import pandas as pd  
  
*# Creating a new dataset*data = {  
 "Employee\_ID": [101, 102, 103, 104, 105, 106],  
 "Name": ["Rajesh", "Meena", "Suresh", "Anita", "Vijay", "Neeta"],  
 "Department": ["HR", "IT", "Finance", "IT", "Finance", "HR"],  
 "Age": [29, 35, 45, 32, 50, 28],  
 "Salary": [70000, 85000, 95000, 64000, 120000, 72000],  
 "City": ["Delhi", "Mumbai", "Bangalore", "Chennai", "Delhi", "Mumbai"]  
}  
  
df = pd.DataFrame(data)  
print(df)  
  
***#Exercise 1: Rename Columns  
#Rename the "Salary" column to "Annual Salary" and "City" to "Location".  
#Print the updated DataFrame.***df\_renamed = df.rename(columns={"Salary" : "Annual Salary","City":"Location"})  
print(df\_renamed)  
  
***#Exercise 2: Drop Columns  
#Drop the "Location" column from the DataFrame.  
#Print the DataFrame after dropping the column****.*df["Location"]=["Delhi", "Mumbai", "Bangalore", "Chennai", "Delhi", "Mumbai"]  
df\_dropped= df.drop(columns=["Location"])  
print(df\_dropped)  
  
***#Exercise 3: Drop Rows  
#Drop the row where "Name" is "Suresh".  
#Print the updated DataFrame.***df\_dropped\_row = df[df['Name'] != 'Suresh']  
print(df\_dropped\_row)  
 ***#Exercise 4: Handle Missing Data  
#Assign None to the "Salary" of "Meena".  
#Fill the missing "Salary" value with the mean salary of the existing employees.  
#Print the cleaned DataFrame.***df.loc[df['Name'] == 'Meena', 'Salary'] = None  
mean\_salary = df['Salary'].mean()  
df['Salary'] = df['Salary'].fillna(mean\_salary)  
print(df)  
  
***#Exercise 5: Create Conditional Columns  
#Create a new column "Seniority" that assigns "Senior" to employees aged 40 or above and "Junior" to employees younger than 40.  
#Print the updated DataFrame.***df['Seniority'] = df["Age"].apply(lambda x: 'Senior' if x>= 40 else 'Junior')  
print(df)  
  
***#Exercise 6: Grouping and Aggregation  
#Group the DataFrame by "Department" and calculate the average salary in each department.  
#Print the grouped DataFrame.***df\_grouped = df.groupby("Department")["Salary"].mean()  
print(df\_grouped)