# Web Scraping Task

**Task Description:**

You are tasked to perform web scraping on a provided HTML page that contains different types of elements. The goal is to extract specific data from the page and process it into structured formats such as CSV or JSON.

---

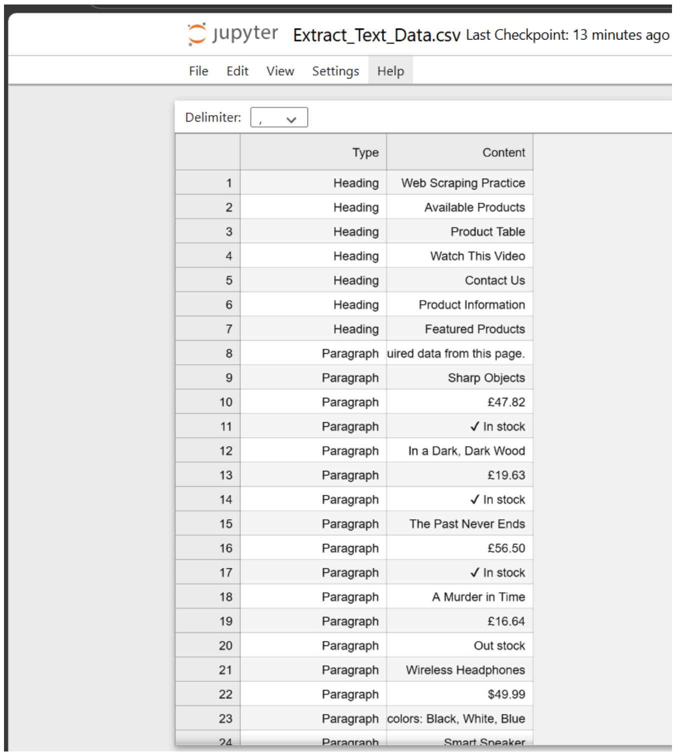
## https://baraasalout.github.io/test.html

## Steps to Complete the Task

**pip install beautifulsoup4**

### 1. Extract Text Data:

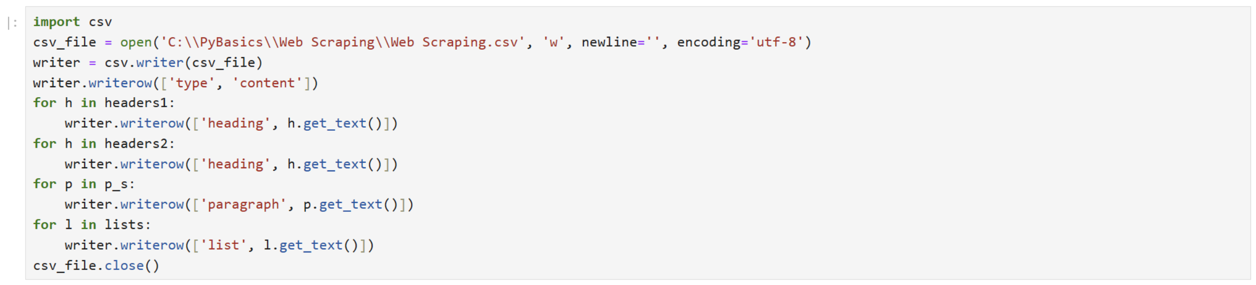
- Extract all headings (<h1>, <h2>).
- Extract all text content inside `<p>` and `<li>` tags.
- Save this data into a **Extract_Text_Data.CSV** file.
  https://www.pythontutorial.net/python-basics/python-write-csv-file/



