

Data Handling with NumPy & Pandas

NumPy Definition:

- NumPy is a Python library used for working with arrays and fast numerical computations. It also has functions for working in domain of linear algebra, fourier transform, and matrices.
- NumPy is widely used for raw numerical operations and scientific computing.

Key Concepts in NumPy:

- ndarray → N-dimensional array, like a list but faster and more efficient.
- Vectorized operations → Perform operations on entire arrays without writing loops.

Applications

- Array creation and manipulation
- Broadcasting
- Indexing and slicing
- Random number generation
- Linear algebra operations

NumPy Example:

```
1. import numpy as np
2. # Create a 2D array
3. arr = np.array([[1, 2, 3], [4, 5, 6]])

4. # Reshape the array
5. reshaped = arr.reshape(3, 2)
6. # [[1, 2], [3, 4], [5, 6]]

7. # Concatenate arrays
8. arr2 = np.array([[7, 8, 9]])
9. combined = np.vstack((arr, arr2))
10. # [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

11. # Arithmetic operation on whole array
12. scaled = arr * 10
13. # [[10, 20, 30], [40, 50, 60]]

14. # Filtering (Boolean indexing)
```

```
15. filtered = arr[arr > 3]
16. # [4, 5, 6]
```

Pandas Definition:

- A Python library built on top of NumPy that provides high-level data structures (Series and DataFrame) and tools for data manipulation, cleaning, analysis, and handling mixed data types (numbers, strings, dates, etc.).
- It works with labeled data and supports heterogeneous data types.

Key Concepts in Pandas:

- Filtering, grouping, sorting
- Handling missing data
- Merging, joining, and reshaping
- Reading/writing from CSV, Excel, SQL, etc.

Pandas Example:

```
1. import pandas as pd

2. # Data Frames
3. data = {
4.     'Name': ['Ali', 'Sara', 'Azhar', 'Ali'],
5.     'Age': [25, 30, 22, 25],
6.     'Salary': [50000, 60000, 55000, 50000]
7. }
8. df = pd.DataFrame(data)

9. # Grouping
10. grouped = df.groupby('Name')['Salary'].mean()
11. # Ali: 50000, Azhar: 55000, Sara: 60000

12. # Merging DataFrames
13. df2 = pd.DataFrame({
14.     'Name': ['Ali', 'Sara', 'Azhar'],
15.     'Department': ['IT', 'HR', 'Finance']
```

```
16. })  
17. merged = pd.merge(df, df2, on='Name')  
  
18. # Sorting  
19. sorted_df = df.sort_values(by='Salary', ascending=False)  
  
20. # Pivot Table  
21. pivot = df.pivot_table(values='Salary', index='Name', aggfunc='mean')
```