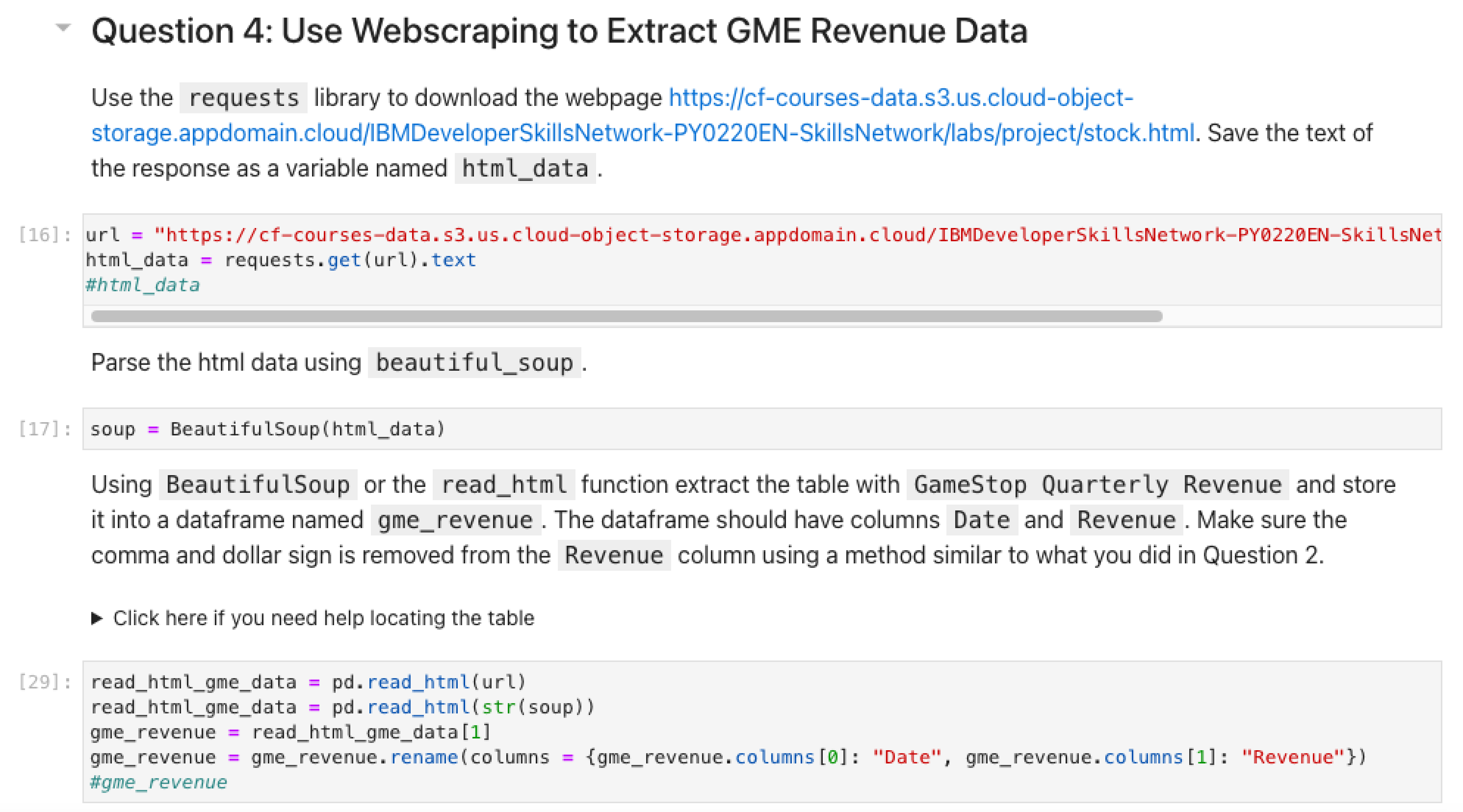


Question 4: Use Webscraping to Extract GME Revenue Data

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

[16]:
```
url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNet
html_data = requests.get(url).text
#html_data
```

