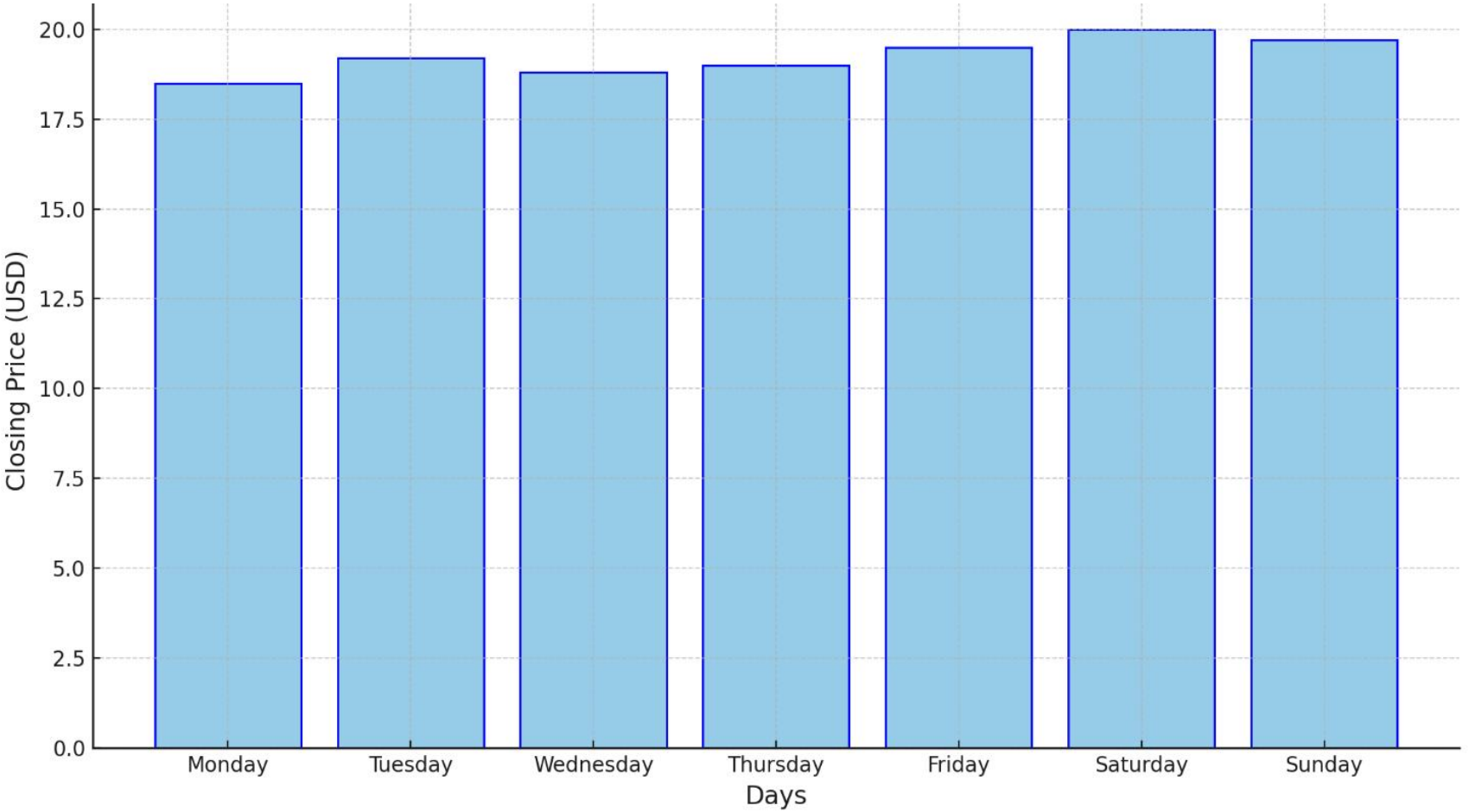

20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
Closing Price (USD)
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Days

```
[**********************100%**************************]  1 of 1 completed
```

