Introduction to Pandas

Pandas Overview and Basics

📌 What Is Pandas?

Pandas is a fast, powerful, flexible, and easy-to-use open-source **data analysis and manipulation tool**, built on top of the **Python** programming language. It is the go-to library for **structured data (tables, spreadsheets, CSVs, SQL results)**.

The name "Pandas" comes from **"Panel Data"**, a term used in econometrics.

🔍 Why Use Pandas?

Feature	Benefit
Powerful Data Structures	`Series` and `DataFrame`
Handles Missing Data	NaN support + built-in fill/clean
Fast and Vectorized Operations	Optimized with NumPy under the hood
Built-in I/O Support	Read/write from CSV, Excel, SQL
Grouping, Merging, Reshaping	SQL-like data transformations

Pandas is widely used in **data science, finance, machine learning, and data engineering**.

📦 Installing Pandas

You can install Pandas using pip or conda:

```
pip install pandas
# or
conda install pandas
```

🧱 Pandas Core Data Structures

1. Series – One-dimensional labeled array:

```
import pandas as pd

s = pd.Series([10, 20, 30, 40], index=["a", "b", "c", "d"])
print(s["b"])  # Output: 20
```

2. DataFrame – Two-dimensional table:

```
data = {
    "Name": ["Alice", "Bob", "Charlie"],
    "Age": [25, 30, 35]
}
df = pd.DataFrame(data)
print(df.head())
```

Real-World Usage & Key Functions

📊 Reading and Writing Data

```
# Read a CSV file
df = pd.read_csv("data.csv")

# Write to Excel
df.to_excel("output.xlsx", index=False)
```

Pandas supports many formats: CSV, Excel, JSON, SQL, Parquet, HTML, and clipboard.

🔍 Data Inspection

```
df.head()       # First 5 rows
df.tail(3)      # Last 3 rows
df.shape        # (rows, columns)
df.info()       # Data types and non-null counts
df.describe()   # Summary statistics
```

🧹 Data Cleaning and Filtering

```
df.dropna()                 # Remove rows with missing values
df.fillna(0)                # Replace missing values with 0
```

```
df[df["Age"] > 30]              # Filter rows
df["Age"] = df["Age"].astype(int)  # Change data type
```

🔄 Merging and Grouping

```
# Group by column
df.groupby("Department")["Salary"].mean()

# Merge two dataframes
pd.merge(df1, df2, on="EmployeeID")
```

Pandas enables **SQL-like joins**, **pivot tables**, and **aggregations**.

📈 Visualization

Although Pandas itself is not a visualization library, it integrates seamlessly with **Matplotlib** and **Seaborn**:

```
import matplotlib.pyplot as plt

df["Salary"].plot(kind="hist")
plt.show()
```

🏁 Conclusion

Pandas is an essential tool for any data professional. With it, you can:

- Load and clean data efficiently
- Explore and visualize trends
- Perform complex data transformations

Whether you're analyzing spreadsheets or building ML pipelines, **Pandas simplifies data workflows**.

> “Pandas makes working with structured data as simple and expressive as working with Python lists and dictionaries.”
