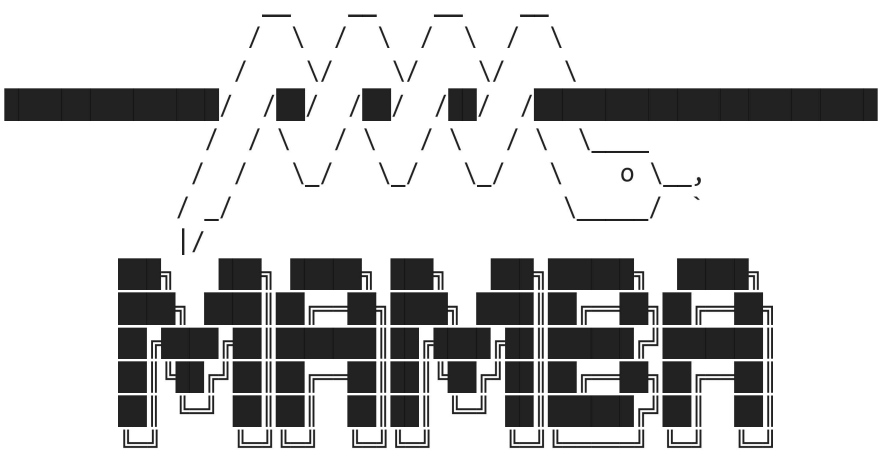
Extracting Stock Data Using a Web Scraping

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.

Table of Contents

Estimated Time Needed: **30 min**

In [1]:
```
#!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
```



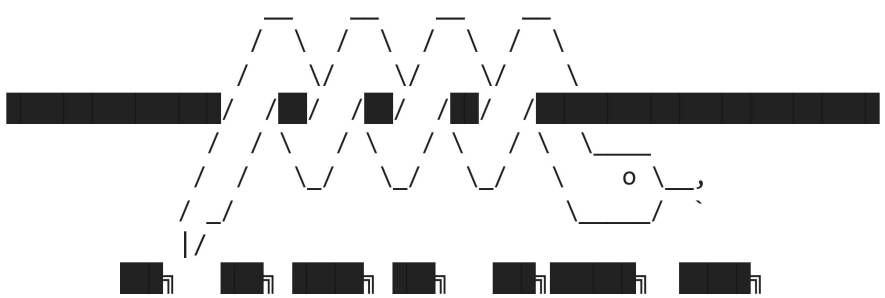
```
mamba (0.15.3) supported by @QuantStack

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

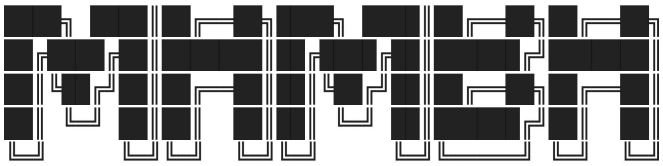
```
Looking for: ['bs4==4.10.0']

