

Github Link: <https://github.com/boming-xi/DSCI-560>

Youtube link: <https://www.youtube.com/watch?v=0QQDVMqXodg>

Name: Boming Xi

USC ID: 5879982708

1.3

```
boming@boming-VMware20-1:~$ python3 --version
Python 3.13.7
```

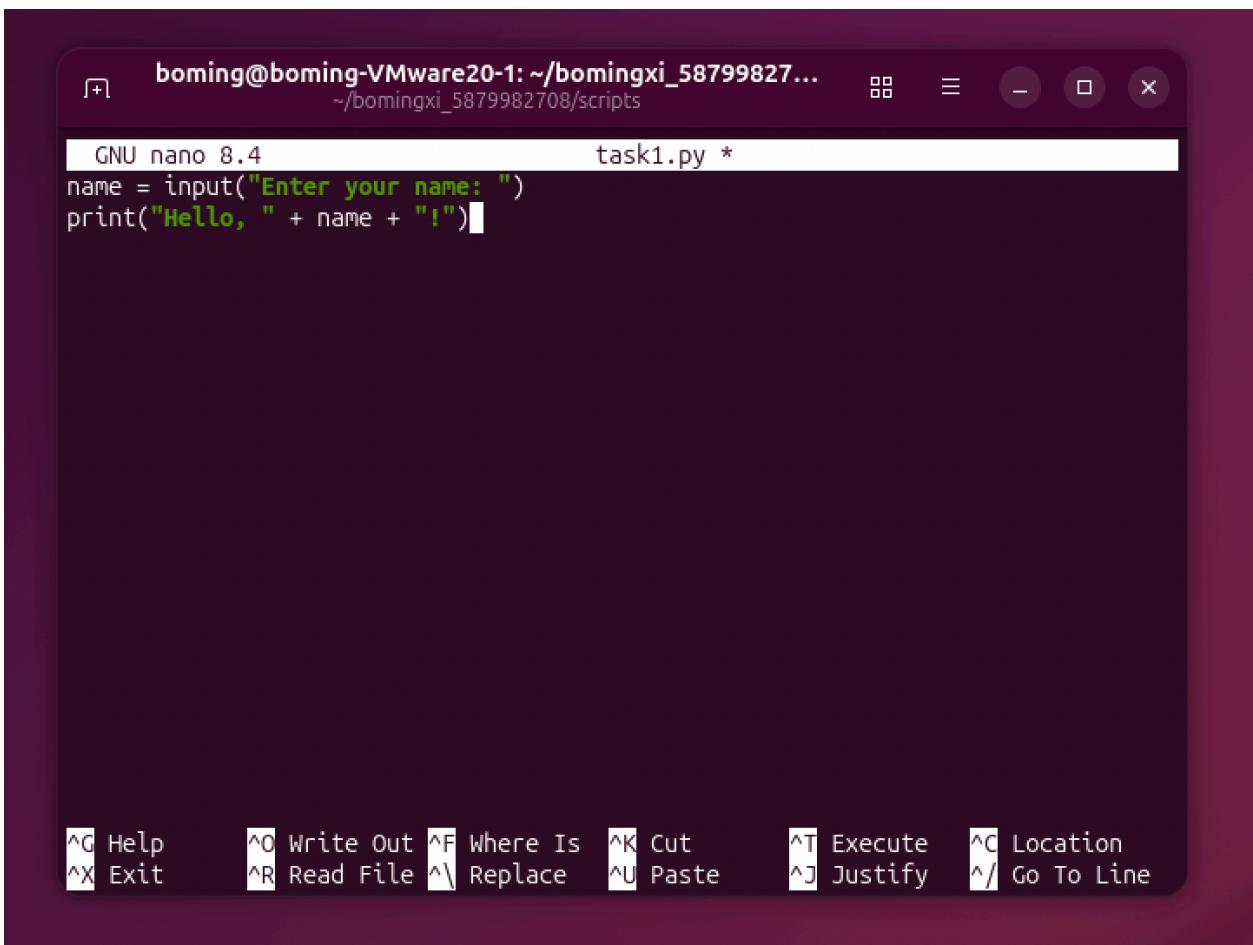
```
boming@boming-VMware20-1:~$ pip3 --version
pip 25.1.1 from /usr/lib/python3/dist-packages/pip (python 3.13)
```

