File Edit View Run Kernel Git Tabs Settings Help

Filter files by name

/ ... / labs / project /

Name	Last Modified
amd.json	2 years ago
apple.json	2 years ago
Final Assig...	2 hours ago
Final Assig...	34 minutes ago
Final_Assi...	4 hours ago
WebScrapi...	5 hours ago

Final Assignment.ipynb

Markdown git Run as Pipeline Python

Skills Network

Extracting and Visualizing Stock Data

Description

Extracting essential data from a dataset and displaying it is a necessary part of data science; therefore individuals can make correct decisions based on the data. In this assignment, you will extract some stock data, you will then display this data in a graph.

Table of Contents

Estimated Time Needed: **30 min**

Note:- If you are working in IBM Cloud Watson Studio, please replace the command for installing nbformat from `!pip install nbformat==4.2.0` to simply `!pip install nbformat`

[71]:

```
!pip install yfinance==0.1.67
!mamba install bs4==4.10.0 -y
!pip install nbformat==4.2.0
```

```
Requirement already satisfied: yfinance==0.1.67 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (0.1.67)
Requirement already satisfied: pandas>=0.24 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance==0.1.67) (1.3.5)
Requirement already satisfied: numpy>=1.15 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance==0.1.67) (1.21.6)
Requirement already satisfied: requests>=2.20 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance==0.1.67) (2.29.0)
Requirement already satisfied: multitasking>=0.0.7 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance==0.1.67) (0.0.11)
Requirement already satisfied: lxml>=4.5.1 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance==0.1.67) (4.6.4)
Requirement already satisfied: python-dateutil>=2.7.3 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from pandas>=0.24->yfinance==0.1.67) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from pandas>=0.24->yfinance==0.1.67) (2023.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from requests>=2.20->yfinance==0.1.67) (3.1.0)
Requirement already satisfied: idna<4,>=2.5 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from requests>=2.20->yfinance==0.1.67) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from requests>=2.20->yfinance==0.1.67) (1.26.15)
Requirement already satisfied: certifi>=2017.4.17 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from requests>=2.20->yfinance==0.1.67) (2023.5.7)
Requirement already satisfied: six>=1.5 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from python-dateutil>=2.7.3->pandas>=0.24->yfinance==0.1.67) (1.16.0)
```

```
        mamba (1.4.2) supported by @QuantStack

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


Looking for: ['bs4==4.10.0']

[+] 0.0s
[+] 0.1s
pkgs/main/linux-64                     0.0 B /  ??.?MB @  ??.?MB/s  0.1s
pkgs/main/noarch                       0.0 B /  ??.?MB @  ??.?MB/s  0.1s
pkgs/r/linux-64                        0.0 B /  ??.?MB @  ??.?MB/s  0.1s
pkgs/r/noarch                          0.0 B /  ??.?MB @  ??.?MB/s  0.1spkgs/main/linux-64
No change
pkgs/main/noarch                                          No change
pkgs/r/noarch                                             No change
```

```
pkgs/r/linux-64                                             No change

Pinned packages:
  - python 3.7.*


Transaction

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

  All requested packages already installed

Requirement already satisfied: nbformat==4.2.0 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (4.2.0)
Requirement already satisfied: ipython-genutils in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from nbformat=
=4.2.0) (0.2.0)
Requirement already satisfied: jsonschema!=2.5.0,>=2.4 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from nb
format==4.2.0) (4.17.3)
Requirement already satisfied: jupyter-core in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from nbformat==4.
2.0) (4.12.0)
Requirement already satisfied: traitlets>=4.1 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from nbformat==
4.2.0) (5.9.0)
Requirement already satisfied: attrs>=17.4.0 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from jsonschema!=
2.5.0,>=2.4->nbformat==4.2.0) (23.1.0)
Requirement already satisfied: importlib-metadata in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from jsonsch
ema!=2.5.0,>=2.4->nbformat==4.2.0) (4.11.4)
Requirement already satisfied: importlib-resources>=1.4.0 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
jsonschema!=2.5.0,>=2.4->nbformat==4.2.0) (5.12.0)
Requirement already satisfied: pkgutil-resolve-name>=1.3.10 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (fr
om jsonschema!=2.5.0,>=2.4->nbformat==4.2.0) (1.3.10)
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in /home/jupyterlab/conda/envs/python/lib/python3.7/
site-packages (from jsonschema!=2.5.0,>=2.4->nbformat==4.2.0) (0.19.3)
Requirement already satisfied: typing-extensions in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from jsonsche
ma!=2.5.0,>=2.4->nbformat==4.2.0) (4.5.0)
Requirement already satisfied: zipp>=3.1.0 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from importlib-reso
urces>=1.4.0->jsonschema!=2.5.0,>=2.4->nbformat==4.2.0) (3.15.0)
```

