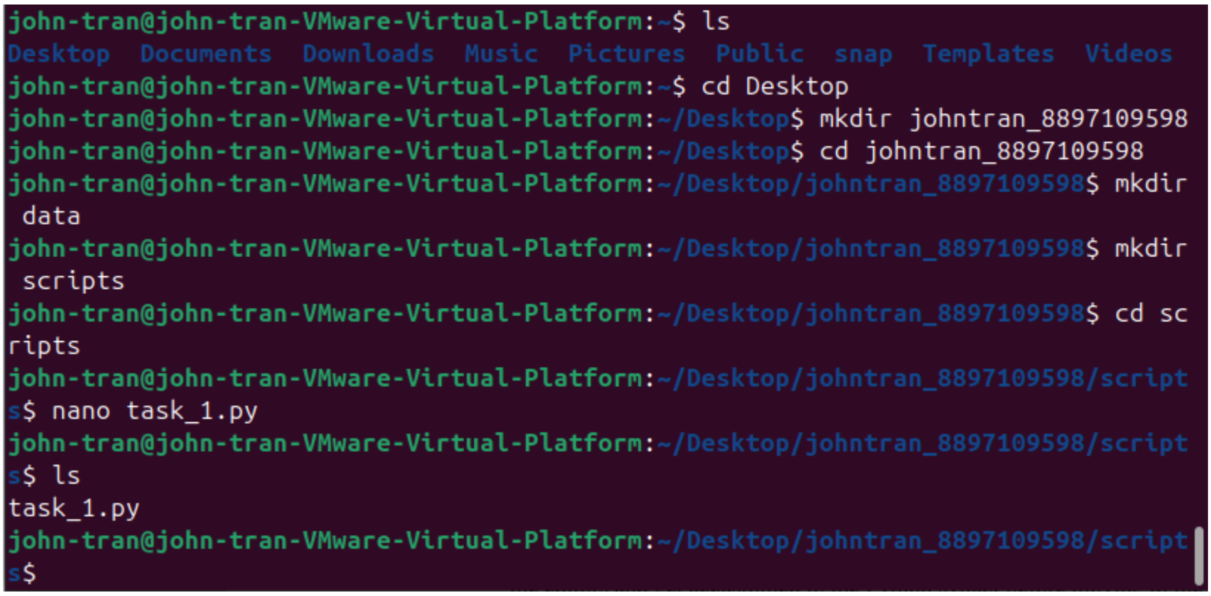
John Tran

8897109598

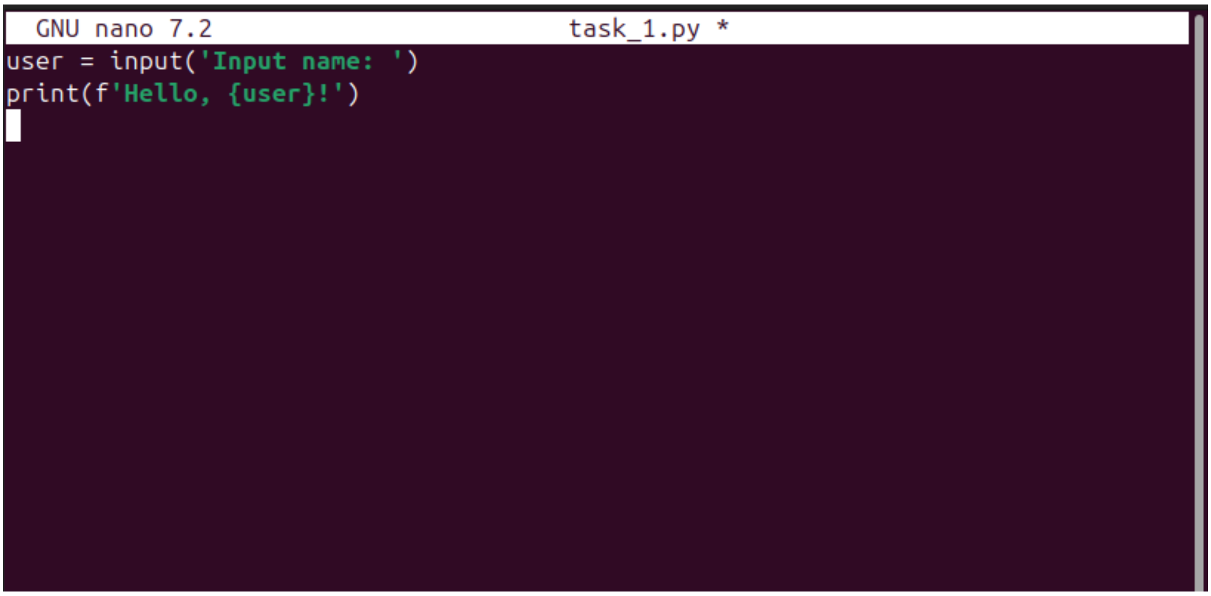
2.1)



```
john-tran@john-tran-VMware-Virtual-Platform:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  snap  Templates  Videos
john-tran@john-tran-VMware-Virtual-Platform:~$ cd Desktop
john-tran@john-tran-VMware-Virtual-Platform:~/Desktop$ mkdir johntran_8897109598
john-tran@john-tran-VMware-Virtual-Platform:~/Desktop$ cd johntran_8897109598
john-tran@john-tran-VMware-Virtual-Platform:~/Desktop/johntran_8897109598$ mkdir
 data
john-tran@john-tran-VMware-Virtual-Platform:~/Desktop/johntran_8897109598$ mkdir
 scripts