2.1

```
boming@boming-VMware20-1:~$ mkdir bomingxi_5879982708
boming@boming-VMware20-1:~$ cd bomingxi_5879982708
boming@boming-VMware20-1:~/bomingxi_5879982708$ mkdir data scripts
boming@boming-VMware20-1:~/bomingxi_5879982708$ cd scripts
bash: cd: scripts: No such file or directory
boming@boming-VMware20-1:~/bomingxi_5879982708$ cd scripts
boming@boming-VMware20-1:~/bomingxi_5879982708/scripts$ touch task1.py
boming@boming-VMware20-1:~/bomingxi_5879982708/scripts$ ls
```

```
task1.py
```

## 2.2



A screenshot of a terminal window titled "boming@boming-VMware20-1: ~/bomingxi\_5879982708/scripts". The window shows the "task1.py" file being edited with the nano text editor. The code contains a single line of Python code that prompts the user for their name and prints a greeting message.

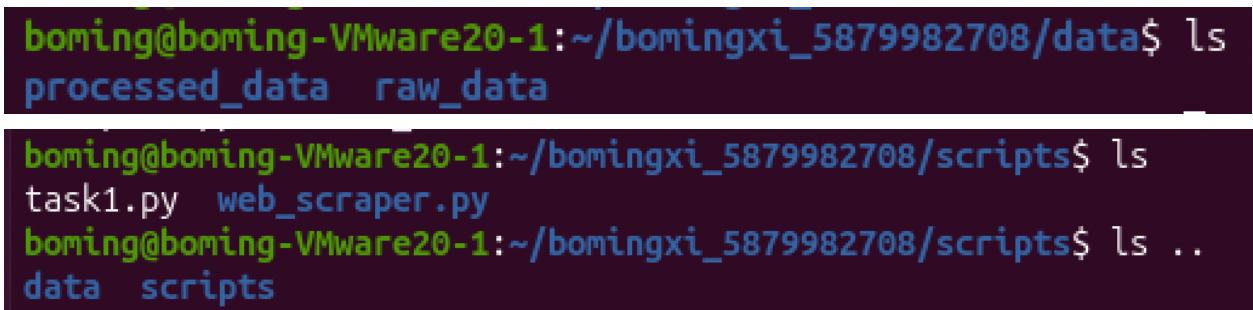
```
GNU nano 8.4          task1.py *
name = input("Enter your name: ")
print("Hello, " + name + "!")
```

The terminal also displays the keyboard shortcuts for the nano editor:

```
^G Help      ^O Write Out  ^F Where Is  ^K Cut      ^T Execute  ^C Location
^X Exit      ^R Read File  ^\ Replace   ^U Paste    ^J Justify  ^/ Go To Line
```

```
boming@boming-VMware20-1:~/bomingxi_5879982708/scripts$ nano task1.py
boming@boming-VMware20-1:~/bomingxi_5879982708/scripts$ python3 task1.py
Enter your name: Boming Xi
Hello, Boming Xi!
```

## 2.3



A screenshot of a terminal window showing directory listings. The first command lists the contents of the "data" directory, which contains two sub-directories: "processed\_data" and "raw\_data". The second command lists the contents of the current directory, which include the files "task1.py", "web\_scraper.py", and the "scripts" directory itself.

```
boming@boming-VMware20-1:~/bomingxi_5879982708/data$ ls
processed_data  raw_data

boming@boming-VMware20-1:~/bomingxi_5879982708/scripts$ ls
task1.py  web_scraper.py
boming@boming-VMware20-1:~/bomingxi_5879982708/scripts$ ls ..
data  scripts
```

Jan 17 16:34

```
boming@boming-VMware20-1: ~/bomingxi_5879982708/scripts — nano web_scraper.py
GNU nano 8.4
from playwright.sync_api import sync_playwright
from bs4 import BeautifulSoup
import os

URL = "https://www.cnbc.com/world/?region=world"
base_dir = os.path.dirname(os.path.abspath(__file__))
raw_dir = os.path.join(base_dir, "..", "data", "raw_data")
os.makedirs(raw_dir, exist_ok=True)

out_path = os.path.join(raw_dir, "web_data.html")

with sync_playwright() as p:
    browser = p.chromium.launch(headless=True)
    context = browser.new_context(
        user_agent = (
            "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
            " AppleWebKit/537.36 (KHTML, like Gecko)"
            " Chrome/91.0.4472.124 Safari/537.36"
        )
    )
    page = context.new_page()
    page.goto(URL, wait_until="domcontentloaded", timeout = 12000)
    page.mouse.wheel(0,1200)
    page.wait_for_selector("span.MarketCard-symbol", timeout = 12000)

    html = page.content()
    browser.close()

soup = BeautifulSoup(html, "html.parser")
pretty_html = soup.prettify()
with open(out_path, "w", encoding="utf-8") as f:
    f.write(pretty_html)

print("Saved HTML to:", os.path.abspath(out_path))
```