Extract_Text_Data.csv Last Checkpoint: 13 minutes ago

Delimiter: ,

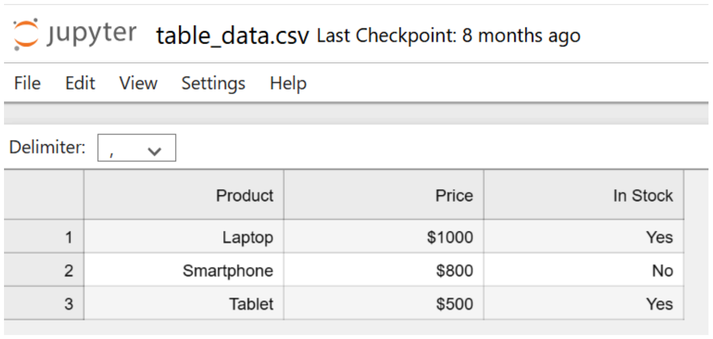
| | Type | Content |
|---|---|---|
| 1 | Heading | Web Scraping Practice |
| 2 | Heading | Available Products |
| 3 | Heading | Product Table |
| 4 | Heading | Watch This Video |
| 5 | Heading | Contact Us |
| 6 | Heading | Product Information |
| 7 | Heading | Featured Products |
| 8 | Paragraph | uired data from this page. |
| 9 | Paragraph | Sharp Objects |
| 10 | Paragraph | £47.82 |
| 11 | Paragraph | ✓ In stock |
| 12 | Paragraph | In a Dark, Dark Wood |
| 13 | Paragraph | £19.63 |
| 14 | Paragraph | ✓ In stock |
| 15 | Paragraph | The Past Never Ends |
| 16 | Paragraph | £56.50 |
| 17 | Paragraph | ✓ In stock |
| 18 | Paragraph | A Murder in Time |
| 19 | Paragraph | £16.64 |
| 20 | Paragraph | Out stock |
| 21 | Paragraph | Wireless Headphones |
| 22 | Paragraph | $49.99 |
| 23 | Paragraph | colors: Black, White, Blue |
| 24 | Paragraph | Smart Speaker |



```
import csv
csv_file = open('C:\\PyBasics\\Web Scraping\\Web Scraping.csv', 'w', newline='', encoding='utf-8')
writer = csv.writer(csv_file)
writer.writerow(['type', 'content'])
for h in headers1:
    writer.writerow(['heading', h.get_text()])
for h in headers2:
    writer.writerow(['heading', h.get_text()])
for p in p_s:
    writer.writerow(['paragraph', p.get_text()])
for l in lists:
    writer.writerow(['list', l.get_text()])
csv_file.close()
```

## 2. Extract Table Data:

- Extract data from the table, including:
    - Product Name.
    - Price.
    - Stock Status.
- Save this data into a **Extract_Table_Data.CSV** file.
- https://www.pythontutorial.net/python-basics/python-write-csv-file/
- https://www.geeksforgeeks.org/pandas/saving-a-pandas-dataframe-as-a-csv/



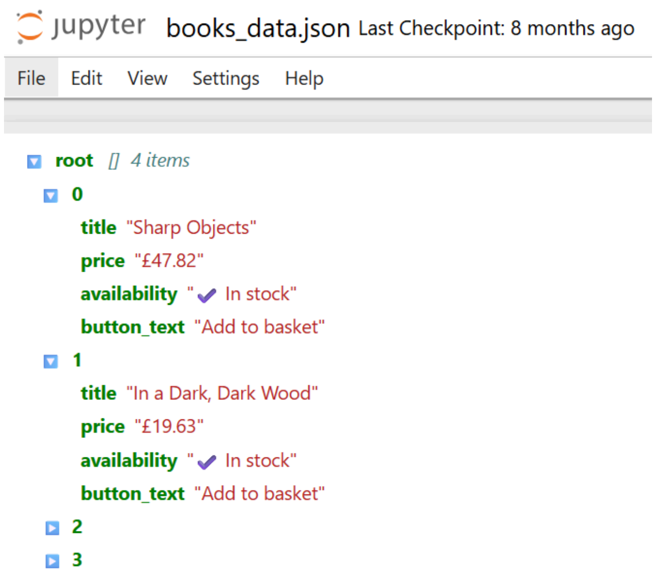
table_data.csv Last Checkpoint: 8 months ago

Delimiter: ,

| | Product | Price | In Stock |
|---|---|---|---|
| 1 | Laptop | $1000 | Yes |
| 2 | Smartphone | $800 | No |
| 3 | Tablet | $500 | Yes |

**3. Extract Product Information (Cards Section):**

- Extract data from the book cards at the bottom of the page, including:
    - Book Title.
    - Price.
    - Stock Availability.
    - Button text (e.g., "Add to basket").
- Save the data into a **books_data.**JSON file.
- https://www.geeksforgeeks.org/how-to-convert-python-dictionary-to-json/



```
root [] 4 items
  0
    title "Sharp Objects"
    price "£47.82"
    availability "✓ In stock"
    button_text "Add to basket"
  1
    title "In a Dark, Dark Wood"
    price "£19.63"
    availability "✓ In stock"
    button_text "Add to basket"
  2
  3
```

