Day 3

---

### Web Scraping Script for Extracting Tables from a Webpage

### Overview

This script scrapes all tables from the given webpage, processes missing values, splits multiple values into separate rows, and saves the cleaned data into a single CSV file.

---

## Prerequisites

### Install Required Libraries

Ensure you have the necessary Python libraries installed before running the script:

```
pip install selenium pandas webdriver-manager
```

### Web Driver Requirements

- The script **automatically installs** the latest **ChromeDriver** using `webdriver_manager`.
- **Google Chrome** should be installed on your system.

---

## Script Explanation

### 1 Importing Required Libraries

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
from webdriver_manager.chrome import ChromeDriverManager
import pandas as pd
import time
```

- `selenium`: Automates web browser interactions.
- `webdriver_manager`: Ensures the correct version of ChromeDriver is used.
- `pandas`: Handles data processing and exporting to CSV.
- `time`: Introduces delays to ensure the page loads completely.

---

### 2 Setting Up WebDriver

```
options = webdriver.ChromeOptions()
options.add_argument("--headless")
options.add_argument("--window-size=1920,1080")

service = Service(ChromeDriverManager().install())
```

```
driver = webdriver.Chrome(service=service, options=options)
```

- **Headless Mode**: Runs Chrome in the background without opening a window.
- **Window Size**: Ensures consistent rendering for scraping.
- **WebDriver Initialization**: Uses `ChromeDriverManager` to install the correct version.

---

### 3 Navigating to the Webpage

```
url = "https://versionsof.net/core/8.0/8.0.0/"
driver.get(url)
time.sleep(2)
```

- Opens the webpage containing the tables.
- Introduces a **2-second delay** to ensure complete page loading.

---

### 4□ Extracting Tables

```
tables = driver.find_elements(By.XPATH, "//table")
all_data = []
```

- Locates **all tables** on the webpage using XPath.
- Initializes an empty list to store table data.

---

### 5□ Iterating Through Each Table

```
for table in tables:
    rows = table.find_elements(By.XPATH, ".//tr")
    table_data = []
```

- Finds **all rows** (`<tr>`) inside each table.
- Creates a list (`table_data`) to store row-wise extracted data.

---

### 6□ Extracting Rows & Handling Missing Values

```
for row in rows:
    cells = row.find_elements(By.XPATH, ".//td | .//th")
    cell_texts = [cell.text.strip() if cell.text.strip() else "-"
for cell in cells]
```

- Extracts **header (`<th>`) and data (`<td>`)** cells from each row.
- **Replaces missing (null) values** with `"-"` to ensure completeness.

---

### 7 Splitting Multiple Values into Separate Rows

```
max_splits = max(len(cell.split("\n")) for cell in cell_texts)
split_rows = [cell.split("\n") + ["-"] * (max_splits -
len(cell.split("\n"))) for cell in cell_texts]

for i in range(max_splits):
    table_data.append([row[i] for row in split_rows])
```

- Checks if a **cell contains multiple values** (separated by new lines `\n`).
- **Splits them into separate rows** while keeping other column values unchanged.

---

### 8□ Storing Data & Exporting to CSV

```
if all_data:
    final_df = pd.concat([pd.DataFrame(data[1:], columns=data[0])
for data in all_data], ignore_index=True)
    final_df.to_csv("scraped_tables.csv", index=False)
    print("Scraping successful! Data saved to scraped_tables.csv")
else:
    print("No tables found!")
```

- **Combines all extracted tables** into a single DataFrame.
- **Exports the cleaned data** into `scraped_tables.csv`.

---

### 9□ Closing the WebDriver

```
driver.quit()
```

- Ensures the **browser instance is properly closed** after execution.

---

### CSV Output Example

**Before Scraping (Table Example)**

| Feature | Version | Status |
|---|---|---|
| Feature A | 8.0.1\n8.0.2 | Active |
| Feature B | 8.0.3 | Deprecated |

**After Processing (CSV Output)**

| Feature | Version | Status |
|---|---|---|
| Feature A | 8.0.1 | Active |
| Feature A | 8.0.2 | Active |
| Feature B | 8.0.3 | Deprecated |

---

### Error Handling & Debugging

#### 1 Common Issues & Fixes

| Issue | Cause | Solution |
|---|---|---|
| `NoSuchElementException` | Table not found | Check XPath or add delay (`time.sleep(2)`). |
| `WebDriverException` | ChromeDriver not installed | Run `pip install webdriver-manager`. |
| Empty CSV | No data extracted | Verify the table exists on the webpage. |

---

### Conclusion

This script efficiently extracts **all tables** from a webpage, cleans missing values, and structures data into a **single CSV file**, making it **ready for analysis**. 🚀