Exploring Python Libraries: NumPy & Pandas

- A Guide to Efficient Data Handling in Python
- Presented by: [Hir Chauhan]
- Date: [5/04/2025]

Why Use Python Libraries?

- - Simplify complex tasks
- - Boost efficiency in data analysis
- - Widely used in data science and machine learning
- - NumPy and Pandas are foundational tools

Introduction to numpy

Introduction to NumPy

- - Stands for Numerical Python
- - Offers support for large multi-dimensional arrays
- - Provides mathematical functions and linear algebra tools
- - Basis for many other Python libraries

NumPy Key Features

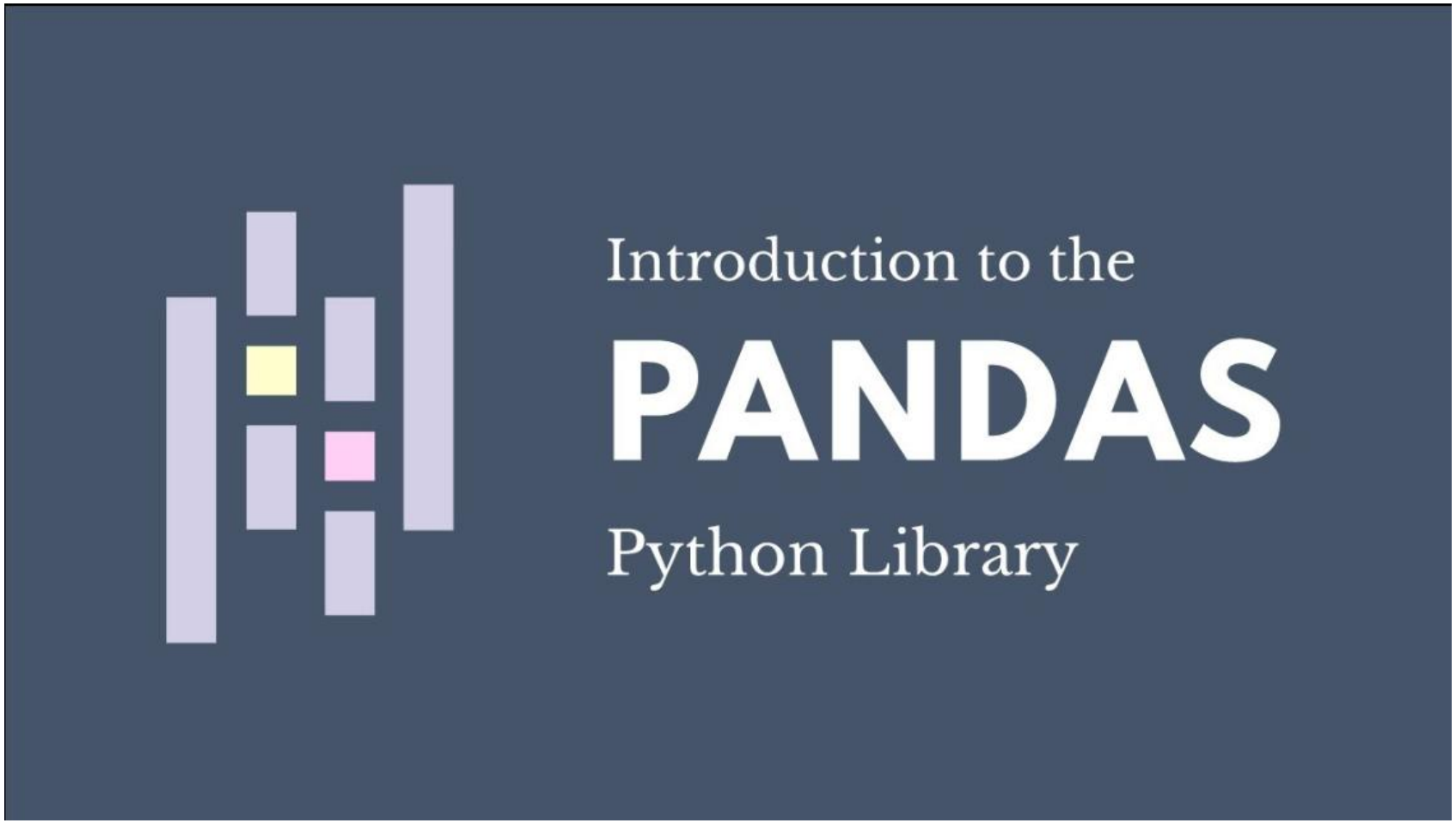
- - Fast and memory-efficient arrays
- - Broadcasting functions
- - Support for vector and matrix operations
- - Random number generation

NumPy Code Example

- import numpy as np

- a = np.array([1, 2, 3])
- b = np.array([4, 5, 6])
- print(a + b) # Output: [5 7 9]

Introduction to pandas



Introduction to Pandas

- - Built on top of NumPy
- - Used for data manipulation and analysis
- - Offers DataFrame and Series data structures
- - Powerful for handling structured data (e.g., CSV, Excel, SQL)

Pandas Key Features

- - Easy handling of missing data
- - Powerful group-by functionality
- - Time-series support
- - Data alignment and reshaping

Pandas Code Example

- import pandas as pd

- data = {'Name': ['Alice', 'Bob'], 'Age': [25, 30]}
- df = pd.DataFrame(data)
- print(df)

Comparing NumPy and Pandas

- Feature | NumPy | Pandas
- ---|---|---
- Data Type | Arrays | DataFrames
- Performance | Faster for math ops | Better for data analysis
- Flexibility | Less flexible | Highly flexible

Where Are These Used?

- - NumPy: Scientific computing, image processing, ML model inputs
- - Pandas: Data cleaning, financial analysis, reporting

Wrapping Up

- - NumPy is essential for numerical computations
- - Pandas makes data handling intuitive and powerful
- - Together, they form the backbone of Python data science

Questions & Discussion

- 🎤 Feel free to ask anything!