

1. What is vectorization in NumPy?

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

It makes the code **faster, shorter, and more efficient**.

Example:

```
import numpy as np

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

c = a + b
print(c)
```

Output:

```
[5 7 9]
```

No for loop is used.

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

<code>reshape()</code>	<code>resize()</code>
Returns a new array	Modifies the original array
Size must remain same Size can change	Size can change
Does not affect original	Affects original array

Example:

```
import numpy as np

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

b = a.reshape(2, 2)
print(b)

a.resize(2, 2)
print(a)
```

3. What are NumPy dimensions and axes?

Dimension → Number of levels in an array

Axis → Direction along which operations are performed

Example:

```
import numpy as np

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

print(a.ndim)
```

Output

```
2
```

- $axis = 0 \rightarrow$ column-wise
- $axis = 1 \rightarrow$ row-wise

```
print(a.sum(axis=0))
print(a.sum(axis=1))
```

4. What is slicing in NumPy arrays?

Slicing is used to **extract a portion of an array** using index ranges.

Example:

```
import numpy as np

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

print(a[1:4])
```

Output

```
[20 30 40]
```

2D slicing:

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

print(b[:, 1])
```

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

NumPy helps AI by:

- Handling **large datasets efficiently**
- Supporting **matrix operations**
- Providing **fast numerical computation**
- Supporting **linear algebra, statistics, and probability**

Example:

```
import numpy as np

w = np.array([0.2, 0.5, 0.3])
x = np.array([10, 20, 30])

y = np.dot(w, x)
print(y)
```

Used in **neural networks, ML models, and AI algorithms.**