



# Day 4: Pandas Basics – Assignment (40 Marks)

You've learned how to create tables (DataFrames), inspect them, check their structure, and compute summaries with aggregation. This assignment tests your grasp with different question types—fill-in-the-blank, matching, true/false, short answers, and hands-on coding based on real-life scenarios. Show all steps and answers clearly.

## Section A: Fill in the Blanks (8 × 1 = 8 Marks)

1. `pd.DataFrame(...)` creates a \_\_\_\_\_ (pandas object that stores table data).
2. `df.head(n)` shows the first \_\_\_\_ rows of the DataFrame.
3. `df.shape` returns a tuple `(____, ____)` representing rows and columns.
4. `df.columns` lists the \_\_\_\_\_ names.
5. `df.dtypes` shows each column's \_\_\_\_\_.
6. To compute summary stats like sum or mean on your table, you use `df.____()`.
7. `df.groupby('col')` first \_\_\_\_\_ the rows by the values in `col`.
8. After grouping, you call `.agg({'col': 'sum'})` to \_\_\_\_\_ the values in that column.

## Section B: Match the Columns (6 × 1 = 6 Marks)

Match **Column A** with the best description in **Column B**. Write the letter (a–f) next to each number (9–14).

Column A	Column B
9. <code>df.head()</code>	a. Removes rows with missing values
10. <code>df.dropna()</code>	b. Shows column names
11. <code>df.columns</code>	c. Computes summary statistics like sum, mean, max
12. <code>df.agg()</code>	d. Shows the first few rows
13. <code>df.groupby('col')</code>	e. Shows data type of each column
14. <code>df.dtypes</code>	f. Splits the DataFrame into groups based on one column

### Section C: True or False (6 × 1 = 6 Marks)

Write **T** for true, **F** for false:

15. `df.shape` returns `(rows, columns)`.
16. `df.tail()` shows the first 5 rows by default.
17. After `df.groupby('city')`, you can call `.agg()` to summarize each city's data.
18. `df.fillna(0)` removes all rows with missing values.
19. `df.describe()` gives count, mean, min, max for numeric columns.
20. `df.info()` shows row count, column names, and data types.

### Section D: Short-Answer Theory (5 × 2 = 10 Marks)

Answer in 1–2 sentences each.

21. **What is a DataFrame?**
22. **Why do we use `df.head()` before any analysis?**
23. **Explain why checking `df.dtypes` is important.**
24. **How does `df.groupby()` help when you have categories?**
25. **When would you use `df.agg()` instead of individual `.sum()` or `.mean()` calls?**

## Section E: Practical Coding (Real-Life Scenarios) (2 × 5 = 10 Marks)

### E1: Weekly Milk Sales (5 Marks)

A dairy shop records litres of milk sold each day:

```
csharp
Copy
[10, 12, 9, 11, 13, 8, 14]
```

- **Tasks:**

1. Create a DataFrame with columns `day` ( `Mon` , `Tue` , ..., `Sun` ) and `litres` .
2. Display the first 3 rows.
3. Print the DataFrame's shape, columns, and dtypes.

*Show your code and its output.*

### E2: Monthly Expenses Summary (5 Marks)

You track your January expenses:

category	amount (₹)
rent	15 000
food	8 000
utilities	2 000
transport	3 000

- **Tasks:**

1. Create a DataFrame with this data.
2. Use a single pandas command to compute **total**, **average**, **maximum**, and **minimum** expense.
3. Print the results clearly.
- 4.



# The 90–Minute Coding Rule

## The 90–Minute Coding Rule

This rule helps you power through complex coding tasks (like your pandas assignment) in one solid, deep-work session. Here's how it works:

### 1. Set a Clear Intention (5 minutes)

- **Pick one goal:** "Finish Section A–D of the pandas assignment."
- **Write it down** on paper or a sticky note.
- **Remove distractions:** silence phone, close email tabs, clear desk.

### 2. Deep-Work Block (80 minutes)

Your brain does best in ~90-minute rhythms. We'll use 80 minutes of focused coding, divided into mini-phases:

Phase	Duration	What to Do
A. Plan & Warm-Up	5 min	Read instructions; open your Python editor
B. Section A	10 min	Fill in the blanks (8 Q's)
C. Section B	10 min	Match the columns (6 Q's)
D. Section C	10 min	True/False (6 Q's)
E. Section D	15 min	Short theory answers (5 Q's)
F. Section E1	20 min	Code "Weekly Milk Sales" (5 marks)
G. Section E2	20 min	Code "Monthly Expenses Summary" (5 marks)

### 3. Reflection & Wrap-Up (5 minutes)

- **Review** your answers and code.
- **Fix** any typos or logic errors.

- **Save** and **submit** your work.

#### 4. Take a Break (at 90 minutes)

- Walk, stretch, get water.
  - You've earned it!
- 

### Why This Works

1. **Biological Peak:** Our brains cycle about every 90 min—from alert to tired. Use the first 80 min when you're sharp.
  2. **Single-Task Focus:** By sticking to one assignment, you avoid "context switching" (jumping between tasks), which kills productivity.
  3. **Micro-Phase Planning:** Breaking 80 min into 7 clear steps keeps you motivated—each mini-goal feels achievable.
  4. **Built-In Review:** The final 5 min reflection catches small mistakes before they count against you.
  5. **Energy Management:** A proper break prevents burnout and resets your focus for the next task.
- 

### Detailed Roadmap Example

Time	Task	Tip
0–5 min	<b>Prepare:</b> Read assignment, open editor	Keep your intention note visible.
5–15 min	<b>Section A:</b> Fill in the blanks (8 Q's)	Work quickly—you know these definitions well.
15–25 min	<b>Section B:</b> Match columns (6 Q's)	Trust your first instinct on pairs.
25–35 min	<b>Section C:</b> True/False (6 Q's)	Answer directly—no overthinking.
35–50 min	<b>Section D:</b> Theory (5 Q's)	One or two clear sentences each.
50–70 min	<b>Section E1:</b> Weekly Milk Sales (5 m)	Write code step-by-step; test after each line.

70–90 min	<b>Section E2:</b> Monthly Expenses Summary	Use <code>.agg()</code> once for all stats; print clearly.
90 min	<b>Break!</b>	Step away and celebrate your progress.

Follow this 90-Minute Coding Rule, and you'll finish the assignment in one focused burst—no stress, no distractions, just clear progress. Good luck, Sagar!



# CodingWise

Let's code your career

India's #1 Most Trusted Training & Placement Institute