john-tran@john-tran-VMware-Virtual-Platform:~/Desktop/johntran_8897109598$ cd sc
ripts
john-tran@john-tran-VMware-Virtual-Platform:~/Desktop/johntran_8897109598/script
s$ nano task_1.py
john-tran@john-tran-VMware-Virtual-Platform:~/Desktop/johntran_8897109598/script
s$ ls
task_1.py
john-tran@john-tran-VMware-Virtual-Platform:~/Desktop/johntran_8897109598/script
s$
```

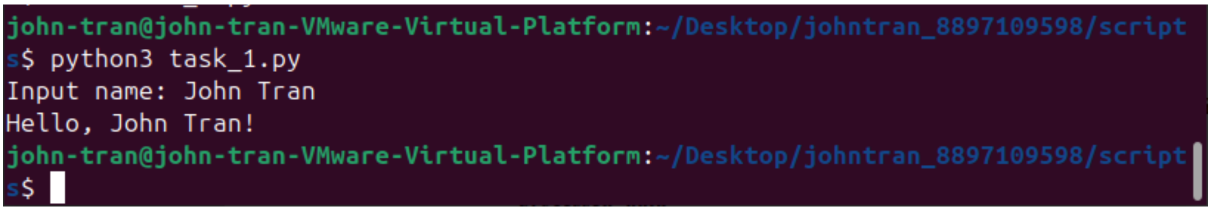
I created my class folder, the folder **data** and **scripts**, and the **task_1.py** file inside of **scripts**.

2.2)



```
  GNU nano 7.2                          task_1.py *
user = input('Input name: ')
print(f'Hello, {user}!')
```

