objects:** Create arrays using NumPy’s built-in functions like zeros(), ones(), arange(), and linspace().

**Using Random Functions:** Create arrays using random functions like rand(), randint(), and normal().

**PROGRAM:**

import numpy as np

**# 1. Creating NumPy arrays from Python data structures (Lists and Tuples) # From a List**

list\_data = [1, 2, 3, 4, 5] array\_from\_list = np.array(list\_data)

print("Array from list:", array\_from\_list)

**# From a Tuple**

tuple\_data = (10, 20, 30, 40) array\_from\_tuple = np.array(tuple\_data)

print("Array from tuple:", array\_from\_tuple)

**# 2. Creating NumPy arrays using Intrinsic NumPy objects # Array of zeros**

zeros\_array = np.zeros((3, 3)) # 3x3 array filled with zeros print("Array of zeros:\n", zeros\_array)

**# Array of ones**

ones\_array = np.ones((2, 4)) # 2x4 array filled with ones print("Array of ones:\n", ones\_array)

**# Array with a range of values (similar to Python's range())**

range\_array = np.arange(0, 10, 2) # Array with values from 0 to 10, step of 2 print("Array with range of values:", range\_array)

**# Array with evenly spaced values over a specified interval**

linspace\_array = np.linspace(0, 5, 10) # 10 evenly spaced values between 0 and 5 print("Array with evenly spaced values:", linspace\_array)

**# 3. Creating NumPy arrays using Random Functions**

**# Random array of uniform distribution between 0 and 1**

random\_uniform = np.random.rand(3, 3) # 3x3 array of random values between 0 and 1 print("Random array of uniform distribution:\n", random\_uniform)

**# Random integer array**

random\_integers = np.random.randint(1, 100, (3, 3)) # 3x3 array of random integers between 1 and 100

print("Random integer array:\n", random\_integers)

**# Random normal distribution array (mean=0, std=1)**

random\_normal = np.random.normal(0, 1, (3, 3)) # 3x3 array of values from normal distribution

print("Random normal distribution array:\n", random\_normal)

**OUTPUT:**

**Array from list:** [1 2 3 4 5]

**Array from tuple:** [10 20 30 40]

**Array of zeros:**

[[0. 0. 0.]

[0. 0. 0.]

[0. 0. 0.]]

**Array of ones:**

[[1. 1. 1. 1.]

[1. 1. 1. 1.]]

**Array with range of values:** [0 2 4 6 8]

**Array with evenly spaced values:** [0. 0.55555556 1.11111111 1.66666667 2.22222222

2.77777778

3.33333333 3.88888889 4.44444444 5. ]

**Random array of uniform distribution:**

[[0.20295035 0.57146718 0.87769284]

[0.11446797 0.62550326 0.88471392]

[0.96374569 0.45176781 0.9509921 ]]

**Random integer array:**

[[48 93 58]

[24 11 38]

[22 46 3]]

**Random normal distribution array:**

[[-1.61150704 -1.30343003 -0.28075371]

[-1.2668373 -0.32262893 0.19193538]

[ 0.07882642 0.19507453 0.89284147]]

**RESULT:**

NumPy arrays were successfully created from Python data structures (lists and tuples), intrinsic NumPy objects (zeros, ones, arange, linspace), and random functions (rand, randint, normal).