[72]:
```
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
```

In Python, you can ignore warnings using the warnings module. You can use the filterwarnings function to filter or ignore specific warning messages or categories.

[73]:
```
import warnings
# Ignore all warnings
warnings.filterwarnings("ignore", category=FutureWarning)
```

Define Graphing Function

In this section, we define the function `make_graph`. You don't have to know how the function works, you should only care about the inputs. It takes a dataframe with stock data (dataframe must contain Date and Close columns), a dataframe with revenue data (dataframe must contain Date and Revenue columns), and the name of the stock.

[74]:
```
def make_graph(stock_data, revenue_data, stock):
    fig = make_subplots(rows=2, cols=1, shared_xaxes=True, subplot_titles=("Historical Share Price", "Historical Revenue"), verti
    stock_data_specific = stock_data[stock_data.Date <= '2021--06-14']
    revenue_data_specific = revenue_data[revenue_data.Date <= '2021-04-30']
    fig.add_trace(go.Scatter(x=pd.to_datetime(stock_data_specific.Date, infer_datetime_format=True), y=stock_data_specific.Close.
    fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data_specific.Date, infer_datetime_format=True), y=revenue_data_specific.Re
    fig.update_xaxes(title_text="Date", row=1, col=1)
    fig.update_xaxes(title_text="Date", row=2, col=1)
    fig.update_yaxes(title_text="Price ($US)", row=1, col=1)
    fig.update_yaxes(title_text="Revenue ($US Millions)", row=2, col=1)
    fig.update_layout(showlegend=False,
    height=900,
    title=stock,
    xaxis_rangeslider_visible=True)
    fig.show()
```

Question 1: Use yfinance to Extract Stock Data

Using the `Ticker` function enter the ticker symbol of the stock we want to extract data on to create a ticker object. The stock is Tesla and its ticker symbol is `TSLA`.

[77]:
```
stock_data = yf.Ticker('TSLA')
print(stock_data)
```

```
yfinance.Ticker object <TSLA>
```

Using the ticker object and the function `history` extract stock information and save it in a dataframe named `tesla_data`. Set the `period` parameter to `max` so we get information for the maximum amount of time.

[78]:
```
tesla_data = stock_data.history(period='max')
tesla_data
```

[78]:

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

2024-02-26	192.289993	201.779999	192.000000	199.399994	111747100	0	0.0
2024-02-27	204.039993	205.600006	198.259995	199.729996	108645400	0	0.0
2024-02-28	200.419998	205.300003	198.440002	202.039993	99806200	0	0.0
2024-02-29	204.179993	205.279999	198.449997	201.880005	85907000	0	0.0
2024-03-01	200.520004	204.520004	198.500000	202.639999	82099200	0	0.0

3442 rows × 7 columns

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.

[79]:
```
tesla_data.reset_index(inplace=True)
tesla_data.head()
```

[79]:

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

Question 2: Use Webscraping to Extract Tesla 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/revenue.htm Save the text of the response as a variable named `html_data`.

[80]:
```
url="https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/pro
```

Parse the html data using `beautiful_soup`.

[81]:
```
data = requests.get(url).text
soup = BeautifulSoup(data, 'html5lib')
```

Using `BeautifulSoup` or the `read_html` function extract the table with `Tesla Revenue` and store it into a dataframe named `tesla_revenue`. The dataframe should have columns `Date` and `Revenue`.