File Help Write Out Where Is Cut Execute Location Undo Set Mark To Bracket  
Exit Read File Replace Paste Justify Go To Line Redo Copy Where Was

Jan 17 16:35

```
boming@boming-VMware20-1: ~/bomingxi_5879982708/scripts — nano web_scraper.py
GNU nano 8.4
out_path = os.path.join(raw_dir, "web_data.html")

with sync_playwright() as p:
    browser = p.chromium.launch(headless=True)
    context = browser.new_context(
        user_agent = (
            "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
            " AppleWebKit/537.36 (KHTML, like Gecko)"
            " Chrome/91.0.4472.124 Safari/537.36"
        )
    )
    page = context.new_page()
    page.goto(URL, wait_until="domcontentloaded", timeout = 12000)
    page.mouse.wheel(0,1200)
    page.wait_for_selector("span.MarketCard-symbol", timeout = 12000)

    html = page.content()
    browser.close()

soup = BeautifulSoup(html, "html.parser")
pretty_html = soup.prettify()
with open(out_path, "w", encoding="utf-8") as f:
    f.write(pretty_html)

print("Saved HTML to:", os.path.abspath(out_path))

print("\nFirst 10 lines of web_data.html")
with open(out_path, "r", encoding="utf-8", errors="ignore") as f:
    for _ in range(10):
        line = f.readline()
        if not line:
            break
        print(line.rstrip("\n"))
```

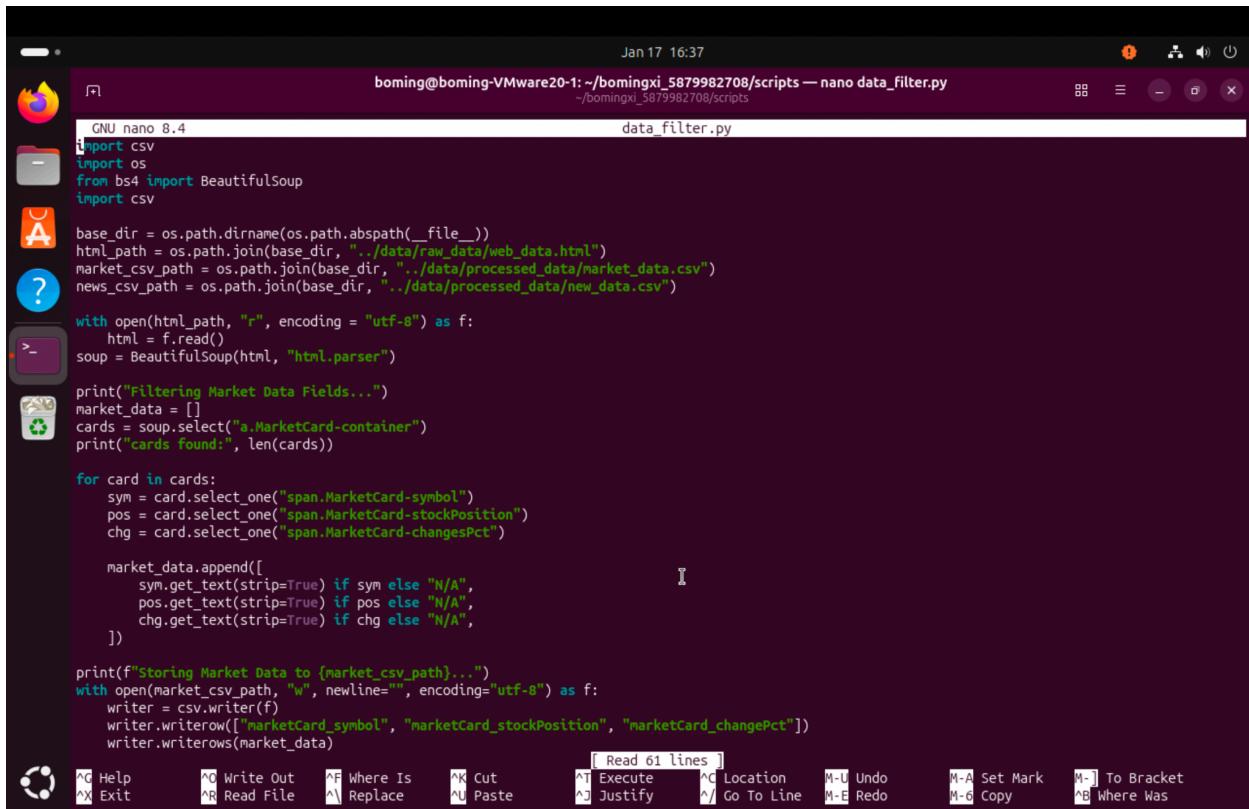
File Help Write Out Where Is Cut Execute Location Undo Set Mark To Bracket  
Exit Read File Replace Paste Justify Go To Line Redo Copy Where Was

```
(.venv) boming@boming-VMware20-1:~/bomingxi_5879982708/scripts$ python web_scraper.py
Saved HTML to: /home/boming/bomingxi_5879982708/data/raw_data/web_data.html

First 10 lines of web_data.html
<!DOCTYPE html>
<html itemscope="" itemtype="https://schema.org/WebPage" lang="en" prefix="og:https://ogp.me/ns#">
<head>
<meta content="A7vZI3v+Gz7JfuRolKNM4Aff6zaGuT7X0mf3wtoZTnKv6497cVMnhy03KDqX7kBz/q/iidW7srW31oQbBt4VhgoAACUeyJvcmlnaw4i0iJodHRwczovL3d3dy5nb29nbGUuY29t0jQ0MyIsImZlYXR1cmUi0iJEaXNhYmxLVGhpcmRQYXJ0eVN0b3JhZ2VQYXXJ0aXRpb25pbmczIiwiZXhwaXJ5IjoxNzU30TgwODAwLCJpc1N1YmRvbWFpbI6dHJ1ZSwiaXNuGlyZFBhcnR5Ijp0cnvlfq==" http-equiv="origin-trial"/>
<noscript>
</noscript>
<script async="" src="https://cdn.cxense.com/cx.cce.js" type="text/javascript">
</script>
<script async="" data-jsonpid="" src="https://cdn-gl.imrworldwide.com/novms/js/2/nlsSDK600.bundle.min.js">
</script>
```

