# Pandas Operations Explained

Pandas is a Python library mainly used for **data analysis**, **data cleaning**, and **data manipulation**
It mainly works with two structures:

- **Series** → 1-Dimensional data (like a single column)
- **Data Frame** → 2-Dimensional data (like an Excel sheet)

Here's a detailed list of all the **important operations**:

---

**1 Creating Data**

- **Series**

python

import pandas as pd

s = pd Series([10, 20, 30])

- **DataFrame**

python

data = {'Name': ['John', 'Alice'], 'Age': [25, 30]}

df = pd DataFrame(data)

---

**2 Reading Data**

- **From CSV**

python

df = pd read_csv('file csv')

- **From Excel**

python

df = pd read_excel('file xlsx')

- **From JSON**

python

df = pd  read_json('file  json')

---

**3  Viewing Data**

- **First Few Rows**

python

df  head()

- **Last Few Rows**

python

df  tail()

- **Summary (Shape, Columns, Info)**

python

df  shape   # (rows, columns)

df  columns  # column names

df  info()  # data types + nulls

df  describe() # statistical summary

---

**4  Selecting Data**

- **Select Column**

python

df['Name']

- **Select Multiple Columns**

python

df[['Name', 'Age']]

- **Select Row by Index (Label-based: loc)**

python

df loc[0] # first row

- **Select Row by Position (Index-based: iloc)**

python

df iloc[0]

---

**5 Filtering Data (Condition-Based Selection)**

python

df[df['Age'] > 25]

(Select rows where Age > 25)

---

**6 Sorting Data**

- **Sort by Column**

python

df sort_values('Age')

- **Descending Order**

python

df sort_values('Age', ascending=False)

---

## 7 Adding / Updating Data

- **Add New Column**

python

```
df['Salary'] = [50000, 60000]
```

- **Update Existing Column**

python

```
df['Age'] = df['Age'] + 1
```

---

## 8 Deleting Data

- **Delete a Column**

python

```
df   drop('Salary', axis=1, inplace=True)
```

- **Delete a Row**

python

```
df   drop(0, axis=0, inplace=True)
```

---

## 9 Handling Missing Data

- **Check Missing Values**

python

```
df   isnull()
df   isnull()   sum()
```

- **Drop Missing Values**

python

df dropna()

- **Fill Missing Values**

python

df fillna(0)

---

## 10 GroupBy Operation (Aggregation)

- **Group and Summarize**

python

df groupby('Department') sum()

---

## 11 Merging and Joining Data

- **Merge Two DataFrames**

python

pd merge(df1, df2, on='ID')

- **Join by Index**

python

df1 join(df2)

---

## 12 Pivot Table

- **Create Pivot Table**

python

df pivot_table(values='Sales', index='Region', columns='Product')

---

### 13 Saving Data

- **Save to CSV**

python

df to_csv('new_file csv', index=False)

- **Save to Excel**

python

df to_excel('new_file xlsx', index=False)

---

### 14 Apply Functions to Data

- **Apply a Function to a Column**

python

df['Age'] apply(lambda x: x + 2)

---

### 15 Reshaping Data

- **Melt (Wide to Long)**

python

pd melt(df, id_vars=['Name'])

- **Pivot (Long to Wide)**

python

df pivot(index='Name', columns='Product', values='Sales')

---

### 📋 Summary Table

| Operation | Purpose |
| --- | --- |
| Create | Build Series/DataFrame |
| Read | Load data from files |
| View | Inspect structure and content |
| Select | Choose rows/columns |
| Filter | Condition-based selection |
| Sort | Arrange data |
| Add/Update | Modify columns or rows |
| Delete | Remove data |
| Handle Missing | Fill or drop missing values |
| GroupBy | Group and aggregate |
| Merge/Join | Combine datasets |
| Pivot Table | Summarize data dynamically |
| Save | Export data to files |
| Apply Functions | Transform data using functions |
| Reshape | Change layout (melt/pivot) |