▶ Click here if you need help locating the table

[82]:
```
tesla_revenue = soup.find('table')
rows = table.find_all('tr')[1:]  # Exclude the header row
tesla_revenue_data = []
for row in rows:
    cols = row.find_all('td')
    date = cols[0].get_text()
    revenue = cols[1].get_text().strip()
    tesla_revenue_data.append({'Date': date, 'Revenue': revenue})
tesla_revenue = pd.DataFrame(tesla_revenue_data)
print(tesla_revenue)
```

```
    Date Revenue
0   2020  $6,466
1   2019  $8,285
2   2018  $8,547
3   2017  $7,965
4   2016  $9,364
5   2015  $9,296
6   2014  $9,040
7   2013  $8,887
8   2012  $9,551
9   2011  $9,474
10  2010  $9,078
11  2009  $8,806
12  2008  $7,094
13  2007  $5,319
14  2006  $3,092
15  2005  $1,843
```

Execute the following line to remove the comma and dollar sign from the `Revenue` column.

[27]:
```
tesla_revenue["Revenue"] = tesla_revenue['Revenue'].str.replace(',|\$',"")
print(tesla_revenue)
```

```
    Date Revenue
0   2021   53823
1   2020   31536
2   2019   24578
3   2018   21461
4   2017   11759
5   2016    7000
6   2015    4046
7   2014    3198
8   2013    2013
9   2012     413
10  2011     204
11  2010     117
12  2009     112
```

Execute the following lines to remove an null or empty strings in the Revenue column.

[83]:
```
tesla_revenue.dropna(inplace=True)

tesla_revenue = tesla_revenue[tesla_revenue['Revenue'] != ""]
print(tesla_revenue)
```

```
    Date Revenue
0   2020  $6,466
1   2019  $8,285
2   2018  $8,547
3   2017  $7,965
4   2016  $9,364
5   2015  $9,296
6   2014  $9,040
7   2013  $8,887
8   2012  $9,551
9   2011  $9,474
10  2010  $9,078
11  2009  $8,806
12  2008  $7,094
13  2007  $5,319
14  2006  $3,092
15  2005  $1,843
```

Display the last 5 row of the `tesla_revenue` dataframe using the `tail` function. Take a screenshot of the results.

[84]:
```
tesla_revenue.tail(5)
```

[84]:

	Date	Revenue
11	2009	$8,806
12	2008	$7,094
13	2007	$5,319
14	2006	$3,092
15	2005	$1,843

Question 3: Use yfinance to Extract Stock Data

Using the `Ticker` function enter the ticker symbol of the stock we want to extract data on to create a ticker object. The stock is GameStop and its ticker symbol is `GME`.

[85]:
```
data = yf.Ticker("GME")

print(data)
```

```
yfinance.Ticker object <GME>
```

Using the ticker object and the function `history` extract stock information and save it in a dataframe named `gme_data`. Set the `period` parameter to `max` so we get information for the maximum amount of time.

[86]:
```
gme_data = data.history(period="max")
(gme_data)
```

[86]:

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
2002-02-13	1.620128	1.693350	1.603296	1.691666	76216000	0.0	0.0
2002-02-14	1.712707	1.716073	1.670626	1.683250	11021600	0.0	0.0
2002-02-15	1.683250	1.687458	1.658002	1.674834	8389600	0.0	0.0
2002-02-19	1.666418	1.666418	1.578047	1.607504	7410400	0.0	0.0
2002-02-20	1.615920	1.662210	1.603296	1.662210	6892800	0.0	0.0
...	...	...	...	...	...	...	...
2024-02-26	13.310000	13.750000	13.200000	13.680000	2278600	0.0	0.0
2024-02-27	13.700000	14.290000	13.680000	14.210000	2795500	0.0	0.0
2024-02-28	14.000000	14.470000	13.890000	14.240000	2440700	0.0	0.0
2024-02-29	14.340000	14.650000	14.030000	14.270000	2638700	0.0	0.0
2024-03-01	14.180000	15.090000	13.990000	14.950000	4890900	0.0	0.0

5550 rows × 7 columns

Reset the index using the `reset_index(inplace=True)` function on the gme_data DataFrame and display the first five rows of the `gme_data` dataframe using the `head` function. Take a screenshot of the results and code from the beginning of Question 3 to the results below.