pkgs/main/noarch         [<=>                ] (00m:00s)
pkgs/main/noarch         [=>                 ] (00m:00s) 280 KB / ?? (920.47 KB/s)
pkgs/main/noarch         [=>                 ] (00m:00s) 280 KB / ?? (920.47 KB/s)
pkgs/r/linux-64          [<=>                ] (00m:00s)
pkgs/main/noarch         [=>                 ] (00m:00s) 280 KB / ?? (920.47 KB/s)
pkgs/r/linux-64          [=>                 ] (00m:00s) 264 KB / ?? (866.93 KB/s)
pkgs/main/noarch         [=>                 ] (00m:00s) 280 KB / ?? (920.47 KB/s)
pkgs/r/linux-64          [=>                 ] (00m:00s) 264 KB / ?? (866.93 KB/s)
pkgs/r/noarch            [<=>                ] (00m:00s)
pkgs/main/noarch         [=>                 ] (00m:00s) 280 KB / ?? (920.47 KB/s)
pkgs/r/linux-64          [=>                 ] (00m:00s) 264 KB / ?? (866.93 KB/s)
pkgs/r/noarch            [=>                 ] (00m:00s) 272 KB / ?? (892.11 KB/s)
pkgs/main/noarch         [=>                 ] (00m:00s) 280 KB / ?? (920.47 KB/s)
pkgs/r/linux-64          [=>                 ] (00m:00s) 264 KB / ?? (866.93 KB/s)
pkgs/r/noarch            [=>                 ] (00m:00s) 272 KB / ?? (892.11 KB/s)
pkgs/main/linux-64       [<=>                ] (00m:00s)
pkgs/main/noarch         [=>                 ] (00m:00s) 280 KB / ?? (920.47 KB/s)
pkgs/r/linux-64          [=>                 ] (00m:00s) 264 KB / ?? (866.93 KB/s)
pkgs/r/noarch            [=>                 ] (00m:00s) 272 KB / ?? (892.11 KB/s)
pkgs/main/linux-64       [=>                 ] (00m:00s) 284 KB / ?? (929.13 KB/s)
pkgs/main/noarch         [<=>                ] (00m:00s) Finalizing...
pkgs/r/linux-64          [=>                 ] (00m:00s) 264 KB / ?? (866.93 KB/s)
pkgs/r/noarch            [=>                 ] (00m:00s) 272 KB / ?? (892.11 KB/s)
pkgs/main/linux-64       [=>                 ] (00m:00s) 284 KB / ?? (929.13 KB/s)
pkgs/main/noarch         [<=>                ] (00m:00s) Done
pkgs/r/linux-64          [=>                 ] (00m:00s) 264 KB / ?? (866.93 KB/s)
pkgs/r/noarch            [=>                 ] (00m:00s) 272 KB / ?? (892.11 KB/s)
pkgs/main/linux-64       [=>                 ] (00m:00s) 284 KB / ?? (929.13 KB/s)
pkgs/main/noarch         [====================] (00m:00s) Done
pkgs/r/linux-64          [=>                 ] (00m:00s) 264 KB / ?? (866.93 KB/s)
pkgs/r/noarch            [=>                 ] (00m:00s) 272 KB / ?? (892.11 KB/s)
```

```
pkgs/main/linux-64      [=>              ] (00m:00s) 284 KB / ?? (929.13 KB/s)
pkgs/r/linux-64         [<=>             ] (00m:00s) 264 KB / ?? (866.93 KB/s)
pkgs/r/noarch           [=>              ] (00m:00s) 272 KB / ?? (892.11 KB/s)
pkgs/main/linux-64      [=>              ] (00m:00s) 284 KB / ?? (929.13 KB/s)
pkgs/r/linux-64         [ <=>               ] (00m:00s) 1 MB / ?? (2.56 MB/s)
pkgs/r/noarch           [=>              ] (00m:00s) 272 KB / ?? (892.11 KB/s)
pkgs/main/linux-64      [=>              ] (00m:00s) 284 KB / ?? (929.13 KB/s)
pkgs/r/linux-64         [ <=>               ] (00m:00s) 1 MB / ?? (2.56 MB/s)
pkgs/r/noarch           [<=>             ] (00m:00s) 272 KB / ?? (892.11 KB/s)
pkgs/main/linux-64      [=>              ] (00m:00s) 284 KB / ?? (929.13 KB/s)
pkgs/r/linux-64         [ <=>               ] (00m:00s) 1 MB / ?? (2.56 MB/s)
pkgs/r/noarch           [ <=>               ] (00m:00s) 1 MB / ?? (2.29 MB/s)
pkgs/main/linux-64      [=>              ] (00m:00s) 284 KB / ?? (929.13 KB/s)
pkgs/r/linux-64         [ <=>               ] (00m:00s) 1 MB / ?? (2.56 MB/s)
pkgs/r/noarch           [ <=>               ] (00m:00s) 1 MB / ?? (2.29 MB/s)
pkgs/main/linux-64      [<=>             ] (00m:00s) 284 KB / ?? (929.13 KB/s)
pkgs/r/linux-64         [ <=>               ] (00m:00s) 1 MB / ?? (2.56 MB/s)
pkgs/r/noarch           [ <=>               ] (00m:00s) 1 MB / ?? (2.29 MB/s)
pkgs/main/linux-64      [ <=>               ] (00m:00s) 1 MB / ?? (2.56 MB/s)
pkgs/r/linux-64         [ <=>               ] (00m:00s) Finalizing...
pkgs/r/noarch           [ <=>               ] (00m:00s) 1 MB / ?? (2.29 MB/s)
pkgs/main/linux-64      [ <=>               ] (00m:00s) 1 MB / ?? (2.56 MB/s)
pkgs/r/linux-64         [ <=>               ] (00m:00s) Done
pkgs/r/noarch           [ <=>               ] (00m:00s) 1 MB / ?? (2.29 MB/s)
pkgs/main/linux-64      [ <=>               ] (00m:00s) 1 MB / ?? (2.56 MB/s)
pkgs/r/linux-64         [====================] (00m:00s) Done
pkgs/r/noarch           [ <=>               ] (00m:00s) 1 MB / ?? (2.29 MB/s)
pkgs/main/linux-64      [ <=>               ] (00m:00s) 1 MB / ?? (2.56 MB/s)
pkgs/r/noarch           [ <=>               ] (00m:00s) Finalizing...
pkgs/main/linux-64      [ <=>               ] (00m:00s) 1 MB / ?? (2.56 MB/s)
pkgs/r/noarch           [ <=>               ] (00m:00s) Done
pkgs/main/linux-64      [ <=>               ] (00m:00s) 1 MB / ?? (2.56 MB/s)
pkgs/r/noarch           [====================] (00m:00s) Done
pkgs/main/linux-64      [ <=>               ] (00m:00s) 1 MB / ?? (2.56 MB/s)
pkgs/main/linux-64      [  <=>              ] (00m:00s) 1 MB / ?? (2.56 MB/s)
pkgs/main/linux-64      [  <=>              ] (00m:00s) 2 MB / ?? (3.58 MB/s)
pkgs/main/linux-64      [   <=>             ] (00m:00s) 2 MB / ?? (3.58 MB/s)
pkgs/main/linux-64      [   <=>             ] (00m:00s) 3 MB / ?? (3.85 MB/s)
pkgs/main/linux-64      [    <=>            ] (00m:00s) 3 MB / ?? (3.85 MB/s)
pkgs/main/linux-64      [    <=>            ] (00m:00s) 4 MB / ?? (4.12 MB/s)
pkgs/main/linux-64      [     <=>           ] (00m:00s) 4 MB / ?? (4.12 MB/s)
pkgs/main/linux-64      [     <=>           ] (00m:00s) 5 MB / ?? (4.33 MB/s)
pkgs/main/linux-64      [     <=>           ] (00m:00s) Finalizing...
pkgs/main/linux-64      [     <=>           ] (00m:01s) Done
pkgs/main/linux-64      [====================] (00m:01s) Done