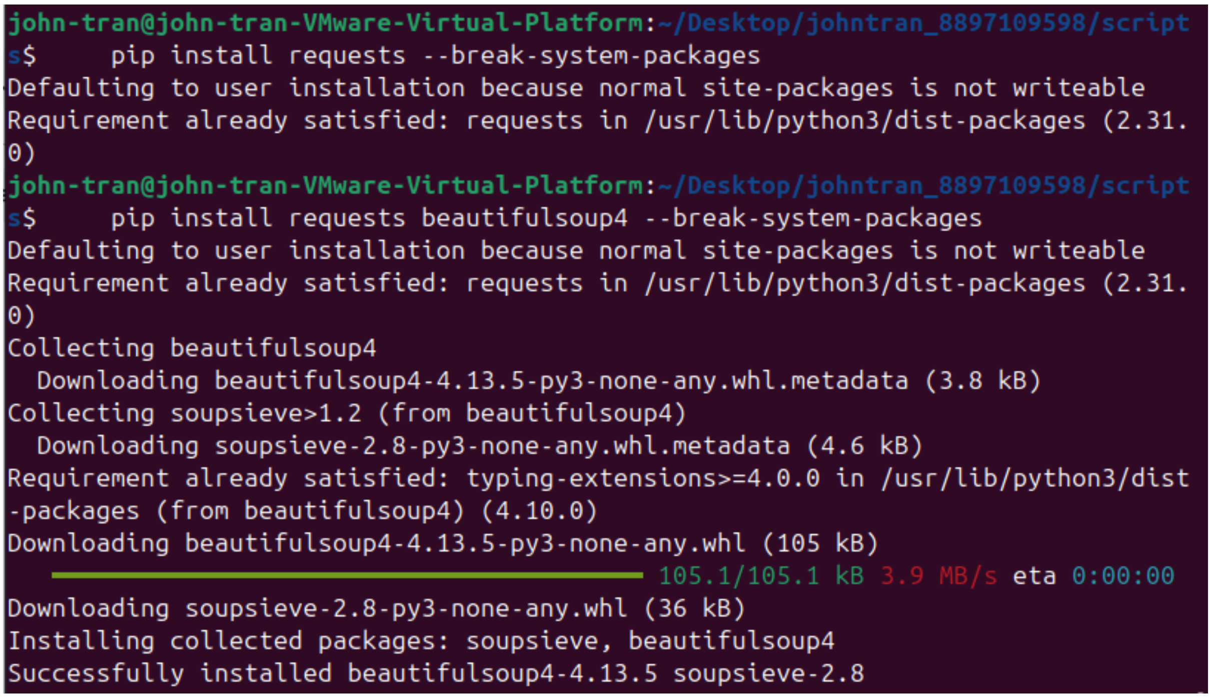
I wrote the python script that reads a user's name as input and greets the user with "Hello, [name]!".



```
john-tran@john-tran-VMware-Virtual-Platform:~/Desktop/johntran_8897109598/script
s$ python3 task_1.py
Input name: John Tran
Hello, John Tran!
john-tran@john-tran-VMware-Virtual-Platform:~/Desktop/johntran_8897109598/script
s$ 
```

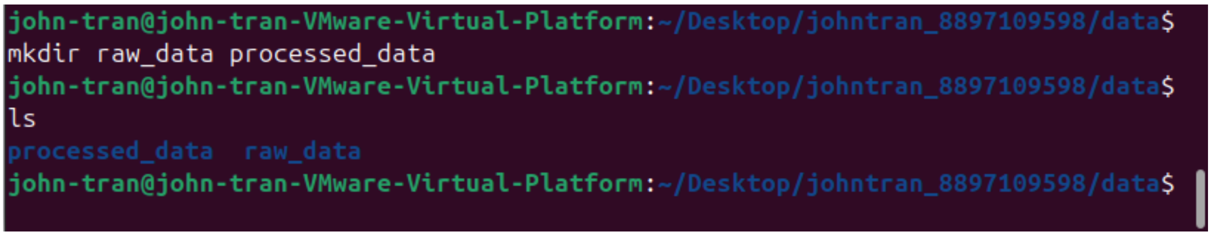
The script prints out ‘Hello, John Tran!’ when I entered my name ‘John Tran’.

2.3)



