NumPy

,

Numpy which stands for Numerical Python, is a fundamental package for numerical computations in Python. It introduces powerful n-dimensional array objects and provides a variety of tools for working with these arrays. Alongside its core array manipulation capabilities, NumPy also offers comprehensive mathematical functions, random number generation, linear algebra routines, Fourier analysis, and more.

Creation of NumPy

Developed initially by Travis Oliphant in 2005,

Key Features of NumPy

1. Efficient Array Storage

NumPy arrays are up to 50 times faster than traditional Python lists. The key reasons are:

Locality of reference: Arrays are stored contiguously in memory, making access patterns more predictable and hence more cache-friendly.

Type consistency: NumPy arrays contain homogeneous elements, reducing the overhead of type checking and allowing seamless hardware-level optimizations.

2. Array Operations

Creating Arrays: Use `np.array()` to create NumPy arrays

- import `numpy as np`

```
arr = np.array([1, 2, 3])
```

- **Array of the Same Elements**:

```
ones = np.ones((3,3))    # Create a 3x3 array filled with ones
```

Array Shape and Reshape:

Shape: Retrieve the shape of an array with `arr.shape`.

Reshape: Change the dimensions with `arr.reshape(new_shape)`.

Copying vs. Viewing Arrays:

Copy: `arr.copy()` creates a deep copy of the array.

View: Slicing an array creates a view, not a copy.

3. Array Manipulation

Iterating: Loop through array elements.

Searching: Use `np.where()` to search for elements.

Sorting: Arrays can be sorted in-place with `np.sort()`.

4. Advanced Operations

Diagonalizing Matrices:

- `diag_matrix = np.diag([1, 2, 3])  # Create a diagonal matrix`

- **Broadcasting**:

Enables operations on arrays of different sizes by "stretching" the smaller array.

Follows rules like dimension compatibility and stretching dimensions with size 1 to align with larger arrays.