Gowtham SB
www.linkedin.com/in/sbgowtham/ Instagram - @dataengineeringtamil

🧠 Pandas Complete Guide (With vs Without Pandas)

📘 What is Pandas?

- Python library to work with **structured data** (tables)
- Helps in **loading, analyzing, processing, and exporting** data
- Core structures:
    - `Series` (1D)
    - `DataFrame` (2D)

✅ 1. Create DataFrame Without File

```
import pandas as pd

data = {
   'Name': ['Alice', 'Bob', 'Charlie'],
   'Age': [25, 30, 35],
   'City': ['New York', 'London', 'Mumbai']
}

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

✅ 2. Read & Write CSV

➤ Read

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

➤ Read without headers

```
df = pd.read_csv("people.csv", header=None, names=['Name', 'Age', 'City'])
```

➤ Write

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

✅ 3. Explore the Data

```
df.head()
df.info()
df.describe()
df.columns
df.dtypes
```

✅ 4. Select / Filter Data

```
df[df['Age'] > 30]
df[['Name', 'City']]
df.iloc[0]            # By row index
df.loc[0, 'Name']     # Specific cell
```

✅ 5. Add / Modify Columns

```
df['Age_in_5_years'] = df['Age'] + 5
```

✅ 6. Grouping & Aggregation

```
df['Total'] = df['Price'] * df['Quantity']
summary = df.groupby('Product')['Total'].sum().reset_index()
```

✅ 7. JOIN Example (WITH Pandas)

```
# Customers table
customers = pd.DataFrame({
    'CustomerID': [1, 2, 3],
    'Name': ['Alice', 'Bob', 'Charlie']
})

# Orders table
orders = pd.DataFrame({
    'OrderID': [101, 102, 103, 104],
    'CustomerID': [1, 2, 1, 3],
    'Product': ['Shirt', 'Pant', 'Shoes', 'Hat']
})

# INNER JOIN on CustomerID
result = pd.merge(customers, orders, on='CustomerID', how='inner')
print(result)
```

😓 JOIN Example (WITHOUT Pandas)

```
customers = [
    {'CustomerID': 1, 'Name': 'Alice'},
    {'CustomerID': 2, 'Name': 'Bob'},
    {'CustomerID': 3, 'Name': 'Charlie'}
]

orders = [
    {'OrderID': 101, 'CustomerID': 1, 'Product': 'Shirt'},
    {'OrderID': 102, 'CustomerID': 2, 'Product': 'Pant'},
    {'OrderID': 103, 'CustomerID': 1, 'Product': 'Shoes'},
    {'OrderID': 104, 'CustomerID': 3, 'Product': 'Hat'}
]

result = []

for c in customers:
    for o in orders:
        if c['CustomerID'] == o['CustomerID']:
            result.append({
```

```
            'CustomerID': c['CustomerID'],
            'Name': c['Name'],
            'OrderID': o['OrderID'],
            'Product': o['Product']
        })

for row in result:
    print(row)
```

🛑 Imagine doing `LEFT JOIN`, `OUTER JOIN`, `GROUP BY`, `SORT`, `FILTER` all like this in raw Python!

✅ 8. Read JSON into DataFrame

```
df = pd.read_json("people.json")
```

✅ 9. Schema Information Like printSchema()

```
df.dtypes
df.info()
```

🧪 Mini Project: Sales Summary Generator

◇ CSV File: `sales.csv`

```
Date,Product,Price,Quantity
2025-01-01,Shirt,500,2
2025-01-01,Pant,800,1
2025-01-02,Shirt,500,1
2025-01-02,Pant,800,3
```

◇ Objective:

- Load sales
- Calculate total per row
- Group by product
- Sort by total sales

◇ Code:

```
import pandas as pd

df = pd.read_csv("sales.csv")
df['Total'] = df['Price'] * df['Quantity']
summary = df.groupby('Product')['Total'].sum().reset_index()
summary = summary.sort_values(by='Total', ascending=False)

print(summary)
```

◇ Output:

```
 Product  Total
0   Pant   3200
1  Shirt   1500
```

🎯 Interview Story to Explain

"In my last project, I had to analyze raw sales data provided as CSVs from multiple stores. Instead of manually doing this in Excel, I used Pandas.

I created a script that:

- Loaded the file
- Calculated per-row totals
- Grouped them by product

- Sorted to find top-selling items

A key line was:

```
df.groupby('Product')['Total'].sum().reset_index()
```

which did what took 15+ Excel clicks in just one line.

I even showed the team how complex joins could be simplified using Pandas vs traditional for-loops in Python. It saved hours every week."

💡 Summary Cheatsheet

Task	Pandas Command
Load CSV	`pd.read_csv()`
Load without header	`pd.read_csv(..., header=None, names=[])`
Explore schema	`df.info()`, `df.dtypes`
Filter rows	`df[df['col'] > x]`
Add column	`df['new'] = ...`
Group by & aggregate	`df.groupby().sum().reset_index()`
Join tables	`pd.merge()`
Read JSON	`pd.read_json()`
Save CSV	`df.to_csv()`

About the Author

Gowtham SB is a **Data Engineering expert, educator, and content creator** with a passion for **big data technologies, as well as cloud and Gen AI** . With years of experience in the field, he has worked extensively with **cloud platforms, distributed systems, and data pipelines**, helping professionals and aspiring engineers master the art of data engineering.

Beyond his technical expertise, Gowtham is a **renowned mentor and speaker**, sharing his insights through engaging content on **YouTube and LinkedIn**. He has built one of the **largest Tamil Data Engineering communities**, guiding thousands of learners to excel in their careers.

Through his deep industry knowledge and hands-on approach, Gowtham continues to **bridge the gap between learning and real-world implementation**, empowering individuals to build **scalable, high-performance data solutions**.

Socials

📽️ **YouTube** - https://www.youtube.com/@dataengineeringvideos

📸 **Instagram** - https://instagram.com/dataengineeringtamil

📸 **Instagram** - https://instagram.com/thedatatech.in

🤝 **Connect for 1:1** - https://topmate.io/dataengineering/

💼 **LinkedIn** - https://www.linkedin.com/in/sbgowtham/

🌐 **Website** - https://codewithgowtham.blogspot.com

💻 **GitHub** - http://github.com/Gowthamdataengineer

Gowtham SB
www.linkedin.com/in/sbgowtham/ Instagram - @dataengineeringtamil

💬 **Whats App** - https://lnkd.in/g5JrHw8q

📧 **Email** - atozknowledge.com@gmail.com

📱 **All My Socials** - https://lnkd.in/gf8k3aCH