Task	Syntax	Description	Example
Load CSV data	`pd.read_csv('filename.csv')`	Read data from a CSV file into a Pandas DataFrame	`df_can=pd.read_csv('data.csv')`
Handling Missing Values	`df.dropna()`	Drop rows with missing values	`df_can.dropna()`
	`df.fillna(value)`	Fill missing values with a specified value	`df_can.fillna(0)`
Removing Duplicates	`df.drop_duplicates()`	Remove duplicate rows	`df_can.drop_duplicates()`
Renaming Columns	`df.rename(columns={'old_name': 'new_name'})`	Rename one or more columns	`df_can.rename(columns={'Age': 'Years'})`
Selecting Columns	`df['column_name']` or `df.column_name`	Select a single column	`df_can.Age` or `df_can['Age]'`
	`df[['col1', 'col2']]`	Select multiple columns	`df_can[['Name', 'Age']]`
Filtering Rows	`df[df['column'] > value]`	Filter rows based on a condition	`df_can[df_can['Age'] > 30]`
Applying Functions to Columns	`df['column'].apply(function_name)`	Apply a function to transform values in a column	`df_can['Age'].apply(lambda x: x + 1)`
Creating New Columns	`df['new_column'] = expression`	Create a new column with values derived from existing ones	`df_can['Total'] = df_can['Quantity'] * df_can['Price']`
Grouping and Aggregating	`df.groupby('column').agg({'col1': 'sum', 'col2': 'mean'})`	Group rows by a column and apply aggregate functions	`df_can.groupby('Category').agg({'Total': 'mean'})`
Sorting Rows	`df.sort_values('column', ascending=True/False)`	Sort rows based on a column	`df_can.sort_values('Date', ascending=True)`
Displaying First n Rows	`df.head(n)`	Show the first n rows of the DataFrame	`df_can.head(3)`
Displaying Last n Rows	`df.tail(n)`	Show the last n rows of the DataFrame	`df_can.tail(3)`
Checking for Null Values	`df.isnull()`	Check for null values in the DataFrame	`df_can.isnull()`
Selecting Rows by Index	`df.iloc[index]`	Select rows based on integer index	`df_can.iloc[3]`
	`df.iloc[start:end]`	Select rows in a specified range	`df_can.iloc[2:5]`
Selecting Rows by Label	`df.loc[label]`	Select rows based on label/index name	`df_can.loc['Label']`
	`df.loc[start:end]`	Select rows in a specified label/index range	`df_can.loc['Age':'Quantity']`
Summary Statistics	`df.describe()`	Generates descriptive statistics for numerical columns	`df_can.describe()`