NUMPY IMPORTANT QUESTIONS WITH SOLUTIONS

1. Why NumPy is faster than Python lists?
NumPy is faster because:
- Written in C language
- Uses continuous memory allocation
- Supports vectorization
- Stores homogeneous data
- Performs operations on entire array at once

2. What is Broadcasting?
Broadcasting means performing operations on arrays of different shapes automatically.
Example:
```
import numpy as np
a = np.array([1,2,3])
b = 2
print(a + b)
```

3. What is Vectorization?
Vectorization means performing operations on entire array without loops.
Example:
```
import numpy as np
a = np.array([1,2,3,4])
print(a * 2)
```

Importance:
- Faster execution
- Less code
- Used in machine learning

4. How NumPy integrates with Machine Learning?
NumPy is base for ML libraries:
- Data storage
- Matrix operations
- Linear algebra
Libraries: Scikit-learn, TensorFlow, Pandas

5. Advantages of NumPy in Industry
- High performance
- Memory efficient
- Mathematical functions
- Used in AI, ML, Data Science
- Easy integration with other libraries

6. Creating arrays
1D array:
```
import numpy as np
a = np.array([1,2,3,4])
```

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

7. Indexing and slicing
1D:
```
a = np.array([10,20,30,40])
print(a[1:3])
```

2D:

```
b = np.array([[1,2,3],[4,5,6]])
print(b[0,1])
print(b[0:2,1:3])
```

8. NumPy properties

shape – dimensions
size – total elements
dtype – datatype
ndim – number of dimensions
ndmin – minimum dimension

Example:

```
a = np.array([[1,2,3],[4,5,6]])
print(a.shape)
print(a.size)
print(a.dtype)
print(a.ndim)
```

9. Statistical operations

```
a = np.array([10,20,30,40,50])
print(np.sum(a))
print(np.mean(a))
print(np.median(a))
print(np.std(a))
print(np.max(a))
print(np.min(a))
```