

Pandas QQuestion:

BASIC (Data Creation & Inspection):

- Create a Pandas Series from a Python list.
- Create a DataFrame using a dictionary containing lists.
- Create a DataFrame from a list of tuples.
- Load a CSV file into a DataFrame.
- Display the first 10 rows of a DataFrame.
- Display the last 5 rows of a DataFrame.
- Print column names of a DataFrame.
- Get total rows and columns in a DataFrame.
- Check data types of all columns.
- Convert a DataFrame to a NumPy array.

INTERMEDIATE (Indexing, Selection & Cleaning):

- Select a single column using both bracket and dot notation.
- Select rows using loc and iloc.
- Filter rows where a column value is greater than a given number.
- Filter rows using multiple conditions (AND / OR).
- Add a new column calculated from existing columns.
- Rename one or more columns.
- Drop a column permanently from a DataFrame.
- Drop rows containing missing values.
- Fill missing values using mean / median.
- Replace specific values in a column.

ADVANCED (Transformation, Grouping & Merging):

- Sort a DataFrame by one column in ascending order.
- Sort a DataFrame by multiple columns with different orders.
- Group data by a column and calculate sum.
- Group data by multiple columns and calculate mean.
- Apply multiple aggregation functions using groupby.
- Use apply() to transform a column.
- Merge two DataFrames using an inner join.
- Merge two DataFrames using left and right joins.
- Concatenate multiple DataFrames vertically and horizontally.
- Convert a column to datetime and extract year, month, and day.

Pandas Question from all topic

1. Import pandas and check their version.
2. Read data from an Excel file named Routine.xlsx into a DataFrame.
3. Read data from a CSV file named sales.csv into a DataFrame.
4. Read data from a JSON file named students.json.
5. Create a DataFrame with student details (Name, Age, Roll_No).
6. Save the DataFrame into CSV, Excel, and JSON formats.
7. Display the first 10 rows of a DataFrame using head().
8. Display the last 5 rows of a DataFrame using tail().
9. Use .info() on a DataFrame and explain the output.
10. Use .describe() on an Employee DataFrame containing Name, Age, ID, and Salary.
11. Print the shape of a DataFrame using .shape.
12. List all column names of a DataFrame using .columns.
13. Select only the Name column from a DataFrame.
14. Select both Name and Age columns from a DataFrame.
15. Filter and display all students with Age > 20.
16. Create a new DataFrame of employees with columns Name, Age, and Salary, then export it to an Excel file named Employee_info.xlsx.
17. Add a new column Bonus equal to **20% of Salary**.
18. Update the Salary of the first employee to 60000 using .loc.
19. Increase all salaries by **10%** and update the DataFrame.
20. Remove the column Bonus permanently.
21. Sort the DataFrame by Salary in **ascending order**.
22. Sort the DataFrame by Age in **descending order**.
23. Select employees whose Salary is greater than 70,000.
24. Select employees whose Age is between 23 and 26.
25. Identify missing values in the Age column using isnull().
26. Drop all rows where any value is missing using dropna().
27. Fill missing values in the Salary column with the **mean salary**.
28. Group employees by Age and calculate the **average Salary**.
29. Group employees by Age and find the **maximum Bonus**.
30. Merge two DataFrames (df1 with employee details and df2 with department details) on the column ID.
31. Concatenate two DataFrames vertically (row-wise).