

1. What is vectorization in NumPy?

Vectorization means performing operations on **entire arrays at once**, instead of using Python loops.

✓ NumPy applies operations **element-wise** using optimized C code → **faster execution**

```
import numpy as np

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

c = a + b    # vectorized operation
```

✗ Without vectorization (slow):

```
python

c = []
for i in range(len(a)):
    c.append(a[i] + b[i])
```

2. Difference between `reshape()` and `resize()`

- `reshape()` → Changes shape **without changing data**, returns a **new array**
- `resize()` → Changes shape **in-place**, may **change data**

Feature	<code>reshape()</code>	<code>resize()</code>
Changes original array	✗ No	✓ Yes
Returns new array	✓ Yes	✗ No
Affects data values	✗ No	✓ May change
Memory reallocation	✗ No	✓ Yes
Safe to use	✓ Yes	⚠ Risky

```

a = np.array([1,2,3,4])

b = a.reshape(2,2)    # new array
a.resize(2,2)         # modifies original

```

3. What are NumPy dimensions and axes?

- **Dimensions (ndim):** Number of levels in an array
- **Axis:** Direction along which operations are performed

Example:

```
python
```

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

- `a.ndim` → 2D
- **Axis 0** → columns (vertical)
- **Axis 1** → rows (horizontal)

```
python
```

```
np.sum(a, axis=0) # column-wise sum  
np.sum(a, axis=1) # row-wise sum
```

4. What is slicing in NumPy arrays?

Slicing is extracting a **portion of an array** using index ranges.

```
python
```

```
a = np.array([10,20,30,40,50])
```

```
a[1:4]    # [20 30 40]
```

For 2D arrays:

```
python
```

```
a[:, 1]    # all rows, 2nd column
```

📌 Format:

```
python
```

```
array[start : stop : step]
```

5. How does NumPy help in mathematical computations for AI?

NumPy is the **core mathematical engine** behind AI and Machine Learning.

It helps by:

- ✓ **Vectorization** – performs operations on whole arrays (no loops)
- ✓ **Fast execution** – implemented in optimized C code
- ✓ **Linear algebra support** – matrices, dot products, eigenvalues
- ✓ **Efficient memory usage** – handles large datasets
- ✓ **Foundation for AI libraries** – used by TensorFlow, PyTorch, Scikit-learn