Pandas Notes – Beginner Friendly 📊💌

1. What is Pandas?

Pandas is a Python library used for **data analysis and data manipulation**.

In simple words: - It helps you **store data** in tables - **Clean messy data** - **Analyze data easily** - Used heavily in **Data Science, AI/ML, Data Mining**

Why Pandas is Important?

- Faster than Excel for large data
- Easy syntax
- Industry standard
- Works perfectly with **NumPy, Matplotlib, Seaborn, Scikit-Learn**

---

2. Installing & Importing Pandas

Installation (one time)

```
pip install pandas
```

Import Pandas

```
import pandas as pd
```

👉 `pd` is just an alias (shortcut). Industry standard.

---

3. Core Pandas Data Structures

Pandas has **2 main data structures**:

✂️ Series (1-D Data)

- Like a **single column**
- Similar to array / list

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

👃 Series has: - Values - Index (0,1,2,3 by default)

✒️ DataFrame (2-D Data) 🕐 MOST IMPORTANT

- Like an **Excel table**
- Rows + Columns

```
data = {
    "Name": ["Alice", "Bob", "Charlie"],
    "Age": [20, 21, 22],
    "Marks": [85, 90, 95]
}

df = pd.DataFrame(data)
print(df)
```

---

4. Understanding Rows, Columns & Index

```
print(df.columns)   # Column names
print(df.index)     # Row index
print(df.shape)     # (rows, columns)
```

👃 Important: - Columns → Features - Rows → Records

---

5. Viewing Data (Very Important)

```
df.head()      # First 5 rows
df.tail()      # Last 5 rows
df.info()      # Structure + data types
df.describe()  # Statistical summary
```

👉 `df.describe()` works mainly on **numerical data**

---

6. Accessing Data (Core Concept)

Access Column

```
df["Name"]
```

Access Multiple Columns

```
df[["Name", "Marks"]]
```

Access Rows

Using `iloc` (index based)

```
df.iloc[0]        # First row
df.iloc[0:2]      # First two rows
```

Using `loc` (label based)

```
df.loc[0]
```

---

7. Adding New Columns

```
df["Passed"] = df["Marks"] > 40
```

---

8. Modifying Data

```
df["Marks"] = df["Marks"] + 5
```

---

9. Dropping Rows & Columns

Drop Column

```
df.drop("Age", axis=1, inplace=True)
```

Drop Row

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

👃 `axis=0` → row | `axis=1` → column

---

10. Handling Missing Values (Very Important)

Check Missing Values

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

Remove Missing Values

```
df.dropna()
```

Fill Missing Values

```
df.fillna(0)
```

---

11. Filtering Data (Real-World Skill)

```
df[df["Marks"] > 80]
```

Multiple Conditions:

```
df[(df["Marks"] > 80) & (df["Age"] > 20)]
```

---

12. Sorting Data

```
df.sort_values(by="Marks", ascending=False)
```

---

13. Reading & Writing Files

Read CSV

```
df = pd.read_csv("data.csv")
```

Write CSV

```
df.to_csv("output.csv", index=False)
```

14. Commonly Used Functions (Must Remember)

Function	Purpose
`head()`	View top rows
`info()`	Data summary
`describe()`	Stats
`shape`	Size
`columns`	Column names
`isnull()`	Missing data

---

15. How to Practice Pandas Properly 🗜️

👈 Create small DataFrames manually 👈 Load CSV files 👈 Perform filtering & sorting daily 👈 Apply Pandas on **college DBMS / CN datasets**

---

16. Learning Path (Suggested)

1. Series → DataFrame
2. Indexing → Filtering
3. Missing values
4. File handling
5. GroupBy (next level)
6. Merge & Join (advanced)

Final Truth 💡

Pandas is **not theory**. It is a **tool**. You only master it by **writing code daily**.

If you want, next I can create: - 😟 **Daily Pandas Practice Sheet** - 😟 **Interview-oriented Pandas questions** - 😟 **Mini Pandas projects**

Tell me how deep you want to go 🟫