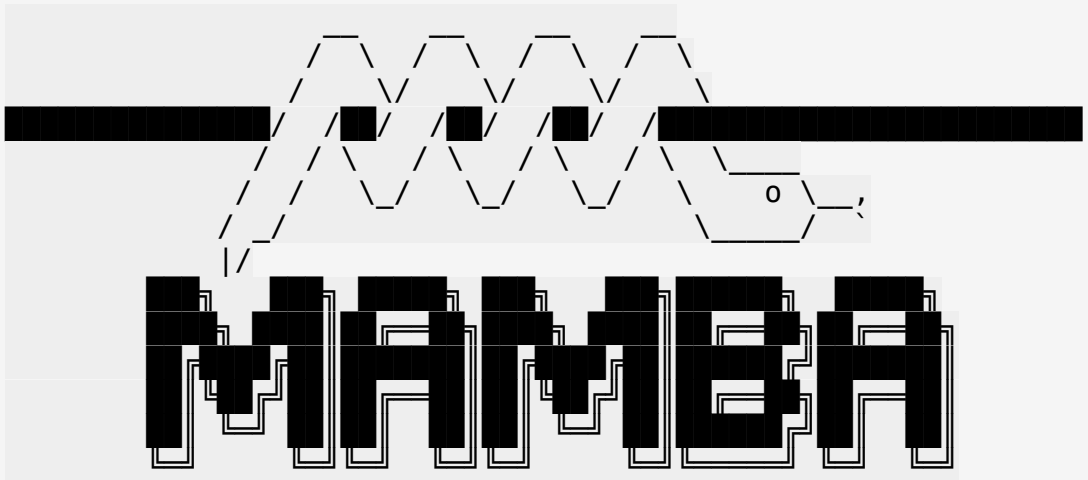
Not all stock data is available via API in this assignment; you will use web-scraping to obtain financial data. You will be quizzed on your results. Using beautiful soup we will extract historical share data from a web-page.

```
#!pip install pandas==1.3.3
#!pip install requests==2.26.0
!mamba install bs4==4.10.0 -y
!mamba install html5lib==1.1 -y
!pip install lxml==4.6.4
#!pip install plotly==5.3.1
```



```
        mamba (0.15.3) supported by @QuantStack

        GitHub:  https://github.com/mamba-org/mamba
        Twitter: https://twitter.com/QuantStack

Looking for: ['bs4==4.10.0']

pkgs/main/noarch          [>                   ] (--:--) No change
pkgs/main/noarch          [====================] (00m:00s) No change
pkgs/main/linux-64        [>                   ] (--:--) No change
pkgs/main/linux-64        [====================] (00m:00s) No change
pkgs/r/noarch             [>                   ] (--:--) No change
pkgs/r/noarch             [====================] (00m:00s) No change
pkgs/r/linux-64           [>                   ] (--:--) No change
pkgs/r/linux-64           [====================] (00m:00s) No change

Pinned packages:
  - python 3.7.*
```

```
Transaction

  Prefix: /home/jupyterlab/conda/envs/python

  All requested packages already installed


                   __    __    __    __
                  /  \  /  \  /  \  /  \
                 /    \/    \/    \/    \
███████████████/  /██/  /██/  /██/  /████████████████████
              /  / \   / \   / \   / \  \____
             /  /   \_/   \_/   \_/   \    o \__,
            / _/                       \_____/`
            |/
         ███╗   ███╗ █████╗ ███╗   ███╗██████╗  █████╗
         ████╗ ████║██╔══██╗████╗ ████║██╔══██╗██╔══██╗
         ██╔████╔██║███████║██╔████╔██║██████╔╝███████║
         ██║╚██╔╝██║██╔══██║██║╚██╔╝██║██╔══██╗██╔══██║
         ██║ ╚═╝ ██║██║  ██║██║ ╚═╝ ██║██████╔╝██║  ██║
         ╚═╝     ╚═╝╚═╝  ╚═╝╚═╝     ╚═╝╚═════╝ ╚═╝  ╚═╝

         mamba (0.15.3) supported by @QuantStack

         GitHub:  https://github.com/mamba-org/mamba
         Twitter: https://twitter.com/QuantStack

███████████████████████████████████████████████████████


Looking for: ['html5lib==1.1']

pkgs/main/linux-64       Using cache
pkgs/main/noarch         Using cache
pkgs/r/linux-64          Using cache
pkgs/r/noarch            Using cache

Pinned packages:
  - python 3.7.*


Transaction

  Prefix: /home/jupyterlab/conda/envs/python

  All requested packages already installed

Requirement already satisfied: lxml==4.6.4 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (4.6.4)
```

```
import pandas as pd
import requests
from bs4 import BeautifulSoup
```

Using Webscraping to Extract Stock Data Example

First we must use the `request` library to downlaod the webpage, and extract the text. We will extract Netflix stock data https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/netflix_data_webpage.html.

```
url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/netflix_data_webpage.html"

data  = requests.get(url).text
```

Next we must parse the text into html using `beautiful_soup`

```
soup = BeautifulSoup(data, 'html5lib')
```