[87]:
```
gme_data.reset_index(inplace=True)
gme_data.head()
```

[87]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2002-02-13	1.620128	1.693350	1.603296	1.691666	76216000	0.0	0.0
1	2002-02-14	1.712707	1.716073	1.670626	1.683250	11021600	0.0	0.0
2	2002-02-15	1.683250	1.687458	1.658002	1.674834	8389600	0.0	0.0
3	2002-02-19	1.666418	1.666418	1.578047	1.607504	7410400	0.0	0.0
4	2002-02-20	1.615920	1.662210	1.603296	1.662210	6892800	0.0	0.0

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`.

[88]:
```
url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/p
```

Parse the html data using `beautiful_soup`.

[90]:
```
html_data = requests.get(url).text
```

Using `BeautifulSoup` or the `read_html` function extract the table with `GameStop 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

[98]:
```
gme_revenue = soup.find('table')
rows = table.find_all('tr')[1:]
gme_revenue_data = []
for row in rows:
    cols = row.find_all('td')
    date = cols[0].get_text()
    revenue = cols[1].get_text().strip()
    gme_revenue_data.append({'Date': date, 'Revenue': revenue})
gme_revenue = pd.DataFrame(gme_revenue_data)
```

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

[97]:
```
gme_revenue.tail()
```

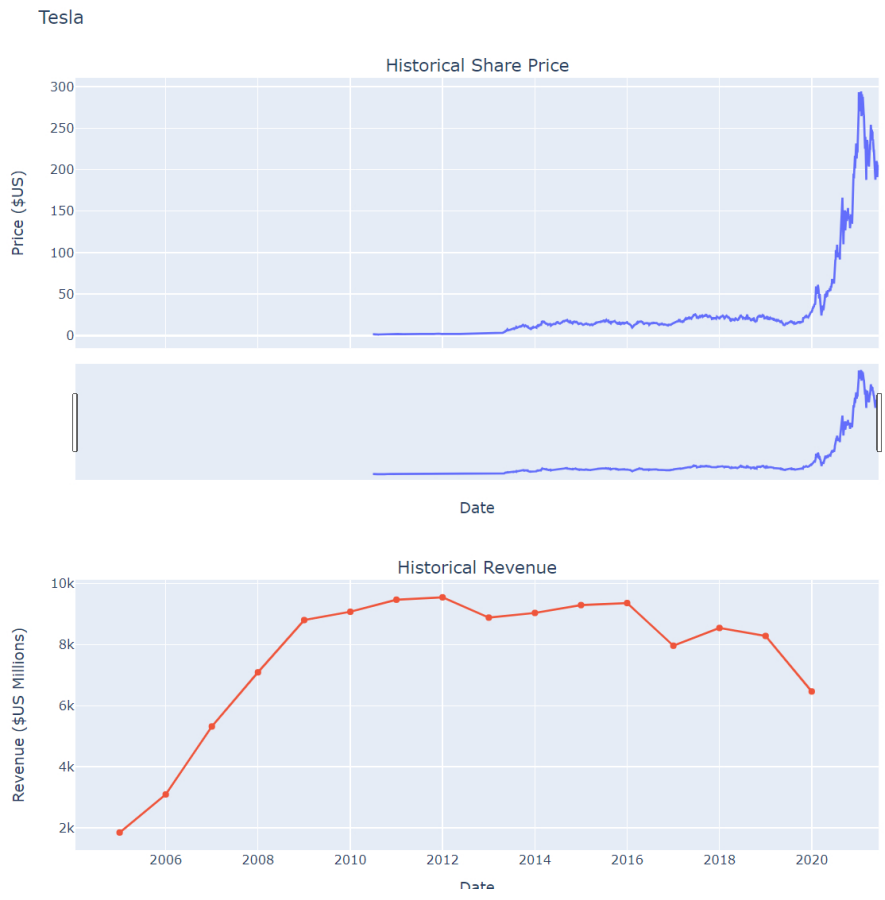
[97]:

	Date	Revenue
11	2009	$8,806
12	2008	$7,094
13	2007	$5,319
14	2006	$3,092
15	2005	$1,843

Question 5: Plot Tesla Stock Graph

Use the `make_graph` function to graph the Tesla Stock Data, also provide a title for the graph. The structure to call the `make_graph` function is `make_graph(tesla_data, tesla_revenue, 'Tesla')`. Note the graph will only show data upto June 2021.

