

NumPy Full Notes (Bangla + English Examples)

Beginner

1. Introduction & Installation

```
# Install NumPy
!pip install numpy
import numpy as np
```

2. Creating Arrays

```
arr1 = np.array([1,2,3])      # Manual array
arr2 = np.zeros((2,3))        # All zeros
arr3 = np.ones((3,2))         # All ones
arr4 = np.arange(0,10,2)      # Range array
arr5 = np.linspace(0,1,5)     # Evenly spaced
```

3. Array Attributes

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

4. List vs NumPy Array

```
# List operations slow, NumPy fast and vectorized
list1 = [1,2,3]
arr1 = np.array(list1)
arr2 = arr1 * 2  # Vectorized multiplication
```

5. Indexing (1D,2D,3D)

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

6. Slicing

```
arr = np.arange(10)
print(arr[2:8:2]) # start:end:step
```

7. Boolean Indexing

```
arr = np.arange(5)
print(arr[arr>2]) # [3 4]
```

8. Fancy Indexing

```
arr = np.array([10,20,30,40])
print(arr[[0,3]]) # [10 40]
```

Intermediate

1. Element-wise Operations

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

2. Broadcasting

```
a = np.array([[1,2],[3,4]])
b = np.array([10,20])
print(a+b)
```

3. Universal Functions

```
arr = np.array([1,4,9])
print(np.sqrt(arr), np.exp(arr), np.log(arr))
```

4. Aggregations

```
arr = np.array([1,2,3,4])
print(np.sum(arr), np.mean(arr), np.median(arr), np.std(arr), np.var(arr))
```

5. Min/Max

```
arr = np.array([1,2,3,4])
print(np.min(arr), np.max(arr), np.argmin(arr), np.argmax(arr))
```

6. Linear Algebra Basics

```
A = np.array([[1,2],[3,4]])
B = np.array([[5,6],[7,8]])
print(np.dot(A,B))
print(A @ B)    # matmul
print(A.T)      # transpose
```

7. Determinant & Inverse

```
print(np.linalg.det(A))
print(np.linalg.inv(A))
```

8. Eigenvalues & Eigenvectors

```
values, vectors = np.linalg.eig(A)
print(values)
print(vectors)
```

Advanced

1. Reshaping

```
arr = np.arange(6)
print(arr.reshape(2,3))
print(arr.ravel())
print(arr.flatten())
```

2. Stacking & Concatenation

```
a = np.array([1,2])
b = np.array([3,4])
print(np.hstack((a,b)))
```

```
print(np.vstack((a,b)))  
print(np.concatenate((a,b)))
```

3. Splitting

```
arr = np.arange(10)  
print(np.split(arr,2))  
print(np.array_split(arr,3))
```

4. Random Numbers

```
print(np.random.rand(2,3))  
print(np.random.randn(3,3))  
print(np.random.randint(10,50,(2,4)))
```

5. Shuffling & Permutation

```
arr = np.array([1,2,3,4])  
np.random.shuffle(arr)  
print(arr)  
print(np.random.permutation(arr))
```

6. Random Seed

```
np.random.seed(42)  
print(np.random.randint(1,100,5))
```

7. Conditional Selection

```
arr = np.array([1,2,3,4,5])  
print(np.where(arr>3))  
mask = arr>2  
print(arr[mask])
```

8. Advanced Broadcasting

```
a = np.array([[1],[2],[3]])  
b = np.array([10,20,30])  
print(a+b)
```

9. Saving & Loading Data

```
arr = np.array([1,2,3])
np.save('my_array.npy', arr)
loaded = np.load('my_array.npy')
np.savetxt('my_array.txt', arr, delimiter=',')
loaded_txt = np.loadtxt('my_array.txt', delimiter=',')
```

10. Performance Optimization

```
# Vectorization instead of loop
arr = np.arange(1,1000000)
squared = arr**2 # Fast

# Memory optimization
dtype_arr = np.arange(1000, dtype=np.int8)
arr += 5 # In-place operation
```