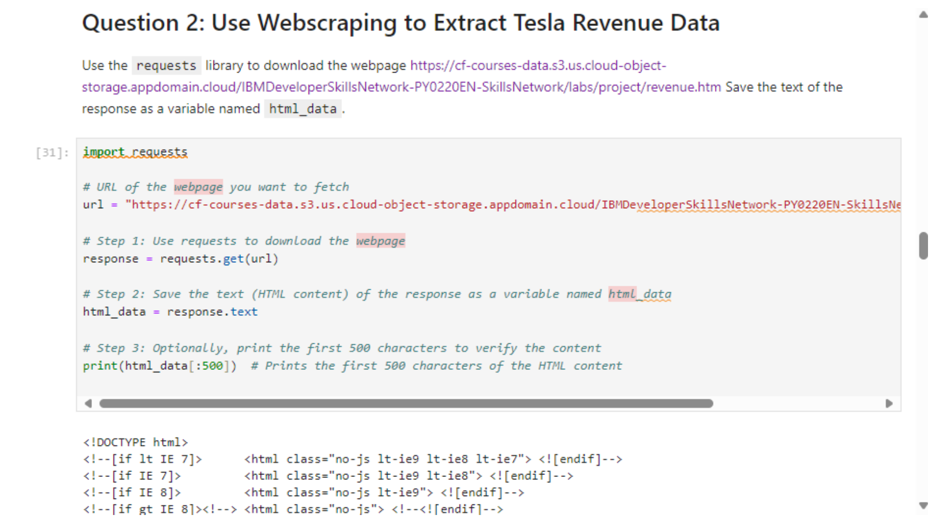


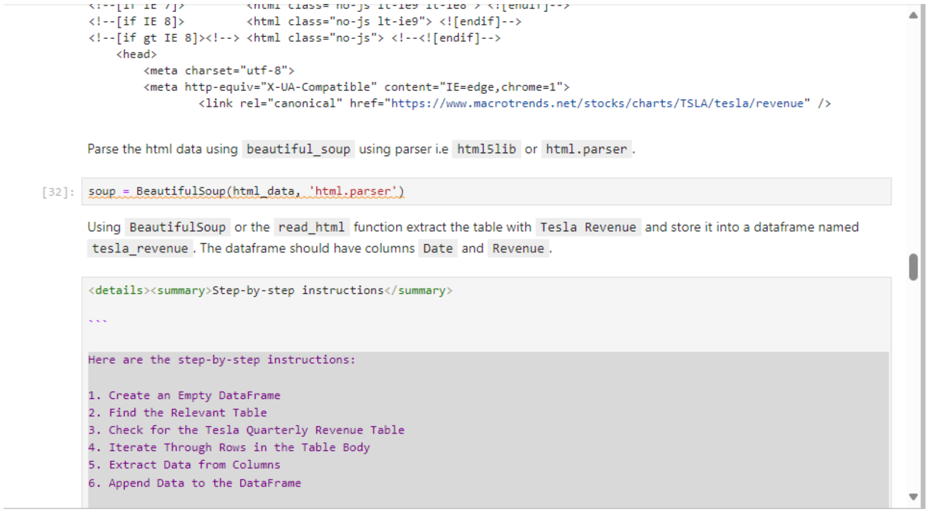
Question 2: Use Webscraping to Extract Tesla Revenue Data

Use the `requests` library to download the webpage https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/revenue.htm Save the text of the response as a variable named `html_data`.

[31]:

```
import requests

# URL of the webpage you want to fetch
url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNe

# Step 1: Use requests to download the webpage
response = requests.get(url)

# Step 2: Save the text (HTML content) of the response as a variable named html_data
html_data = response.text

# Step 3: Optionally, print the first 500 characters to verify the content
print(html_data[:500])  # Prints the first 500 characters of the HTML content
```

```
<!DOCTYPE html>
<!--[if lt IE 7]>      <html class="no-js lt-ie9 lt-ie8 lt-ie7"> <![endif]-->
<!--[if IE 7]>         <html class="no-js lt-ie9 lt-ie8"> <![endif]-->
<!--[if IE 8]>         <html class="no-js lt-ie9"> <![endif]-->
<!--[if gt IE 8]><!--> <html class="no-js"> <!--<![endif]-->
```



```
<!--[if IE 7]>         <html class="no-js lt-ie9 lt-ie8"> <![endif]-->
<!--[if IE 8]>         <html class="no-js lt-ie9"> <![endif]-->
<!--[if gt IE 8]><!--> <html class="no-js"> <!--<![endif]-->
```

```
    <head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
                <link rel="canonical" href="https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue" />
```

Parse the html data using `beautiful_soup` using parser i.e `html5lib` or `html.parser`.

