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`.

[4]:
```
import requests
```

[31]:
```
url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNet
response = requests.get(url)
html_data = response.text
```

Parse the html data using `beautiful_soup` using parser i.e `html5lib` or `html.parser`. Make sure to use the `html_data` with the content parameter as follow `html_data.content`.

[21]:
```
soup = BeautifulSoup(response.content, '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`.

[33]:
```
# Find the table containing Tesla revenue (assuming it's the second table)
tables = soup.find_all('table')

# Extract the Tesla revenue table (adjust index if necessary)
tesla_table = tables[1]

# Extract table rows
rows = tesla_table.find_all('tr')

# Create a list to hold the extracted data
data = []

# Loop through the rows and extract the text for each cell (Date and Revenue)
for row in rows[1:]:  # Skipping the header row
    cols = row.find_all('td')
    date = cols[0].text.strip()
    revenue = cols[1].text.strip()
    data.append([date, revenue])

# Convert the list into a pandas DataFrame
tesla_revenue = pd.DataFrame(data, columns=['Date', 'Revenue'])

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

[34]:
```
tesla_revenue["Revenue"] = tesla_revenue['Revenue'].str.replace(',|\$',"", regex=True)
```

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

[35]:
```
tesla_revenue.dropna(inplace=True)

tesla_revenue = tesla_revenue[tesla_revenue['Revenue'] != ""]
```

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

[36]:
```
#rows.append({'Date': date, 'Revenue': revenue})
#tesla_revenue = pd.DataFrame(rows)
print(tesla_revenue.tail())
```

```
          Date Revenue
48  2010-09-30      31
49  2010-06-30      28
50  2010-03-31      21
52  2009-09-30      46
53  2009-06-30      27
```