

1. What is a Python library? Why do we use Python libraries?

A Python library is a collection of pre-written code that can be used to perform specific tasks. Libraries provide a way to reuse code, which helps to save time and effort. We use Python libraries to leverage existing functionality, avoid "reinventing the wheel," and enhance productivity by using tested and optimized code.

2. What is the difference between a NumPy array and a list?

NumPy Array: A NumPy array is a powerful n-dimensional array object that is part of the NumPy library. It supports vectorized operations, which means operations can be performed on entire arrays without the need for explicit loops. NumPy arrays are homogeneous, meaning all elements must be of the same data type.

List: A Python list is a built-in data structure that can hold a collection of items of different data types. Lists are flexible but do not support vectorized operations, making them less efficient for numerical computations compared to NumPy arrays.

3. Find the shape, size, and dimension of the following array:

```
import numpy as np
```

```
array = np.array([[1.2, 3, 4], [5.6, 7, 8], [0.10, 1, 12]])
```

```
shape = array.shape # Shape: (3, 3)
```

```
size = array.size # Size: 9
```

```
dimension = array.ndim # Dimension: 2
```

4. Write Python code to access the first row of the following array:

```
array = np.array([[0.23, 4], [5.6, 7.8], [9, 10, 11, 12]])
```

```
first_row = array[0] # Accessing the first row
```

5. How do you access the element at the third row and fourth column from the given NumPy array?

```
array = np.array([[1.2, 3.4], [5, 6, 7.8], [9, 10, 11, 12]])
```

```
element = array[2, 3] # Accessing the element at the third row and fourth column
```

6. Write code to extract all odd-indexed elements from the given NumPy array:

```
array = np.array([[0.23, 4], [5.8, 6.7, 8], [9, 10, 1, 12]])
```

```
odd_indexed_elements = array[:, 1::2] # Extracting odd-indexed elements
```

7. How can you generate a random 3x3 matrix with values between 0 and 12?

```
random_matrix = np.random.randint(0, 13, size=(3, 3)) # Random 3x3 matrix with values  
between 0 and 12
```

8. Describe the difference between `np.random.rand` and `np.random.randn`?

`np.random.rand`: Generates random numbers from a uniform distribution over the interval $[0, 1)$. The numbers are drawn from a uniform distribution.

`np.random.randn`: Generates random numbers from a standard normal distribution (mean 0, variance 1). The numbers are drawn from a Gaussian distribution.

9. Write code to increase the dimension of the following array:

```
array = np.array([[0.23, 4], [5.6, 7.8], [9, 10, 1, 12]])
```

```
expanded_array = np.expand_dims(array, axis=0) # Increasing the dimension
```

10. How to transpose the following array in NumPy?

```
array = np.array([[1.23, 4], [5.6, 7, 8], [9.10, 1, 12]])
```

```
transposed_array = np.transpose(array) # Transposing the array
```

11. Consider the following matrices:

Matrix A: `[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 1, 12]]`

Matrix B: [[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]]

Perform the following operations:

1) Index-wise multiplication:

```
A = np.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 1, 12]])
```

```
B = np.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])
```

```
index_wise_multiplication = A * B
```

2) Matrix multiplication:

```
matrix_multiplication = np.dot(A, B.T) # Using transpose for proper multiplication
```

3) Add both matrices:

```
addition = A + B
```

4) Subtract matrix B from A:

```
subtraction = A - B
```

5) Divide Matrix B by A:

```
division = B / A
```

12. Which function in NumPy can be used to swap the byte order of an array?

The function `numpy.swapaxes()` can be used to swap the axes of an array, but to specifically swap the byte order, you can use `array.byteswap()`.

13. What is the significance of the `np.linalg.inv` function?

14. What does the `np.reshape` function do, and how is it used?

The `np.reshape` function changes the shape of an array without changing its data. For example, if you have a 1D array of 6 elements, you can reshape it into a 2D array of

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

```
reshaped_array = np.reshape(array, (2, 3)) # Reshaping to 2 rows and 3 columns
```

15. What is broadcasting in NumPy?

Broadcasting is a powerful mechanism that allows NumPy to perform operations on arrays of different shapes. When performing arithmetic operations, NumPy automatically expands the smaller array across the larger array so that they have compatible shapes. This eliminates the need for explicit replication of data and enables efficient computation.