Price	Date	Adj Close	Close	High	Low	Open
Ticker		TSLA	TSLA	TSLA	TSLA	TSLA
0	2020-01-02	28.684000	28.684000	28.713333	28.114000	28.299999
1	2020-01-03	29.534000	29.534000	30.266666	29.128000	29.366667
2	2020-01-06	30.102667	30.102667	30.104000	29.333332	29.364668
3	2020-01-07	31.270666	31.270666	31.441999	30.224001	30.760000
4	2020-01-08	32.809334	32.809334	33.232666	31.215334	31.580000

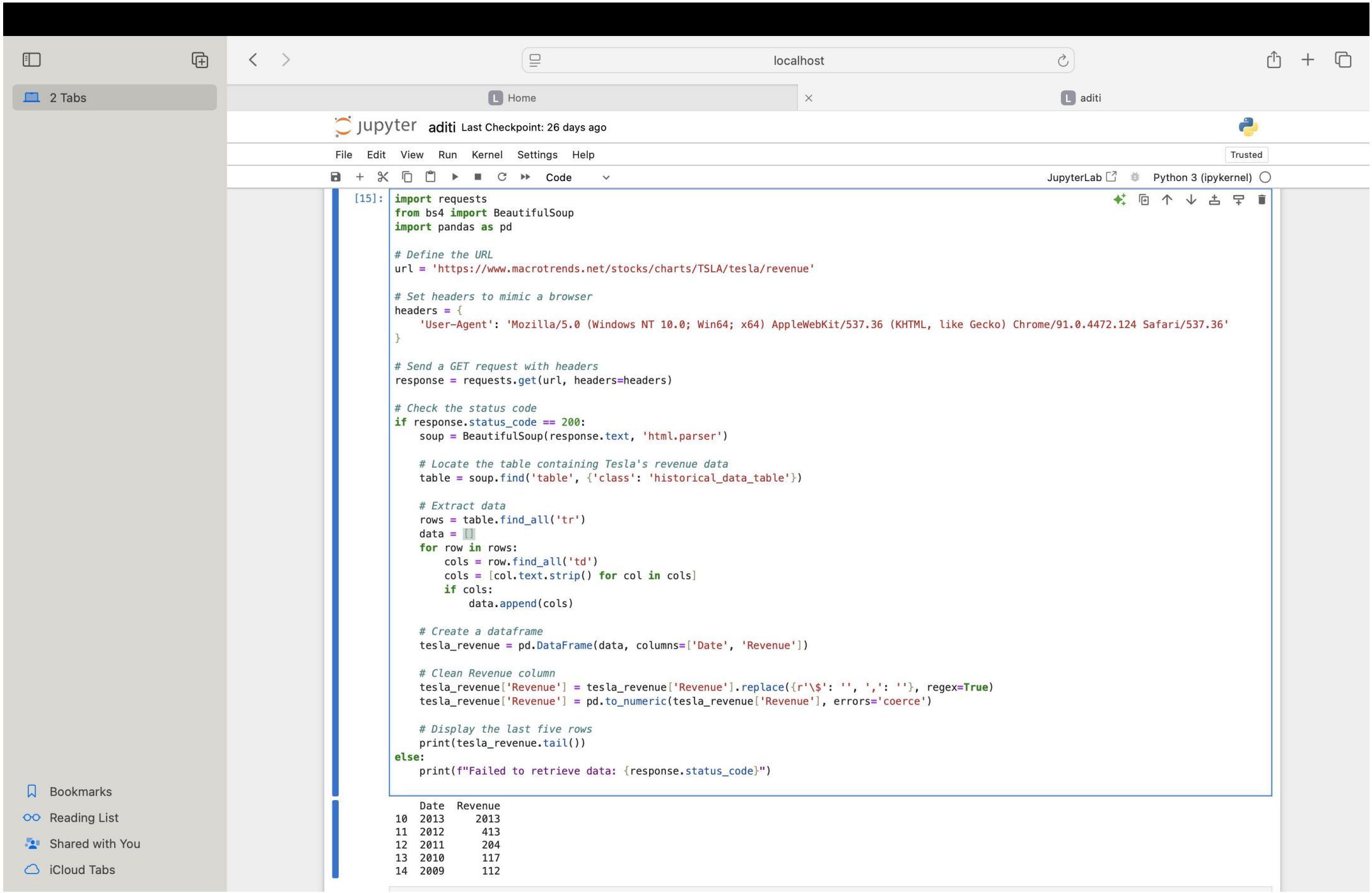
Price	Volume
Ticker	TSLA
0	142981500
1	266677500
2	151995000
3	268231500
4	467164500

```python
import yfinance as yf

# Extract GameStop stock data
gme_data = yf.download('GME', start='2020-01-01', end='2023-01-01')

# Reset the index
gme_data.reset_index(inplace=True)

# Save the dataframe to a CSV file
gme_data.to_csv('gme_stock_data.csv', index=False)

# Display the first five rows of the dataframe
print(gme_data.head())
```

```
[**********100%**********]  1 of 1 completed
Price        Date Adj Close   Close    High     Low    Open    Volume
Ticker                   GME     GME     GME     GME     GME       GME
0      2020-01-02    1.5775  1.5775  1.6175  1.5175  1.5350  17814400
1      2020-01-03    1.4700  1.4700  1.5625  1.4600  1.5525  14175600
2      2020-01-06    1.4625  1.4625  1.4775  1.4000  1.4500  13579200
3      2020-01-07    1.3800  1.3800  1.4575  1.3600  1.4425  20912000
4      2020-01-08    1.4300  1.4300  1.4625  1.3525  1.3725  22517600
```