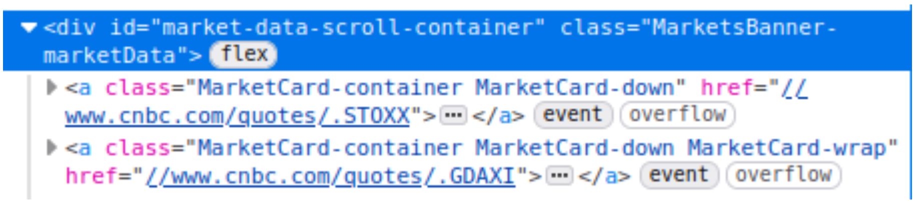
```
john-tran@john-tran-VMware-Virtual-Platform:~/Desktop/johntran_8897109598/script
s$     pip install requests --break-system-packages
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (2.31.
0)
john-tran@john-tran-VMware-Virtual-Platform:~/Desktop/johntran_8897109598/script
s$     pip install requests beautifulsoup4 --break-system-packages
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (2.31.
0)
Collecting beautifulsoup4
  Downloading beautifulsoup4-4.13.5-py3-none-any.whl.metadata (3.8 kB)
Collecting soupsieve>1.2 (from beautifulsoup4)
  Downloading soupsieve-2.8-py3-none-any.whl.metadata (4.6 kB)
Requirement already satisfied: typing-extensions>=4.0.0 in /usr/lib/python3/dist
-packages (from beautifulsoup4) (4.10.0)
Downloading beautifulsoup4-4.13.5-py3-none-any.whl (105 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 105.1/105.1 kB 3.9 MB/s eta 0:00:00
Downloading soupsieve-2.8-py3-none-any.whl (36 kB)
Installing collected packages: soupsieve, beautifulsoup4
Successfully installed beautifulsoup4-4.13.5 soupsieve-2.8
```

I installed **requests** and **beautifulsoup4**.

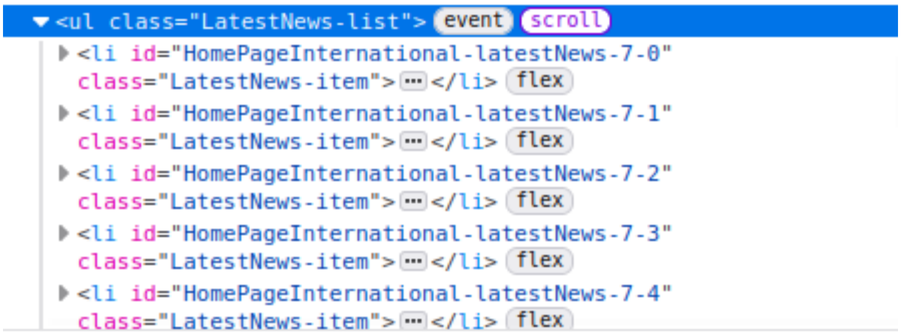


```
john-tran@john-tran-VMware-Virtual-Platform:~/Desktop/johntran_8897109598/data$
mkdir raw_data processed_data
john-tran@john-tran-VMware-Virtual-Platform:~/Desktop/johntran_8897109598/data$
ls
processed_data  raw_data
john-tran@john-tran-VMware-Virtual-Platform:~/Desktop/johntran_8897109598/data$
```

I created the folders **processed_data** and **raw_data**.

