What is Pandas?

Pandas is a powerful Python library for:

- Data analysis and manipulation
- Handling structured data (tables, spreadsheets, CSV files, etc.)
- Working with **DataFrames** and **Series**

Install Pandas (if not already):

```
pip install pandas
```

Import it:

```
import pandas as pd
```

---

1. Pandas Data Structures

Series

A **one-dimensional labeled array** (like an Excel column).

```
import pandas as pd

s = pd.Series([10, 20, 30, 40])
print(s)
```

Output:

```
0   10
1   20
2   30
3   40
dtype: int64
```

You can also label your index:

```
s = pd.Series([10, 20, 30], index=['A', 'B', 'C'])
print(s['B'])  # 20
```

DataFrame

A **two-dimensional table** (like an Excel sheet).

```
data = {
  'Name': ['John', 'Sara', 'Mike'],
  'Age': [25, 30, 22],
  'City': ['New York', 'London', 'Delhi']
}
df = pd.DataFrame(data)
print(df)
```

Output:

```
  Name  Age     City
0  John   25  New York
1  Sara   30    London
2  Mike   22     Delhi
```

2. Reading and Writing Data

Read data:

```
df = pd.read_csv('data.csv')       # From CSV file
df = pd.read_excel('data.xlsx')    # From Excel file
```

Write data:

```
df.to_csv('output.csv',  index=False)
df.to_excel('output.xlsx', index=False)
```

3. Viewing Data

```
df.head()    # First 5 rows
df.tail(3)   # Last 3 rows
df.info()    # Summary (columns, data types)
df.describe() # Statistics (mean, std, etc.)
```

4. Selecting Data

By column name:

```
print(df['Name'])
print(df[['Name', 'Age']])
```

By row index:

```
print(df.iloc[0])     # First row (by position)
print(df.loc[1])      # Row with index label 1
```

By condition:

```
print(df[df['Age'] > 25])
```

5. Adding / Modifying / Deleting Columns

```
df['Salary'] = [50000, 60000, 40000]  # Add new column
df['Age'] = df['Age'] + 1             # Modify column
df.drop('City', axis=1, inplace=True)  # Delete column
```

6. Filtering Rows

```
df[df['Age'] > 25]          # Filter condition
df[(df['Age'] > 25) & (df['Salary'] > 50000)]   # Multiple conditions
```

7. Sorting

```
df.sort_values(by='Age')              # Ascending
df.sort_values(by='Salary', ascending=False)  # Descending
```

8. Aggregations and Statistics

```
df['Age'].mean()
df['Salary'].sum()
```

```
df['Age'].max()
df['Age'].min()
```

Or with multiple columns:

```
df.groupby('City')['Salary'].mean()
```

---

9. Handling Missing Data

```
df.isnull().sum()       # Check missing values
df.fillna(0, inplace=True)   # Replace missing with 0
df.dropna(inplace=True)     # Drop rows with missing values
```

---

10. Merge / Join / Concatenate

Merge:

```
df1 = pd.DataFrame({'ID':[1,2,3], 'Name':['A','B','C']})
df2 = pd.DataFrame({'ID':[1,2,4], 'Salary':[100,200,300]})

merged = pd.merge(df1, df2, on='ID', how='inner')
print(merged)
```

Concatenate:

```
df1 = pd.DataFrame({'A':[1,2], 'B':[3,4]})
df2 = pd.DataFrame({'A':[5,6], 'B':[7,8]})

result = pd.concat([df1, df2])
print(result)
```

---

11. Reshaping Data

Pivot Table:

```
pivot = df.pivot_table(values='Salary', index='City', aggfunc='mean')
```

Transpose:

```
df.T
```

---

12. Basic Plotting

```
import matplotlib.pyplot as plt

df.plot(kind='bar', x='Name', y='Salary')
plt.show()
```

Example: Combine Everything

```
import pandas as pd

# Create DataFrame
data = {
    'Name': ['John', 'Sara', 'Mike', 'Anna'],
    'Age': [25, 30, 22, 28],
    'City': ['New York', 'London', 'Delhi', 'Paris'],
    'Salary': [50000, 60000, 40000, 55000]
}
df = pd.DataFrame(data)

# Filter
print("Employees older than 25:\n", df[df['Age'] > 25])

# Add new column
df['Bonus'] = df['Salary'] * 0.1

# Sort
df = df.sort_values(by='Salary', ascending=False)

print("\nFinal DataFrame:\n", df)
```