Question 1: Use yfinance to Extract Stock Data

In [26]:
```
import yfinance as yf
import pandas as pd
import requests
from bs4 import BeautifulSoup
import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

Define the ticker symbol for Tesla

In [9]:
```
ticker = "TSLA"
```

Use yfinance to get the stock data

In [10]:
```
tesla_stock = yf.Ticker(ticker)
```

Get the historical stock data for Tesla

In [11]:
```
tesla_history = tesla_stock.history(period="max")
```

Reset the index using the reset_index(inplace=True) function on the tesla_data DataFrame and display the first five rows of the tesla_data dataframe using the head function. Take a screenshot of the results and code from the beginning of Question 1 to the results below.

In [13]:
```
tesla_history.reset_index(inplace=True)
tesla_history.head()
```

Out[13]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2010-06-29 00:00:00-04:00	1.266667	1.666667	1.169333	1.592667	281494500	0.0	0.0
1	2010-06-30 00:00:00-04:00	1.719333	2.028000	1.553333	1.588667	257806500	0.0	0.0
2	2010-07-01 00:00:00-04:00	1.666667	1.728000	1.351333	1.464000	123282000	0.0	0.0
3	2010-07-02 00:00:00-04:00	1.533333	1.540000	1.247333	1.280000	77097000	0.0	0.0
4	2010-07-06 00:00:00-04:00	1.333333	1.333333	1.055333	1.074000	103003500	0.0	0.0

Question 2: Use Webscraping to Extract Tesla Revenue Data

```
Make a GET request to the Tesla revenue webpage
```

In [19]:
```
url = 'https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue'
```

Send a GET request to the URL and get the page content

In [22]:
```
response = requests.get(url)
```

```
Create a BeautifulSoup object to parse the HTML content
```

In [21]:
```
soup = BeautifulSoup(response.text, 'html.parser')
```

```
Find the revenue table on the page using its HTML class
```

In [23]:
```
revenue_table = soup.find('table', class_='historical_data_table')
```

Extract the revenue data from the table rows

In [33]:
```
rows = revenue_table.find_all('tr')
```

Print the last five rows of the revenue table

In [32]:
```
for row in rows[-5:]:
    print(row.get_text())
```