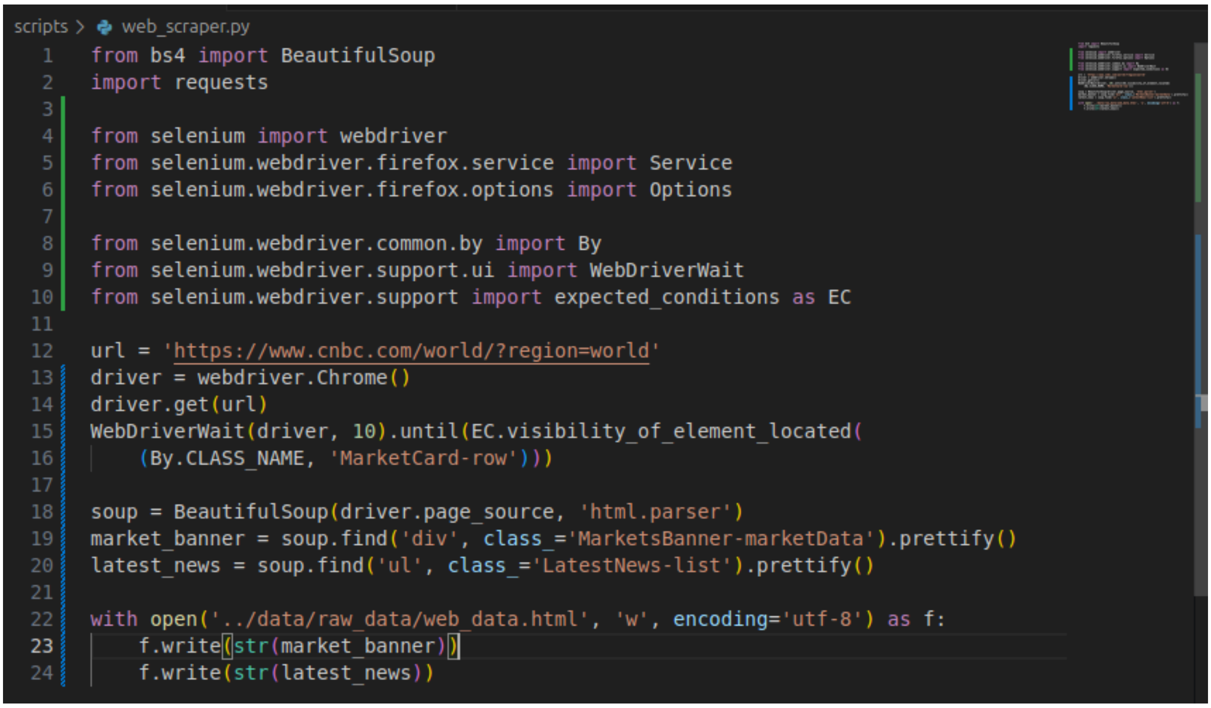


```
▼<div id="market-data-scroll-container" class="MarketsBanner-
 marketData"> flex
  ▶<a class="MarketCard-container MarketCard-down" href="//
   www.cnbc.com/quotes/.STOXX">⋯</a> event overflow
  ▶<a class="MarketCard-container MarketCard-down MarketCard-wrap"
   href="//www.cnbc.com/quotes/.GDAXI">⋯</a> event overflow
```



```
▼<ul class="LatestNews-list"> event scroll
  ▶<li id="HomePageInternational-latestNews-7-0"
   class="LatestNews-item">⋯</li> flex
  ▶<li id="HomePageInternational-latestNews-7-1"
   class="LatestNews-item">⋯</li> flex
  ▶<li id="HomePageInternational-latestNews-7-2"
   class="LatestNews-item">⋯</li> flex
  ▶<li id="HomePageInternational-latestNews-7-3"
   class="LatestNews-item">⋯</li> flex
  ▶<li id="HomePageInternational-latestNews-7-4"
   class="LatestNews-item">⋯</li> flex
```

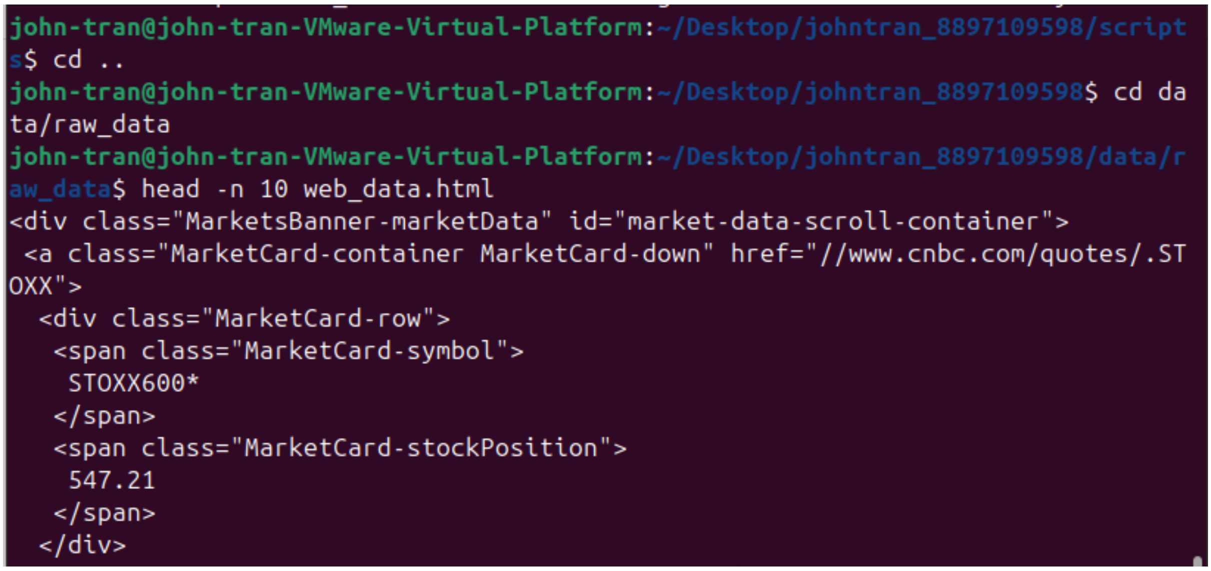
I found the tags for both the Market Banner and the Latest News section.