Write a Python web scraper (web\_scraper.py) that uses **Playwright** to open CNBC's World page in a real headless Chromium browser, waits for dynamic market elements to load, then grabs the fully rendered HTML. Then use **BeautifulSoup** to prettify/format that HTML, save it to data/raw\_data/web\_data.html, and print the saved path and the first 10 lines.

## 2.4



```
Jan 17 16:37
boming@boming-VMware20-1: ~/bomingxi_5879982708/scripts -- nano data_filter.py
GNU nano 8.4
import csv
import os
from bs4 import BeautifulSoup
import csv

base_dir = os.path.dirname(os.path.abspath(__file__))
html_path = os.path.join(base_dir, "../data/raw_data/web_data.html")
market_csv_path = os.path.join(base_dir, "../data/processed_data/market_data.csv")
news_csv_path = os.path.join(base_dir, "../data/processed_data/new_data.csv")

with open(html_path, "r", encoding = "utf-8") as f:
    html = f.read()
soup = BeautifulSoup(html, "html.parser")

print("Filtering Market Data Fields...")
market_data = []
cards = soup.select("a.MarketCard-container")
print("cards found:", len(cards))

for card in cards:
    sym = card.select_one("span.MarketCard-symbol")
    pos = card.select_one("span.MarketCard-stockPosition")
    chg = card.select_one("span.MarketCard-changesPct")

    market_data.append([
        sym.get_text(strip=True) if sym else "N/A",
        pos.get_text(strip=True) if pos else "N/A",
        chg.get_text(strip=True) if chg else "N/A",
    ])

print(f"Storing Market Data to {market_csv_path}...")
with open(market_csv_path, "w", newline="", encoding="utf-8") as f:
    writer = csv.writer(f)
    writer.writerow(["MarketCard_symbol", "MarketCard_stockPosition", "MarketCard_changePct"])
    writer.writerows(market_data)
```