Pinned packages:
  - python 3.7.*


Transaction

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

  Updating specs:
```

```
  - bs4==4.10.0
  - ca-certificates
  - certifi
  - openssl


  Package                 Version  Build           Channel                   Size
-----------------------------------------------------------------------------------
  Install:
-----------------------------------------------------------------------------------

  + bs4                   4.10.0   hd3eb1b0_0      pkgs/main/noarch          10 KB

  Upgrade:
-----------------------------------------------------------------------------------

  - ca-certificates    2022.9.24   ha878542_0      installed
  + ca-certificates   2023.01.10   h06a4308_0      pkgs/main/linux-64       120 KB
  - certifi            2022.9.24   pyhd8ed1ab_0    installed
  + certifi            2022.12.7   py37h06a4308_0  pkgs/main/linux-64       150 KB
  - openssl               1.1.1s   h0b41bf4_1      installed
  + openssl               1.1.1t   h7f8727e_0      pkgs/main/linux-64         4 MB

  Downgrade:
-----------------------------------------------------------------------------------

  - beautifulsoup4        4.11.1   pyha770c72_0    installed
  + beautifulsoup4        4.10.0   pyh06a4308_0    pkgs/main/noarch          85 KB

  Summary:

  Install: 1 packages
  Upgrade: 3 packages
  Downgrade: 1 packages

  Total download: 4 MB

-----------------------------------------------------------------------------------