scripts > web_scraper.py

```python
from bs4 import BeautifulSoup
import requests

from selenium import webdriver
from selenium.webdriver.firefox.service import Service
from selenium.webdriver.firefox.options import Options

from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

url = 'https://www.cnbc.com/world/?region=world'
driver = webdriver.Chrome()
driver.get(url)
WebDriverWait(driver, 10).until(EC.visibility_of_element_located(
    (By.CLASS_NAME, 'MarketCard-row')))

soup = BeautifulSoup(driver.page_source, 'html.parser')
market_banner = soup.find('div', class_='MarketsBanner-marketData').prettify()
latest_news = soup.find('ul', class_='LatestNews-list').prettify()

with open('../data/raw_data/web_data.html', 'w', encoding='utf-8') as f:
    f.write(str(market_banner))
    f.write(str(latest_news))
```

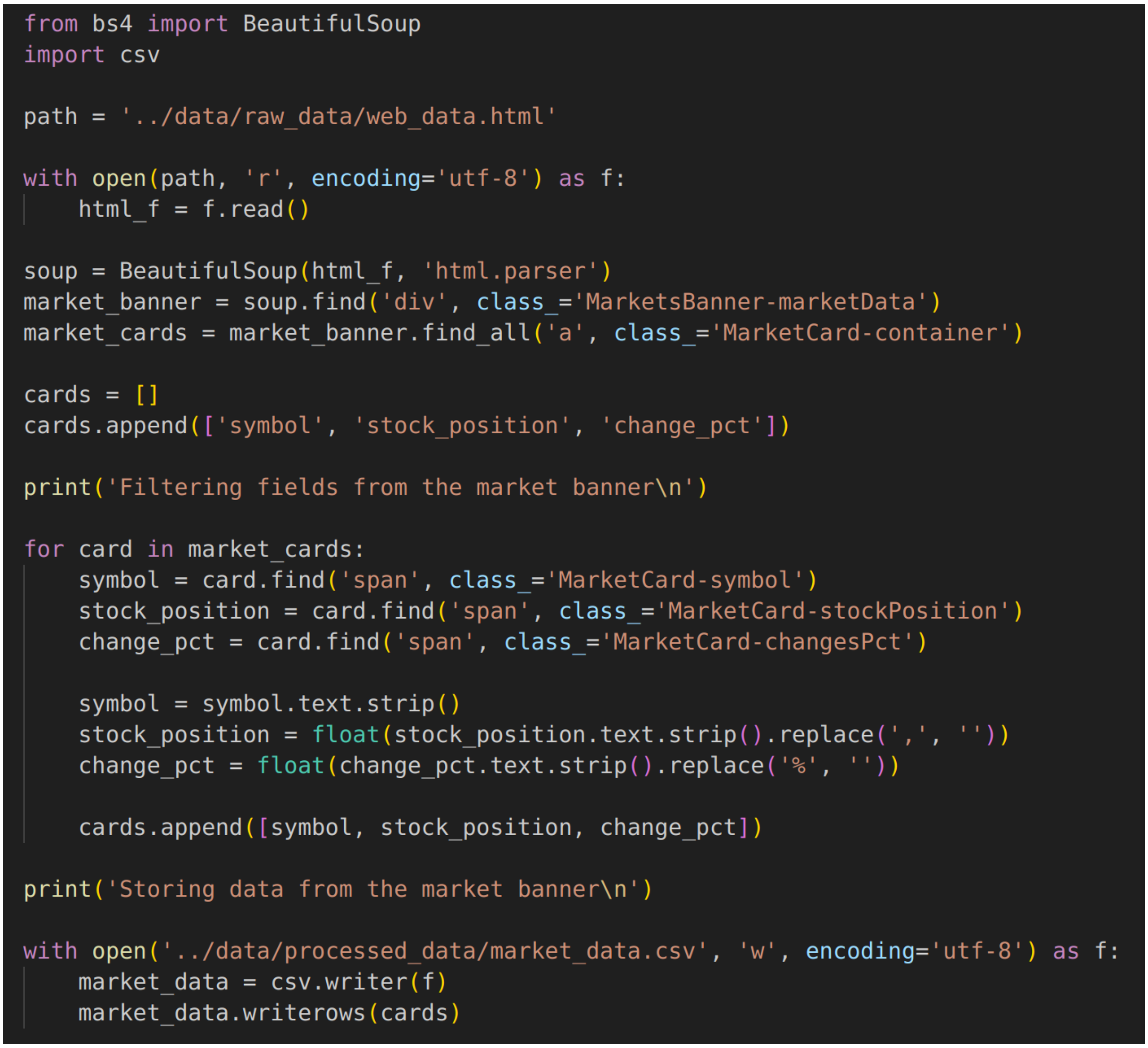
Since the Market Banner includes fields that are dynamically loaded, after some research, I used the **selenium** library in order to scrape data. After scraping all the tags from the Market Banner and Latest News section, I wrote them to **web_data.html** that is located in the **raw_data** within **data**.