```
2013
$2,013


2012
$413


2011
$204


2010
$117


2009
$112
```

Question 3: Use yfinance to Extract Stock Data

Create a ticker object for GameStop with the ticker symbol "GME"

In [36]:
```
gamestop = yf.Ticker("GME")
```

Use the history function to extract stock information and save it in the gme_data DataFrame

In [37]:
```
gme_data = gamestop.history(period="max")
```

Reset the index of the gme_data DataFrame in place

In [38]:
```
gme_data.reset_index(inplace=True)
```

Display the first five rows of the gme_data DataFrame using the head function

In [39]:
```
gme_data.head()
```

Out[39]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2002-02-13 00:00:00-05:00	1.620128	1.693350	1.603296	1.691667	76216000	0.0	0.0
1	2002-02-14 00:00:00-05:00	1.712708	1.716074	1.670626	1.683251	11021600	0.0	0.0
2	2002-02-15 00:00:00-05:00	1.683250	1.687458	1.658002	1.674834	8389600	0.0	0.0
3	2002-02-19 00:00:00-05:00	1.666418	1.666418	1.578047	1.607504	7410400	0.0	0.0
4	2002-02-20 00:00:00-05:00	1.615920	1.662209	1.603296	1.662209	6892800	0.0	0.0

Question 4: Use Webscraping to Extract GME Revenue Data

Download the webpage and save the response text as html_data

In [41]:
```
url = "https://www.macrotrends.net/stocks/charts/GME/gamestop/revenue"
html_data = requests.get(url).text
```

Parse the html data using BeautifulSoup

In [42]:
```
soup = BeautifulSoup(html_data, "html5lib")
```

Extract the table with GameStop Quarterly Revenue and store it in the gme_revenue DataFrame

In [46]:
```
gme_revenue = pd.read_html(url, match="GameStop Quarterly Revenue", flavor='bs4')[0]
gme_revenue = gme_revenue.rename(columns={"GameStop Quarterly Revenue (Millions of US $
gme_revenue["Revenue"] = gme_revenue["Revenue"].str.replace(",", "", regex=False).str.r
gme_revenue = gme_revenue.reset_index(drop=True)
```

Display the last five rows of the gme_revenue DataFrame using the tail function

In [47]:
```
gme_revenue.tail()
```

Out[47]:

	Date	Revenue
52	2010-01-31	3524
53	2009-10-31	1835
54	2009-07-31	1739
55	2009-04-30	1981
56	2009-01-31	3492

Question 5: Plot Tesla Stock Graph

Plot Tesla stock graph

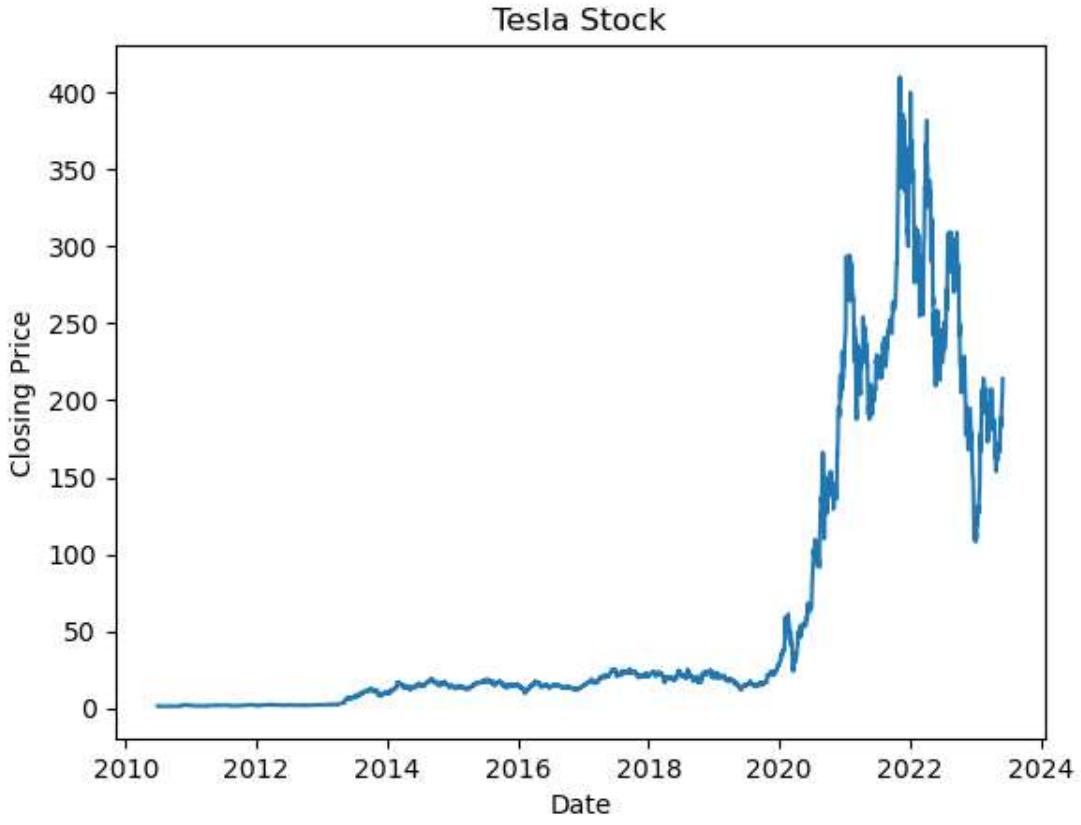
In [54]:
```
import yfinance as yf
import matplotlib.pyplot as plt

# Create a Ticker object for Tesla with the ticker symbol "TSLA"
tesla = yf.Ticker("TSLA")

# Extract the historical stock data for Tesla
tesla_data = tesla.history(period="max")

# Define the make_graph function
def make_graph(x, y, title, x_label, y_label):
    plt.plot(x, y)
    plt.title(title)
    plt.xlabel(x_label)
    plt.ylabel(y_label)
    plt.show()

# Plot the Tesla stock graph
make_graph(tesla_data.index, tesla_data['Close'], 'Tesla Stock', 'Date', 'Closing Price
```



Question 6: Plot GameStop Stock Graph

In [55]:
```
import yfinance as yf
import matplotlib.pyplot as plt

# Create a Ticker object for GameStop with the ticker symbol "GME"
gamestop = yf.Ticker("GME")

# Extract the historical stock data for GameStop
gme_data = gamestop.history(period="max")

# Define the make_graph function
def make_graph(x, y, title, x_label, y_label):
    plt.plot(x, y)
    plt.title(title)
    plt.xlabel(x_label)
    plt.ylabel(y_label)
    plt.show()

# Plot the GameStop stock graph
make_graph(gme_data.index, gme_data['Close'], 'GameStop Stock', 'Date', 'Closing Price'
```