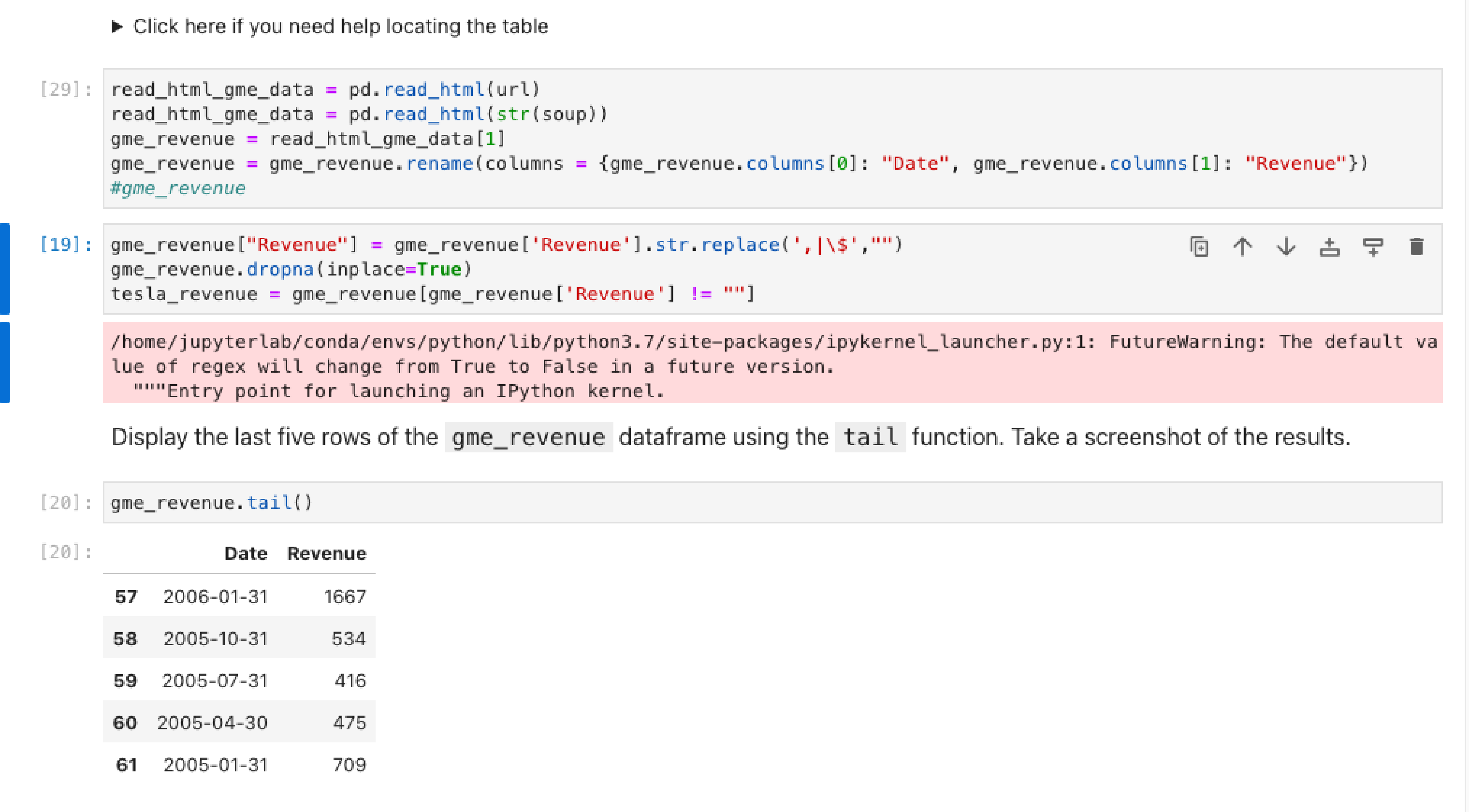
Parse the html data using `beautiful_soup`.

[17]:
```
soup = BeautifulSoup(html_data)
```

Using `BeautifulSoup` or the `read_html` function extract the table with `GameStop Quarterly Revenue` and store it into a dataframe named `gme_revenue`. The dataframe should have columns `Date` and `Revenue`. Make sure the comma and dollar sign is removed from the `Revenue` column using a method similar to what you did in Question 2.

▶ Click here if you need help locating the table

[29]:
```
read_html_gme_data = pd.read_html(url)
read_html_gme_data = pd.read_html(str(soup))
gme_revenue = read_html_gme_data[1]
gme_revenue = gme_revenue.rename(columns = {gme_revenue.columns[0]: "Date", gme_revenue.columns[1]: "Revenue"})
#gme_revenue
```



▶ Click here if you need help locating the table

[29]:

```
read_html_gme_data = pd.read_html(url)
read_html_gme_data = pd.read_html(str(soup))
gme_revenue = read_html_gme_data[1]
gme_revenue = gme_revenue.rename(columns = {gme_revenue.columns[0]: "Date", gme_revenue.columns[1]: "Revenue"})
#gme_revenue
```

[19]:

```
gme_revenue["Revenue"] = gme_revenue['Revenue'].str.replace(',|\$',"")
gme_revenue.dropna(inplace=True)
tesla_revenue = gme_revenue[gme_revenue['Revenue'] != ""]
```

```
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages/ipykernel_launcher.py:1: FutureWarning: The default va
lue of regex will change from True to False in a future version.
  """Entry point for launching an IPython kernel.
```

Display the last five rows of the `gme_revenue` dataframe using the `tail` function. Take a screenshot of the results.

[20]:

```
gme_revenue.tail()
```

[20]:

	Date	Revenue
57	2006-01-31	1667
58	2005-10-31	534
59	2005-07-31	416
60	2005-04-30	475
61	2005-01-31	709