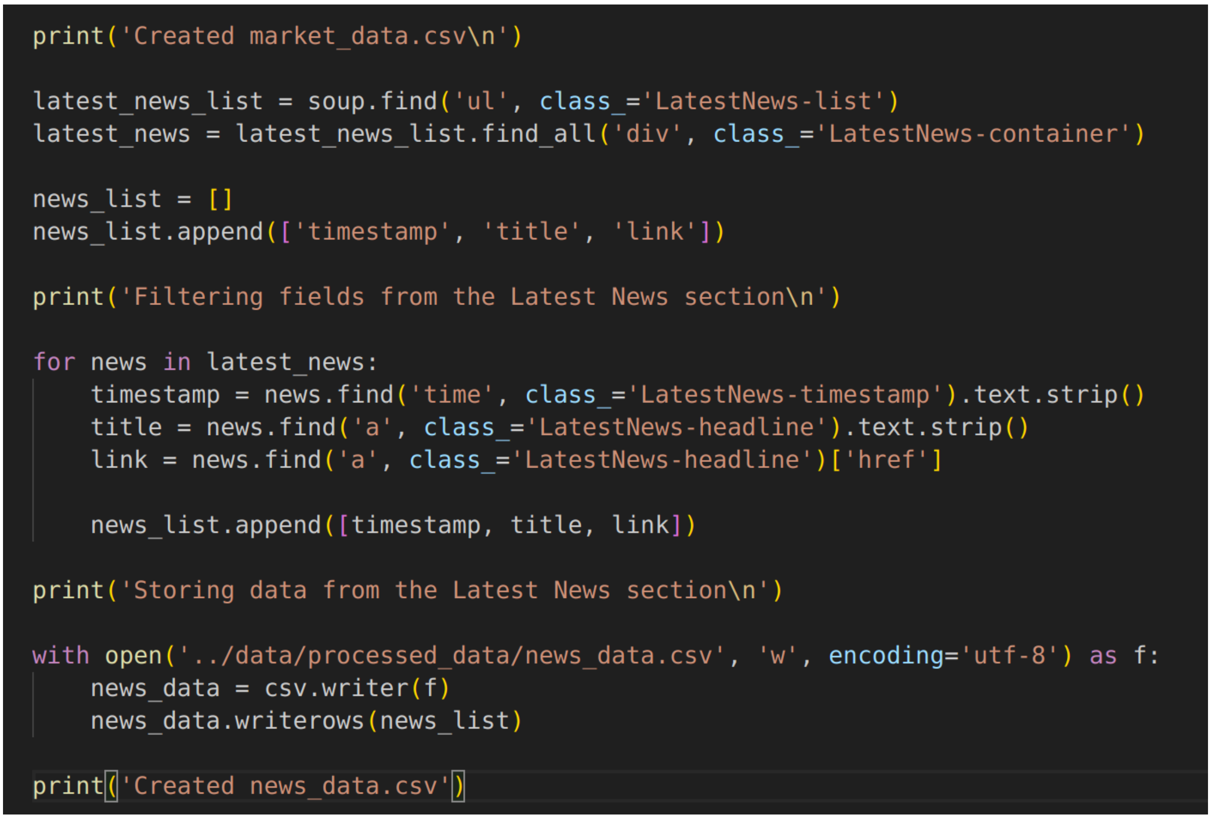
```
john-tran@john-tran-VMware-Virtual-Platform:~/Desktop/johntran_8897109598/script
s$ cd ..
john-tran@john-tran-VMware-Virtual-Platform:~/Desktop/johntran_8897109598$ cd da
ta/raw_data
john-tran@john-tran-VMware-Virtual-Platform:~/Desktop/johntran_8897109598/data/r
aw_data$ head -n 10 web_data.html
<div class="MarketsBanner-marketData" id="market-data-scroll-container">
 <a class="MarketCard-container MarketCard-down" href="//www.cnbc.com/quotes/.ST
OXX">
  <div class="MarketCard-row">
   <span class="MarketCard-symbol">
    STOXX600*
   </span>
   <span class="MarketCard-stockPosition">
    547.21
   </span>
  </div>
```