Now we can turn the html table into a pandas dataframe

```
netflix_data = pd.DataFrame(columns=["Date", "Open", "High", "Low", "Close", "Volume"])

# First we isolate the body of the table which contains all the information
# Then we loop through each row and find all the column values for each row
for row in soup.find("tbody").find_all('tr'):
    col = row.find_all("td")
    date = col[0].text
    Open = col[1].text
    high = col[2].text
    low = col[3].text
    close = col[4].text
    adj_close = col[5].text
    volume = col[6].text

    # Finally we append the data of each row to the table
    netflix_data = netflix_data.append({"Date":date, "Open":Open, "High":high, "Low":low, "Close":close, "Adj Close":adj_close, "Volume":volume}, ignore_index=True)
```

We can now print out the dataframe

```
netflix_data.head()
```

	Date	Open	High	Low	Close	Volume	Adj Close
0	Jun 01, 2021	504.01	536.13	482.14	528.21	78,560,600	528.21
1	May 01, 2021	512.65	518.95	478.54	502.81	66,927,600	502.81
2	Apr 01, 2021	529.93	563.56	499.00	513.47	111,573,300	513.47
3	Mar 01, 2021	545.57	556.99	492.85	521.66	90,183,900	521.66
4	Feb 01, 2021	536.79	566.65	518.28	538.85	61,902,300	538.85

We can also use the pandas `read_html` function using the url

```
read_html_pandas_data = pd.read_html(url)
```

Or we can convert the BeautifulSoup object to a string

```
read_html_pandas_data = pd.read_html(str(soup))
```

Beacause there is only one table on the page, we just take the first table in the list returned

```
netflix_dataframe = read_html_pandas_data[0]

netflix_dataframe.head()
```

	Date	Open	High	Low	Close*	Adj Close**	Volume
0	Jun 01, 2021	504.01	536.13	482.14	528.21	528.21	78560600
1	May 01, 2021	512.65	518.95	478.54	502.81	502.81	66927600
2	Apr 01, 2021	529.93	563.56	499.00	513.47	513.47	111573300
3	Mar 01, 2021	545.57	556.99	492.85	521.66	521.66	90183900
4	Feb 01, 2021	536.79	566.65	518.28	538.85	538.85	61902300

Using Webscraping to Extract Stock Data Exercise

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

```
url = " https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/amazon_data_webpage.html"
html_data = requests.get(url).text
```

Parse the html data using `beautiful_soup`.

```
soup = BeautifulSoup(html_data, 'html5lib')
```

Question 1 What is the content of the title attribute:

```
soup.title
```

```
<title>Amazon.com, Inc. (AMZN) Stock Historical Prices &amp;amp; Data - Yahoo Finance</title>
```

Using beautiful soup extract the table with historical share prices and store it into a dataframe named `amazon_data`. The dataframe should have columns Date, Open, High, Low, Close, Adj Close, and Volume. Fill in each variable with the correct data from the list `col`.

```
amazon_data = pd.DataFrame(columns=["Date", "Open", "High", "Low", "Close", "Volume"])

for row in soup.find("tbody").find_all("tr"):
    col = row.find_all("td")
    date =col[0].text
    Open =col[1].text
    high =col[2].text
    low =col[3].text
    close =col[4].text
    adj_close =col[5].text
    volume =col[6].text

    amazon_data = amazon_data.append({"Date":date, "Open":Open, "High":high, "Low":low, "Close":close, "Adj Close":adj_close, "Volume":volume}, ignore_index=True)
```

Print out the first five rows of the `amazon_data` dataframe you created.

```
amazon_data.head()
```

	Date	Open	High	Low	Close	Volume	Adj Close
0	Jan 01, 2021	3,270.00	3,363.89	3,086.00	3,206.20	71,528,900	3,206.20
1	Dec 01, 2020	3,188.50	3,350.65	3,072.82	3,256.93	77,556,200	3,256.93
2	Nov 01, 2020	3,061.74	3,366.80	2,950.12	3,168.04	90,810,500	3,168.04
3	Oct 01, 2020	3,208.00	3,496.24	3,019.00	3,036.15	116,226,100	3,036.15
4	Sep 01, 2020	3,489.58	3,552.25	2,871.00	3,148.73	115,899,300	3,148.73

Question 2 What is the name of the columns of the dataframe

```
amazon_data.columns
```

```
Index(['Date', 'Open', 'High', 'Low', 'Close', 'Volume', 'Adj Close'], dtype='object')
```

Question 3 What is the `Open` of the last row of the amazon_data dataframe?

```
amazon_data.iloc[-1, 1]

# or amazon_data['Open'].tail(1)

# or amazon_data.iloc[-1]['Open']
```

```
'656.29'
```

Question 4 What is the Open of Jun 01, 2019 of the dataframe?

```
amazon_data.loc[amazon_data["Date"]=="Jun 01, 2019"]
```

```
            Date      Open      High       Low     Close      Volume
Adj Close
19  Jun 01, 2019  1,760.01  1,935.20  1,672.00  1,893.63  74,746,500
1,893.63
```

Joseph Santarcangelo has a PhD in Electrical Engineering, his research focused on using machine learning, signal processing, and computer vision to determine how videos impact human cognition. Joseph has been working for IBM since he completed his PhD.

Azim Hirjani

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2021-06-09	1.2	Lakshmi Holla	Added URL in question 3
2020-11-10	1.1	Malika Singla	Deleted the Optional part
2020-08-27	1.0	Malika Singla	Added lab to GitLab