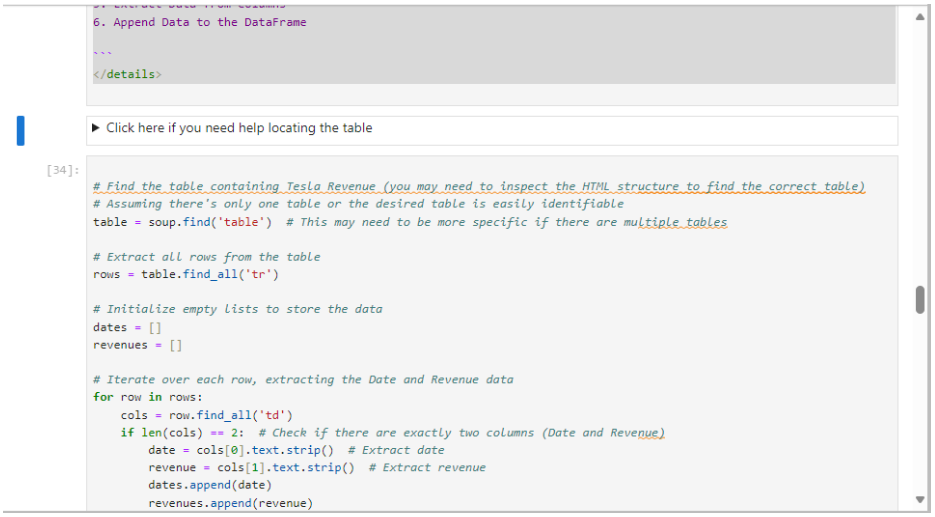
[32]:

```
soup = BeautifulSoup(html_data, 'html.parser')
```

Using `BeautifulSoup` or the `read_html` function extract the table with `Tesla Revenue` and store it into a dataframe named `tesla_revenue`. The dataframe should have columns `Date` and `Revenue`.

```
<details><summary>Step-by-step instructions</summary>

```

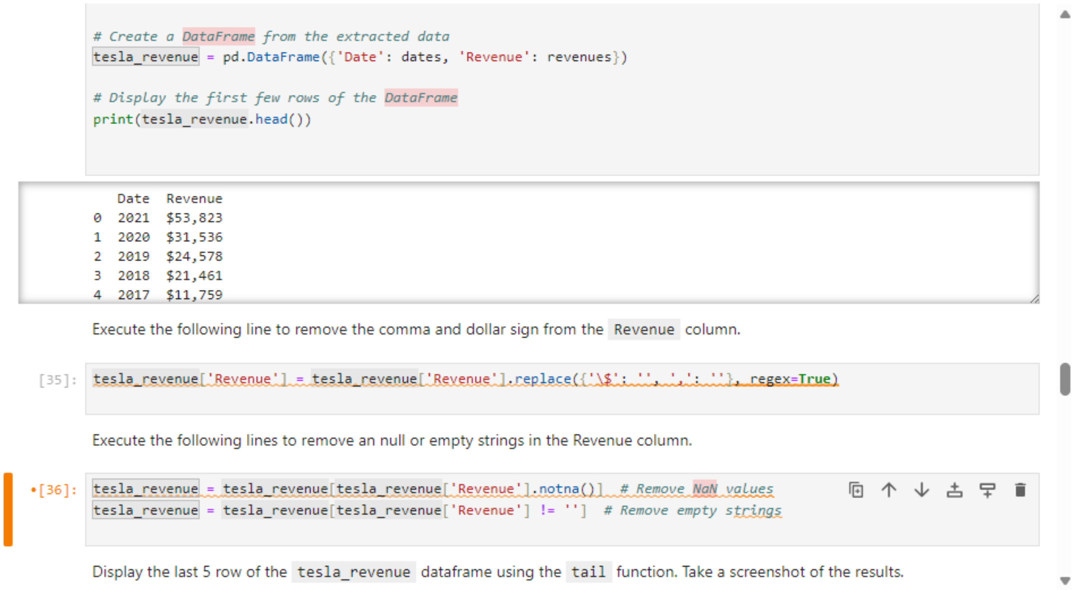
Here are the step-by-step instructions:

1. Create an Empty DataFrame
2. Find the Relevant Table
3. Check for the Tesla Quarterly Revenue Table
4. Iterate Through Rows in the Table Body
5. Extract Data from Columns
6. Append Data to the DataFrame
```




```
6. Append Data to the DataFrame
```


```
```
</details>
```

▶ Click here if you need help locating the table

[34]:

```
# Find the table containing Tesla Revenue (you may need to inspect the HTML structure to find the correct table)
# Assuming there's only one table or the desired table is easily identifiable
table = soup.find('table')  # This may need to be more specific if there are multiple tables

# Extract all rows from the table
rows = table.find_all('tr')

# Initialize empty lists to store the data
dates = []
revenues = []

# Iterate over each row, extracting the Date and Revenue data
for row in rows:
    cols = row.find_all('td')
    if len(cols) == 2:  # Check if there are exactly two columns (Date and Revenue)
        date = cols[0].text.strip()  # Extract date
        revenue = cols[1].text.strip()  # Extract revenue
        dates.append(date)
        revenues.append(revenue)
```



```
# Create a DataFrame from the extracted data
tesla_revenue = pd.DataFrame({'Date': dates, 'Revenue': revenues})

# Display the first few rows of the DataFrame
print(tesla_revenue.head())
```

```
   Date  Revenue
0  2021  $53,823
1  2020  $31,536
2  2019  $24,578
3  2018  $21,461
4  2017  $11,759
```

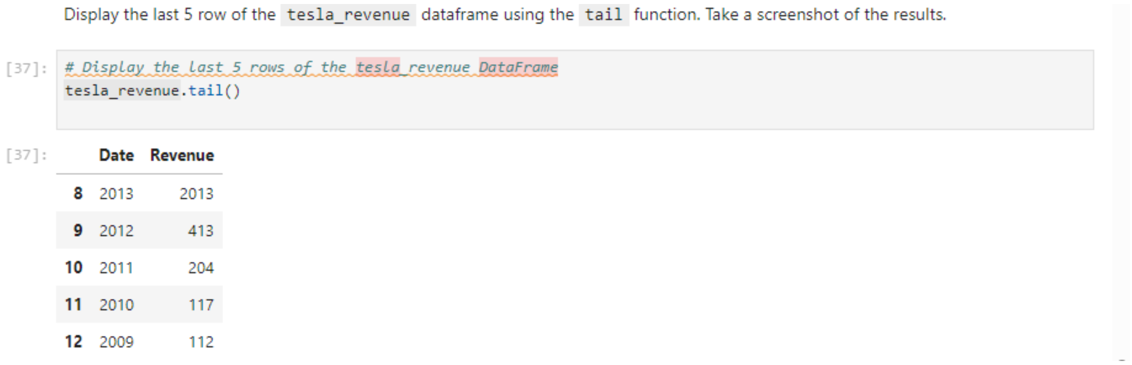
Execute the following line to remove the comma and dollar sign from the `Revenue` column.

[35]:
```
tesla_revenue['Revenue'] = tesla_revenue['Revenue'].replace({'\$': '', ',': ''}, regex=True)
```

Execute the following lines to remove an null or empty strings in the Revenue column.

•[36]:
```
tesla_revenue = tesla_revenue[tesla_revenue['Revenue'].notna()]  # Remove NaN values
tesla_revenue = tesla_revenue[tesla_revenue['Revenue'] != '']  # Remove empty strings
```

Display the last 5 row of the `tesla_revenue` dataframe using the `tail` function. Take a screenshot of the results.



Display the last 5 row of the `tesla_revenue` dataframe using the `tail` function. Take a screenshot of the results.

[37]:
```
# Display the last 5 rows of the tesla_revenue DataFrame
tesla_revenue.tail()
```

[37]:

	Date	Revenue
8	2013	2013
9	2012	413
10	2011	204
11	2010	117
12	2009	112