I printed the first 10 lines of **web_data.html**.

2.4)



```
from bs4 import BeautifulSoup
import csv

path = '../data/raw_data/web_data.html'

with open(path, 'r', encoding='utf-8') as f:
    html_f = f.read()

soup = BeautifulSoup(html_f, 'html.parser')
market_banner = soup.find('div', class_='MarketsBanner-marketData')
market_cards = market_banner.find_all('a', class_='MarketCard-container')

cards = []
cards.append(['symbol', 'stock_position', 'change_pct'])

print('Filtering fields from the market banner\n')

for card in market_cards:
    symbol = card.find('span', class_='MarketCard-symbol')
    stock_position = card.find('span', class_='MarketCard-stockPosition')
    change_pct = card.find('span', class_='MarketCard-changesPct')

    symbol = symbol.text.strip()
    stock_position = float(stock_position.text.strip().replace(',', ''))
    change_pct = float(change_pct.text.strip().replace('%', ''))

    cards.append([symbol, stock_position, change_pct])

print('Storing data from the market banner\n')

with open('../data/processed_data/market_data.csv', 'w', encoding='utf-8') as f:
    market_data = csv.writer(f)
    market_data.writerows(cards)
```



```
print('Created market_data.csv\n')

latest_news_list = soup.find('ul', class_='LatestNews-list')
latest_news = latest_news_list.find_all('div', class_='LatestNews-container')

news_list = []
news_list.append(['timestamp', 'title', 'link'])

print('Filtering fields from the Latest News section\n')

for news in latest_news:
    timestamp = news.find('time', class_='LatestNews-timestamp').text.strip()
    title = news.find('a', class_='LatestNews-headline').text.strip()
    link = news.find('a', class_='LatestNews-headline')['href']

    news_list.append([timestamp, title, link])

print('Storing data from the Latest News section\n')

with open('../data/processed_data/news_data.csv', 'w', encoding='utf-8') as f:
    news_data = csv.writer(f)
    news_data.writerows(news_list)

print('Created news_data.csv')
```

I read **web_data.html** and used **BeautifulSoup** to find the relevant information for both the Market Banner and the Latest News section. After cleaning the information retrieved, I add them to lists and store them under 2 csv files: **market_data.csv** and **news_data.csv**. I also added messages to be printed in the console.