Downloading  [>                                        ] (00m:00s)   39.09 KB/s
Extracting   [>                                                      ] (--:--)
Finished bs4                                (00m:00s)                10 KB     39
KB/s
Downloading  [>                                        ] (00m:00s)   39.09 KB/s
Extracting   [>                                                      ] (--:--)
Downloading  [>                                        ] (00m:00s)   39.09 KB/s
Extracting   [>                                                      ] (--:--)
Downloading  [>                                        ] (00m:00s)   39.09 KB/s
Extracting   [>                                                      ] (--:--)
Downloading  [>                                        ] (00m:00s)  362.19 KB/s
Extracting   [>                                                      ] (--:--)
Downloading  [>                                        ] (00m:00s)  362.19 KB/s
Extracting   [========>                                ] (00m:00s)        1 / 5
Finished beautifulsoup4                     (00m:00s)                85 KB    323
KB/s
Downloading  [>                                        ] (00m:00s)  362.19 KB/s
```

```
Extracting   [========>                                ] (00m:00s)        1 / 5
Downloading  [>                                       ] (00m:00s)  362.19 KB/s
Extracting   [========>                                ] (00m:00s)        1 / 5
Downloading  [>                                       ] (00m:00s)  362.19 KB/s
Extracting   [========>                                ] (00m:00s)        1 / 5
Downloading  [==>                                     ] (00m:00s)  806.08 KB/s
Extracting   [========>                                ] (00m:00s)        1 / 5
Finished ca-certificates                    (00m:00s)             120 KB    450
KB/s
Downloading  [==>                                     ] (00m:00s)  806.08 KB/s
Extracting   [========>                                ] (00m:00s)        1 / 5
Downloading  [==>                                     ] (00m:00s)  806.08 KB/s
Extracting   [========>                                ] (00m:00s)        1 / 5
Downloading  [===>                                    ] (00m:00s)    1.33 MB/s
Extracting   [========>                                ] (00m:00s)        1 / 5
Finished certifi                            (00m:00s)             150 KB    561
KB/s
Downloading  [===>                                    ] (00m:00s)    1.33 MB/s
Extracting   [========>                                ] (00m:00s)        1 / 5
Downloading  [===>                                    ] (00m:00s)    1.33 MB/s
Extracting   [========>                                ] (00m:00s)        1 / 5
Downloading  [===>                                    ] (00m:00s)    1.33 MB/s
Extracting   [================>                      ] (00m:00s)        2 / 5
Downloading  [===>                                    ] (00m:00s)    1.33 MB/s
Extracting   [================>                      ] (00m:00s)        2 / 5
Downloading  [======================================== ] (00m:00s)   13.78 MB/s
Extracting   [================>                      ] (00m:00s)        2 / 5
Downloading  [======================================== ] (00m:00s)   13.78 MB/s
Extracting   [========================>              ] (00m:00s)        3 / 5
Downloading  [======================================== ] (00m:00s)   13.78 MB/s
Extracting   [========================>              ] (00m:00s)        3 / 5
Downloading  [======================================== ] (00m:00s)   13.78 MB/s
Extracting   [================================>      ] (00m:00s)        4 / 5
Finished openssl                            (00m:00s)              4 MB     13
MB/s
Downloading  [======================================== ] (00m:00s)   13.78 MB/s
Extracting   [================================>      ] (00m:00s)        4 / 5
Downloading  [======================================== ] (00m:00s)   13.78 MB/s
Extracting   [================================>      ] (00m:00s)        4 / 5
Downloading  [======================================== ] (00m:00s)   13.78 MB/s
Extracting   [================================>      ] (00m:00s)        4 / 5
Downloading  [======================================== ] (00m:00s)   13.78 MB/s
Extracting   [======================================== ] (00m:00s)        5 / 5
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```





```
        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

  Updating specs:

   - html5lib==1.1
   - ca-certificates
   - certifi
   - openssl


  Package         Version  Build         Channel                 Size
-------------------------------------------------------------------------
  Install:
-------------------------------------------------------------------------

  + html5lib          1.1  pyhd3eb1b0_0  pkgs/main/noarch       91 KB
  + webencodings    0.5.1  py37_1        pkgs/main/linux-64     19 KB

  Summary:

  Install: 2 packages

  Total download: 110 KB

-------------------------------------------------------------------------


Downloading  [================================>       ] (00m:00s)  667.81 KB/s
Extracting   [>                                        ] (--:--)
Finished html5lib                          (00m:00s)             91 KB    667
```

```
KB/s
Downloading  [====================================>       ] (00m:00s)  667.81 KB/s
Extracting   [>                                            ] (--:--)
Downloading  [====================================>       ] (00m:00s)  667.81 KB/s
Extracting   [>                                            ] (--:--)
Downloading  [====================================>       ] (00m:00s)  667.81 KB/s
Extracting   [>                                            ] (--:--)
Downloading  [============================================] (00m:00s)  790.77 KB/s
Extracting   [>                                            ] (--:--)
Finished webencodings                          (00m:00s)              19 KB    137
KB/s
Downloading  [============================================] (00m:00s)  790.77 KB/s
Extracting   [>                                            ] (--:--)
Downloading  [============================================] (00m:00s)  790.77 KB/s
Extracting   [>                                            ] (--:--)
Downloading  [============================================] (00m:00s)  790.77 KB/s
Extracting   [====================>                       ] (00m:00s)        1 / 2
Downloading  [============================================] (00m:00s)  790.77 KB/s
Extracting   [====================>                       ] (00m:00s)        1 / 2
Downloading  [============================================] (00m:00s)  790.77 KB/s
Extracting   [============================================] (00m:00s)        2 / 2
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Collecting lxml==4.6.4
  Downloading lxml-4.6.4-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.man
ylinux_2_24_x86_64.whl (6.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.3/6.3 MB 62.8 MB/s eta 0:00:00:00:
0100:01
Installing collected packages: lxml
  Attempting uninstall: lxml
    Found existing installation: lxml 4.9.1
    Uninstalling lxml-4.9.1:
      Successfully uninstalled lxml-4.9.1
Successfully installed lxml-4.6.4
```