[15]:
```python
import requests
from bs4 import BeautifulSoup
import pandas as pd

# Define the URL
url = 'https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue'

# Set headers to mimic a browser
headers = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36'
}

# Send a GET request with headers
response = requests.get(url, headers=headers)

# Check the status code
if response.status_code == 200:
    soup = BeautifulSoup(response.text, 'html.parser')

    # Locate the table containing Tesla's revenue data
    table = soup.find('table', {'class': 'historical_data_table'})

    # Extract data
    rows = table.find_all('tr')
    data = []
    for row in rows:
        cols = row.find_all('td')
        cols = [col.text.strip() for col in cols]
        if cols:
            data.append(cols)

    # Create a dataframe
    tesla_revenue = pd.DataFrame(data, columns=['Date', 'Revenue'])

    # Clean Revenue column
    tesla_revenue['Revenue'] = tesla_revenue['Revenue'].replace({r'\$': '', ',': ''}, regex=True)
    tesla_revenue['Revenue'] = pd.to_numeric(tesla_revenue['Revenue'], errors='coerce')

    # Display the last five rows
    print(tesla_revenue.tail())
else:
    print(f"Failed to retrieve data: {response.status_code}")
```

```
    Date  Revenue
10  2013     2013
11  2012      413
12  2011      204
13  2010      117
14  2009      112
```


2 Tabs
localhost
Home
aditi
jupyter aditi Last Checkpoint: 26 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[15]:
import requests
from bs4 import BeautifulSoup
import pandas as pd
# Define the URL
url = 'https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue'
# Set headers to mimic a browser
headers = {
'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36'
}
# Send a GET request with headers
response = requests.get(url, headers=headers)
# Check the status code
if response.status_code == 200:
soup = BeautifulSoup(response.text, 'html.parser')
# Locate the table containing Tesla's revenue data
table = soup.find('table', {'class': 'historical_data_table'})
# Extract data
rows = table.find_all('tr')
data = []
for row in rows:
cols = row.find_all('td')
cols = [col.text.strip() for col in cols]
if cols:
data.append(cols)
# Create a dataframe
tesla_revenue = pd.DataFrame(data, columns=['Date', 'Revenue'])
# Clean Revenue column
tesla_revenue['Revenue'] = tesla_revenue['Revenue'].replace({r'\$': '', ',': ''}, regex=True)
tesla_revenue['Revenue'] = pd.to_numeric(tesla_revenue['Revenue'], errors='coerce')
# Display the last five rows
print(tesla_revenue.tail())
else:
print(f"Failed to retrieve data: {response.status_code}")
Date Revenue
10 2013 2013
11 2012 413
12 2011 204
13 2010 117
14 2009 112
Bookmarks
Reading List
Shared with You
iCloud Tabs