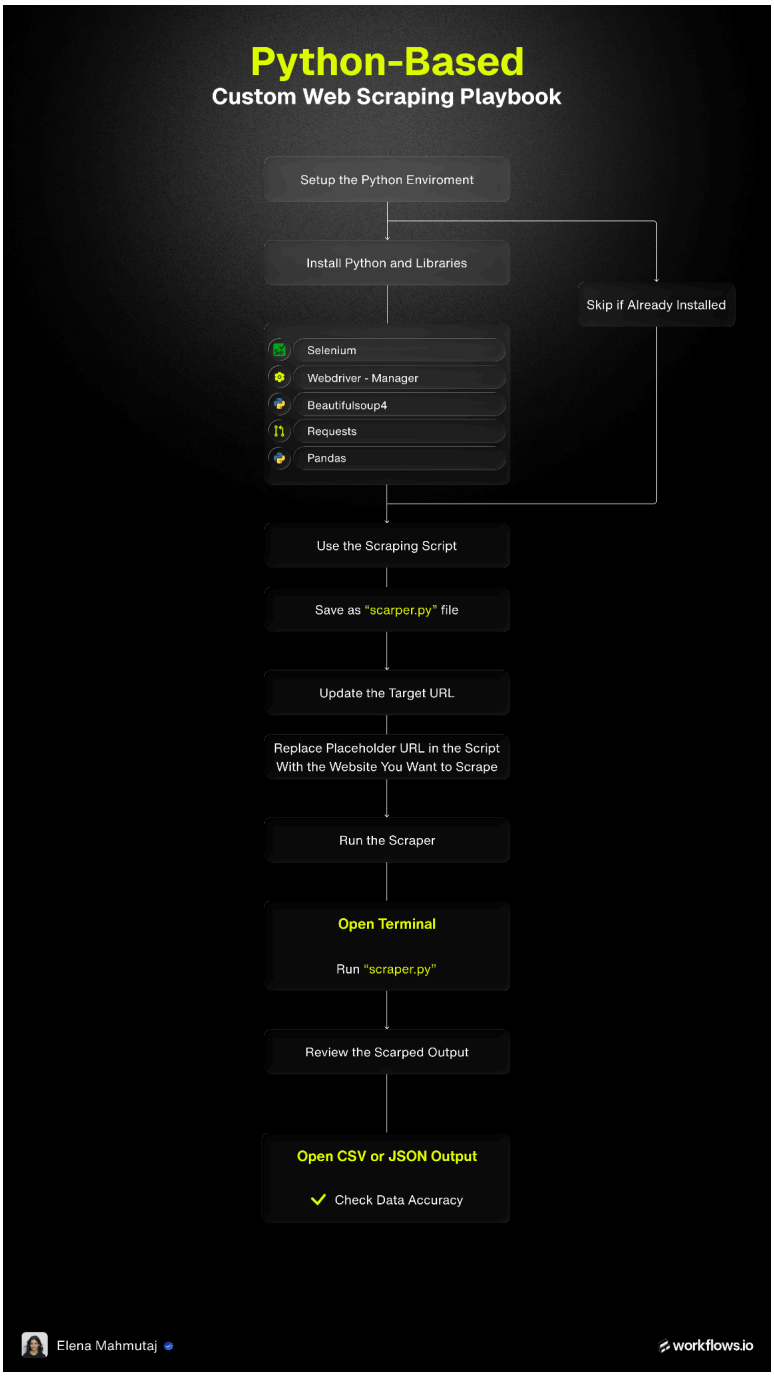
Python-Based Custom Web Scraping Playbook



How it works?

If the website you want to scrape is publicly accessible, you can write a Python script and run it locally using an IDE like VS Code, Cursor or directly from the command line.

Prerequisites

Before running the script, make sure your environment is properly set up.

1. Install Python
    a. Download Python from: https://www.python.org/downloads/
    b. During installation, check “Add Python to PATH”
    c. Verify the installation using this command in the terminal:
    python -- version
    If Python is installed correctly, you’ll see the version number.
2. Install Required Libraries
    a. Open Command Line (Windows) or Terminal (Mac / Linux) and run:

```
pip install selenium
pip install webdriver-manager
pip install beautifulsoup4
pip install requests
pip install pandas
```

These libraries allow you to:

- Control a browser (Selenium)
- Parse HTML (BeautifulSoup)
- Make HTTP requests
- Export data to CSV or JSON

3. Install Chrome WebDriver

Since this script automates Chrome, you need ChromeDriver.

You have two options:

- Option A (Recommended): Let Selenium manage it automatically
- Option B: Download manually from https://chromedriver.chromium.org/

How to Use the Script

Step 1: Save the Script

Copy the example code shared below and save it as scraper.py

Step 2: Modify the Target URL

Inside the script, replace the URL with your target website:

```
url = "YOUR_TARGET_WEBSITE_URL_HERE"
```

Step 3: Run the Script

From the folder where the file is saved:

```
python scraper.py
```

Step 4: Access the Output Files

Once the script finishes, you'll find:

- walmart_brands.json
- walmart_brands.csv

Both files will be saved in the same directory as the script.

Understanding the Example Code

This example scrapes brand names and URLs from Walmart's brand directory.

You can use it for other websites by modifying:

- URL: the target websites
- Selectors: XPath or CSS Selectors
- Data fields: names, links, emails, phone numbers

If you're unsure about selectors, inspect the page in Chrome or ask Claude to help you find them on the website.

Example Python Scraper Code

Code Link:
https://github.com/Workflowsio/workflows/blob/main/playbooks/python-scraping-script.py

Troubleshooting Common Issues

1. **Python is not recognised as a command**
- Python was not added to your system PATH during installation
- Reinstall Python and make sure to select "Add Python to PATH"
- Verify by running: `python --version`

2. **No module named selenium**
- Selenium is not installed in your environment
- Run:
  `pip install selenium`
- If it still fails, try:
  `pip3 install selenium`

3. **Chrome browser opens, but nothing happens**
- The website may have anti-bot or bot-detection mechanisms
- Add delays using `time.sleep()` between actions
- Use explicit waits instead of fixed sleeps where possible
- Ask Claude to update the script with basic stealth or human-like behaviour

4. **Script crashes with timeout errors**
- Increase wait or timeout values in the script
- Make sure your internet connection is stable
- The website's DOM structure may have changed, requiring selector updates