**🐼 DAY 15 – HOME ASSIGNMENTS: Introduction to Pandas**

**🔹 Part A: Creating and Inspecting Series/DataFrames**

1. **Create a Series**
   * Create a Pandas Series from a list of 5 city names.
2. **Create a DataFrame**
   * Create a DataFrame called students with columns:
     + Name (string), Age (int), Marks (float)
3. **Check Properties**
   * Print:
     + .shape
     + .columns
     + .dtypes
     + .info()
     + .describe() (for numeric columns)

**🔹 Part B: Selecting and Indexing**

1. **Column Selection**
   * From the students DF, select the Name column and print it.
2. **Row Selection by Position**
   * Use .iloc to print the first 3 rows.
3. **Row Selection by Label**
   * Set Name as the index and use .loc to fetch a student by name.
4. **Conditional Selection**
   * Select students with Marks > 75.
   * Select students with Age <= 20.

**🔹 Part C: Adding, Modifying, and Removing Columns**

1. **Add a New Column**
   * Add a column Pass which is True if Marks >= 40 else False.
2. **Modify an Existing Column**
   * Increase every student’s marks by 5.
3. **Remove a Column**
   * Drop the Age column and print the resulting DF.

**🔹 Part D: Handling Missing Values**

1. **Introduce Missing Values**
   * Create a DF with columns Name, Age, Marks where some Age and Marks values are NaN.
2. **Check for Missing Values**
   * Use .isnull() and .sum() to count NaNs.
3. **Fill Missing Values**
   * Fill NaN in the Marks column with the average marks.
4. **Drop Missing Values**
   * Drop rows where any NaN occurs.

**🔹 Part E: Aggregation and Grouping**

1. **Summary Stats**
   * Find the total, average, max, and min marks from the DF.
2. **Grouping**
   * Add a column Class (e.g., Class A or Class B).
   * Group the DF by Class and compute the average marks per class.

**🔹 Part F: Miscellaneous Practice**

1. **Sorting**
   * Sort the DF by Marks in ascending and then in descending order.
2. **Rename Columns**
   * Rename columns Name -> Student Name, Marks -> Total Marks.
3. **Export DF**
   * Export the final DF to a .csv file named student\_records.csv.
4. **Apply a Function**
   * Use .apply() to categorize marks:
     + Marks >= 80: Excellent
     + Marks >= 60: Good
     + Marks >= 40: Pass
     + Marks < 40: Fail
5. **Check Duplicates**
   * Identify and remove duplicate rows in the DF.
6. **Merge Two DataFrames**
   * Create another DF with columns Name and Class Teacher.
   * Merge both DFs on the Name column.