In [ ]:
```
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.

In [ ]:
```
url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDevelo

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

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

In [ ]:
```
soup = BeautifulSoup(data, 'html5lib')
```

Now we can turn the html table into a pandas dataframe

In [ ]:
```
netflix_data = pd.DataFrame(columns=["Date", "Open", "High", "Low", "Close", "Volum

# 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
```

```
[<td class="Py(10px) Ta(start) Pend(10px)" data-reactid="1087"><span data-reactid
="1088">Sep 01, 2015</span></td>, <td class="Py(10px) Pstart(10px)" data-reactid
="1089"><span data-reactid="1090">109.35</span></td>, <td class="Py(10px) Pstart(1
0px)" data-reactid="1091"><span data-reactid="1092">111.24</span></td>, <td class
="Py(10px) Pstart(10px)" data-reactid="1093"><span data-reactid="1094">93.55</span
></td>, <td class="Py(10px) Pstart(10px)" data-reactid="1095"><span data-reactid
="1096">103.26</span></td>, <td class="Py(10px) Pstart(10px)" data-reactid="1097">
<span data-reactid="1098">103.26</span></td>, <td class="Py(10px) Pstart(10px)" da
ta-reactid="1099"><span data-reactid="1100">497,401,200</span></td>]
```

We can now print out the dataframe

In [6]:
```
netflix_data.head()
```

Out[6]:

	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

In [7]:
```
read_html_pandas_data = pd.read_html(url)
```

Or we can convert the BeautifulSoup object to a string

In [8]:
```
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

In [9]:
```
netflix_dataframe = read_html_pandas_data[0]

netflix_dataframe.head()
```

Out[9]:

	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`.

In [10]:
```
html_data=requests.get("https://cf-courses-data.s3.us.cloud-object-storage.appdomai
```

Parse the html data using `beautiful_soup`.

In [11]:
```
data=BeautifulSoup(html_data,'html5lib')
```

Question 1 What is the content of the title attribute:

In [12]:
```
data.find("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`.

In [14]:
```
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]
    Open = col[1]#ADD_CODE
    high = col[2]#ADD_CODE
    low = col[3]#ADD_CODE
```

```
    close = col[4]#ADD_CODE
    adj_close = col[4]#ADD_CODE
    volume = col[5]#ADD_CODE

    amazon_data = amazon_data.append({"Date":date, "Open":Open, "High":high, "Low":
```

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

In [15]:
```
amazon_data.head(5)
```

Out[15]:

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

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

In [20]:
```
amazon_data.columns
```

Out[20]:
```
Index(['Date', 'Open', 'High', 'Low', 'Close', 'Volume', 'Adj Close'], dtype='obje
ct')
```

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

In [18]:
```
print(amazon_data['Open'][4])
```

```
<td class="Py(10px) Pstart(10px)" data-reactid="114"><span data-reactid="115">536.
79</span></td>
```

About the Authors:

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