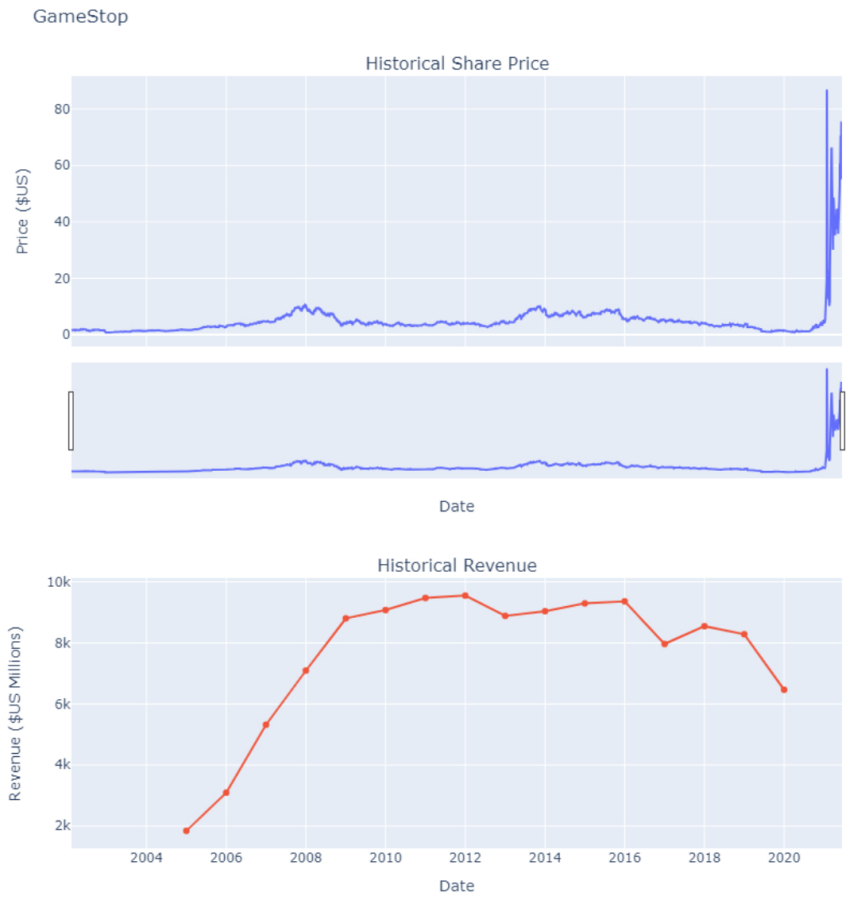
[104]:
```
tesla_revenue["Revenue"] = tesla_revenue['Revenue'].str.replace(',|\$',"")
make_graph(tesla_data, tesla_revenue, 'Tesla')
```



Question 6: Plot GameStop Stock Graph

Use the `make_graph` function to graph the GameStop Stock Data, also provide a title for the graph. The structure to call the `make_graph` function is `make_graph(gme_data, gme_revenue, 'GameStop')`. Note the graph will only show data upto June 2021.

[106]:
```
gme_revenue["Revenue"] = gme_revenue['Revenue'].str.replace(',|\$',"")
make_graph(gme_data, gme_revenue, 'GameStop')
```



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
2022-02-28	1.2	Lakshmi Holla	Changed the URL of GameStop
2020-11-10	1.1	Malika Singla	Deleted the Optional part
2020-08-27	1.0	Malika Singla	Added lab to GitLab


Simple
0
3
Fully initialized
Python | Idle
Mem: 764.54 / 6144.00 MB
Mode: Command
Ln 1, Col 1
English (United States)
Final Assignment.ipynb
1