**4. Extract Form Details:**

- Extract all input fields from the form, including:
    - Field name (e.g., `username`, `password`).
    - Input type (e.g., `text`, `password`, `checkbox`, etc.).
    - Default values, if any.
    - Save the data into a JSON file.
    - https://www.geeksforgeeks.org/how-to-convert-python-dictionary-to-json/

**5. Extract Links and Multimedia:**

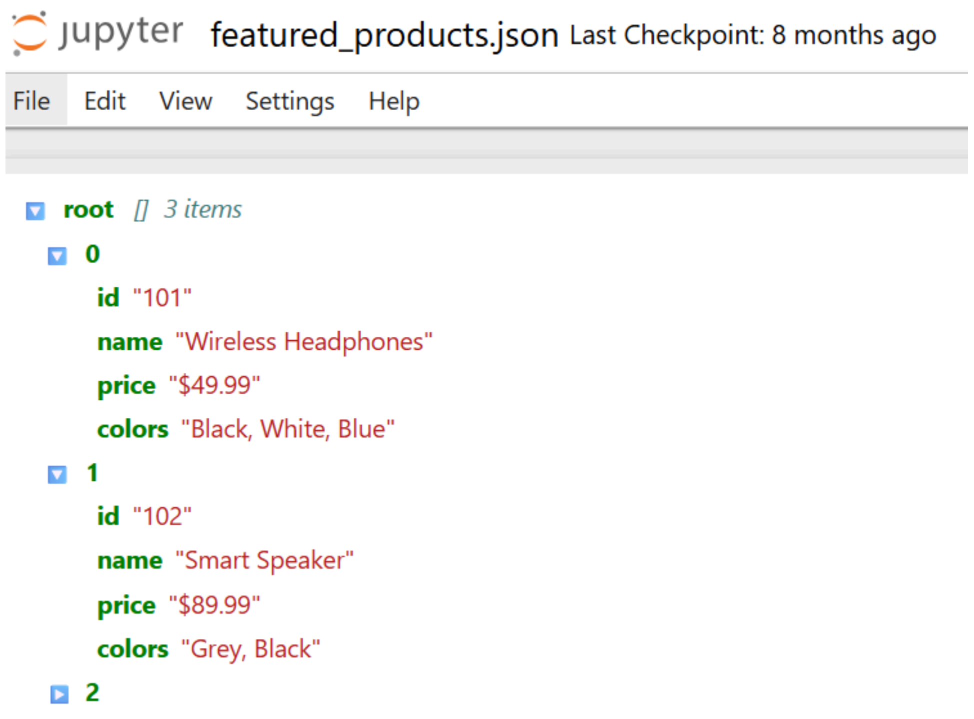
- Extract the video link from the `<iframe>` tag.
- Save the data into a JSON file.
- https://www.geeksforgeeks.org/how-to-convert-python-dictionary-to-json/

**6. Scraping Challenge:**

Students must write a script to extract data from the **Featured Products** section with the following requirements:

- Product Name: Located within `<span class="name">`.
- Hidden Price: Located within `<span class="price">`, which has `style="display: none;"`.
- Available Colors: Located within `<span class="colors">`.
- Product ID: The value stored in the `data-id` attribute.
- **Example Output:**

[ {'id': '101', 'name': 'Wireless Headphones', 'price': '$49.99', 'colors': 'Black, White, Blue'}, …, ]



```
root [] 3 items
  0
    id "101"
    name "Wireless Headphones"
    price "$49.99"
    colors "Black, White, Blue"
  1
    id "102"
    name "Smart Speaker"
    price "$89.99"
    colors "Grey, Black"
  2
```

]

## Deliverables

1. **CSV Files**
    - `Extract_Text_Data.csv`
    - `Extract_Table_Data.csv`
2. **JSON Files**
    - `Product_Information.json`
    - `Form_Details.json`
    - `Multimedia.json`
    - `Featured_Products.json`
3. **Python Script**
    - Well-commented scraping script.
4. **Documentation**
    - Short explanation of approach, libraries used (e.g., BeautifulSoup, requests), and challenges faced.
5. **GitHub Repo**
    - Upload all files & script, then **submit a repo link.**