```python
# Pandas HandsOn
# https://codeshare.io/0bp9Nm
# SaiPrabath Chowdary S

import pandas as pd

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)


# 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_dropcol = df_renamed.drop(columns=["Location"])
print(df_dropcol)


# Exercise 3: Drop Rows
# Drop the row where "Name" is "Suresh".
# Print the updated DataFrame.

df_droprow = df_dropcol.drop(df[df["Name"] == "Suresh"].index)
print(df_droprow)
```

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.

```python
df.loc[df["Name"] == "Meena", "Salary"] = None
mean_salary = df["Salary"].mean()
df.update(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.

```python
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.

```python
grouped_df = df.groupby("Department")["Salary"].mean()
print(grouped_df)
```