The terminal shows the script being run and the resulting output. The status bar at the bottom indicates "Read 61 lines".

```

Jan 17 16:37
boming@boming-VMware20-1: ~/bomingxi_5879982708/scripts -- nano data_filter.py
GNU nano 8.4
    chg.get_text(strip=True) if chg else "N/A",
    ])

print(f"Storing Market Data to {market_csv_path}...")
with open(market_csv_path, "w", newline="", encoding="utf-8") as f:
    writer = csv.writer(f)
    writer.writerow(["MarketCard_symbol", "MarketCard_stockPosition", "MarketCard_changePct"])
    writer.writerows(market_data)
print("Market CSV created")

print("Filtering Latest News Fields...")
news_data = []
news_items = soup.select("li.LatestNews-item")
print(f"news_items found: {len(news_items)}")

for item in news_items:
    time_tag = item.select_one("time.LatestNews-timestamp")
    timestamp = time_tag.get_text(strip=True) if time_tag else "N/A"

    title_tag = item.select_one("a.LatestNews-headline")
    title = title_tag.get_text(strip=True) if title_tag else "N/A"
    link = title_tag.get("href") if title_tag else "N/A"

    if link and link.startswith("/"):
        link = "https://www.cnbc.com" + link

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

print(f"Storing News Data to {news_csv_path}...")
with open(news_csv_path, "w", newline="", encoding="utf-8") as f:
    writer = csv.writer(f)
    writer.writerow(["LatestNews-timestamp", "title", "link"])
    writer.writerows(news_data)
print("News CSV created")

```

(.venv) boming@boming-VMware20-1:~/bomingxi\_5879982708/scripts\$ python data\_filter.py

Filtering Market Data Fields...

cards found: 5

Storing Market Data to /home/boming/bomingxi\_5879982708/scripts/../data/processed\_data/market\_data.csv...

Market CSV created

Filtering Latest News Fields...

news\_items found: 30

Storing News Data to /home/boming/bomingxi\_5879982708/scripts/../data/processed\_data/news\_data.csv...

News CSV created

Write and run `data_filter.py` to **parse the saved CNBC HTML (data/raw\_data/web\_data.html)** with **BeautifulSoup**, extract two datasets—**market card fields** (symbol, stock position, percent change) and **latest news fields** (timestamp, headline, link)—handle missing values with "N/A" and convert relative links to full CNBC URLs, then **export the cleaned results into CSV files:** `data/processed_data/market_data.csv` and `data/processed_data/news_data.csv`.

Open new\_data.csv  
~/bomingxi\_5879982708/data/processed\_data

market\_data.csv new\_data.csv data\_filter.py task1.py

```
LatestNews-timestamp,title,link
5 Hours Ago,Week in review: Stocks battled a flood of news and we booked some profits,https://www.cnbc.com/2026/01/17/week-in-review-stocks-battled-a-flood-of-news-and-we-booked-some-profits.html
6 Hours Ago,Trump threatens to sue JPMorgan Chase for 'debanking' him,https://www.cnbc.com/2026/01/17/trump-jpmorgan-chase-debanking.html
8 Hours Ago,Trump: NATO members to face tariffs up to 25% until a Greenland deal is struck,https://www.cnbc.com/2026/01/17/trump-greenland-tariffs-nato.html
9 Hours Ago,"Led by Texas, states race to prove they can put bitcoin on public balance sheet",https://www.cnbc.com/2026/01/17/texas-us-states-budgets-bitcoin-crypto-strategic-reserve.html
10 Hours Ago,Unshaken: Why Brazilian stocks have looked past the Venezuela attack,https://www.cnbc.com/2026/01/17/unshaken-why-brazilian-stocks-have-looked-past-the-venezuela-attack.html
10 Hours Ago,Bestselling author: How to create better habits without relying on discipline,https://www.cnbc.com/2026/01/17/james-clear-how-to-create-better-habits-without-relying-on-discipline.html
10 Hours Ago,"Warren Buffett: To maximize your potential, ask yourself this question",https://www.cnbc.com/2026/01/17/warren-buffett-to-maximize-your-potential-ask-yourself-this-question.html
11 Hours Ago,"Buy these five stocks ahead of earnings, Bank of America says",https://www.cnbc.com/2026/01/17/stocks-to-buy-ahead-of-earnings-bank-of-america-says.html
11 Hours Ago,This week's most overbought names include Darden Restaurants and Target,https://www.cnbc.com/2026/01/17/this-weeks-most-overbought-names-include-darden-restaurants-and-target.html
11 Hours Ago,"Buffett on parenting, giving up horse betting and why he stopped talking politics",https://www.cnbc.com/2026/01/17/warren-buffett-on-parenting-horse-betting-and-why-he-stopped-talking-politics.html
```

Open market\_data.csv  
~/bomingxi\_5879982708/data/processed\_data

market\_data.csv new\_data.csv data\_filter.py task1.py

```
marketCard_symbol,marketCard_stockPosition,marketCard_changePct
STOXX600*,614.38,-0.03%
DAX*, "25,297.13",-0.22%
FTSE*, "10,235.29",-0.04%
CAC*, "8,258.94",-0.65%
FTSE MIB*, "45,799.69",-0.11%
```