***### \*\*Exercise 5: Handling Missing Values\*\*  
#1. Create a DataFrame with missing values:  
# ```python  
# data = {  
# "Name": ["Amit", "Neha", "Raj", "Priya"],  
# "Age": [28, None, 35, 29],  
# "City": ["Delhi", "Mumbai", None, "Chennai"]  
# }  
# df = pd.DataFrame(data)  
# ```  
#2. Fill missing values in the `"Age"` column with the average age.  
#3. Drop rows where any column has missing data.***import pandas as pd  
  
data = {  
 "Name": ["Amit", "Neha", "Raj", "Priya"],  
 "Age": [28, None, 35, 29],  
 "City": ["Delhi", "Mumbai", None, "Chennai"]  
}  
df = pd.DataFrame(data)  
  
average\_age = df['Age'].mean()  
df['Age'] = df['Age'].fillna(average\_age)  
  
df\_cleaned = df.dropna()  
print(df\_cleaned)  
 ***### \*\*Exercise 6: Adding and Removing Columns\*\*  
#1. Add a new column `"Salary"` with the following values: `[50000, 60000, 70000, 65000]`.  
#2. Remove the `"City"` column from the DataFrame.***df['Salary'] = [50000, 60000, 70000, 65000]  
df = df.drop(columns=['City'])  
print("New Data:")  
print(df)  
  
***### \*\*Exercise 7: Sorting Data\*\*  
#1. Sort the DataFrame by `"Age"` in ascending order.  
#2. Sort the DataFrame first by `"City"` and then by `"Age"` in descending order.***df["City"] =["Delhi", "Mumbai", None, "Chennai"]  
df\_sorted\_by\_age = df.sort\_values(by="Age", ascending=True)  
  
df\_sorted\_by\_city\_and\_age = df.sort\_values(by=["City", "Age"], ascending=[True, False])  
  
print("Sorted by Age:")  
print(df\_sorted\_by\_age)  
print("\nSorted by City and Age:")  
print(df\_sorted\_by\_city\_and\_age)  
  
***### \*\*Exercise 8: Grouping and Aggregation\*\*  
#1. Group the DataFrame by `"City"` and calculate the average `"Age"` for each city.  
#2. Group the DataFrame by `"City"` and `"Age"`, and count the number of occurrences for each group.***average\_age\_by\_city = df.groupby("City")["Age"].mean()  
  
count\_by\_city\_and\_age = df.groupby(["City", "Age"]).size().reset\_index(name='Count')  
  
print("Average Age by City:")  
print(average\_age\_by\_city)  
print("\nCount by City and Age:")  
print(count\_by\_city\_and\_age)  
 ***### \*\*Exercise 9: Merging DataFrames\*\*  
#1. Create two DataFrames:A  
# ```python  
# df1 = pd.DataFrame({  
# "Name": ["Amit", "Neha", "Raj"],  
# "Department": ["HR", "IT", "Finance"]  
# })  
  
# df2 = pd.DataFrame({  
# "Name": ["Neha", "Raj", "Priya"],  
# "Salary": [60000, 70000, 65000]  
# })  
# ```  
#2. Merge `df1` and `df2` on the `"Name"` column using an inner join.  
#3. Merge the same DataFrames using a left join.***import pandas as pd  
  
df1 = pd.DataFrame({  
 "Name": ["Amit", "Neha", "Raj"],  
 "Department": ["HR", "IT", "Finance"]  
})  
df2 = pd.DataFrame({  
 "Name": ["Neha", "Raj", "Priya"],  
 "Salary": [60000, 70000, 65000]  
})  
  
merged\_inner = pd.merge(df1, df2, on="Name", how="inner")  
  
merged\_left = pd.merge(df1, df2, on="Name", how="left")  
  
print("Inner Join data:")  
print(merged\_inner)  
print("\nLeft Join data:")  
print(merged\_left)