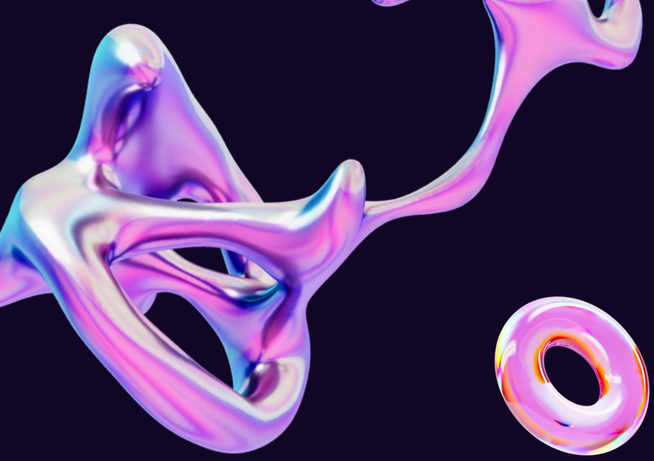
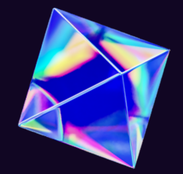
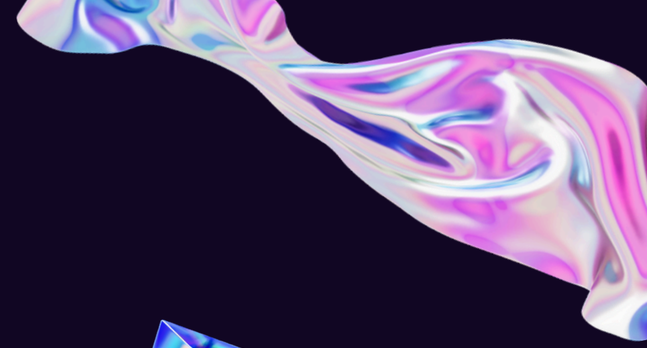
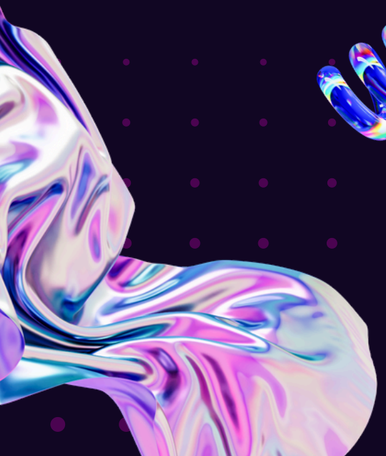
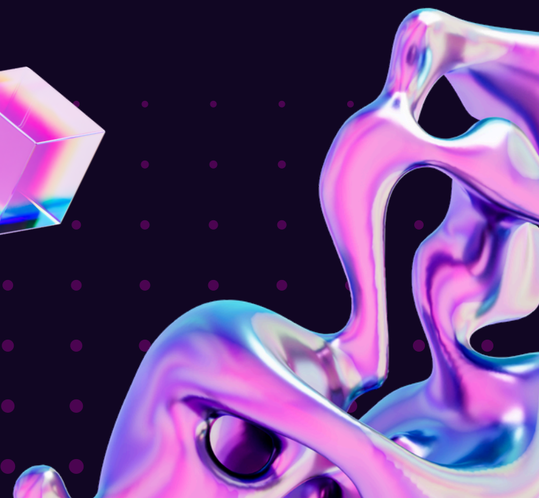


WEB SCRAPING

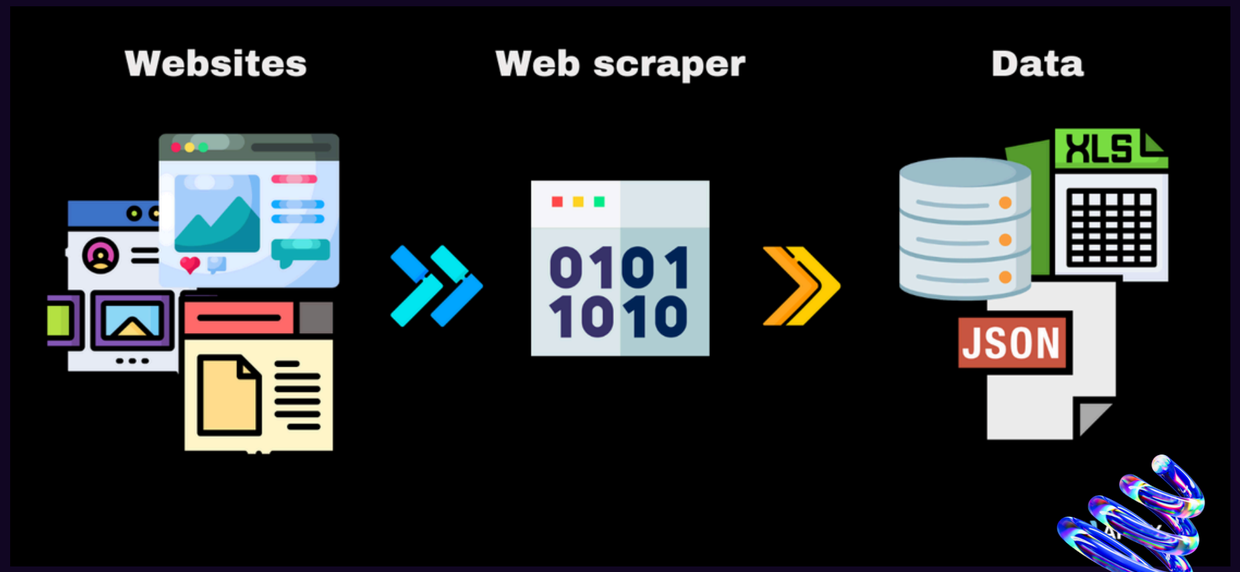
ALWIN SAJAN



WHAT IS WEB SCRAPING

Web scraping is the process of using bots to extract content and data from a website. Unlike screen scraping, which only copies pixels displayed onscreen, web scraping extracts underlying HTML code and, with it, data stored in a database. The scraper can then replicate entire website content elsewhere.



WHY WEB SCRAPING

Automate data collection, gather insights, and perform analysis efficiently.

USE CASES

- MARKET RESEARCH
- PRICE COMPARISON
- SENTIMENT ANALYSIS
- JOB AGGREGATION

TOOLS FOR WEB SCRAPING

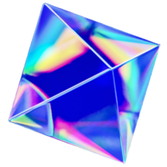
PYTHON LIBRARIES FOR WEB SCRAPING:

BEAUTIFUL SOUP

SCRAPY

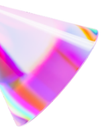
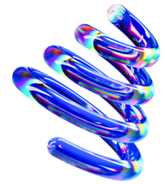
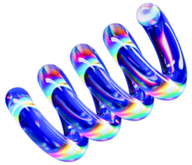
SELENIUM



SELENIUM

Handles JavaScript-heavy websites.
Mimics real user interactions (clicks, form submissions, etc.).


WHAT IS SELENIUM?

A powerful browser automation tool.

HOW IT WORKS:

01

OPENS A WEB BROWSER (E.G., CHROME, FIREFOX).

02

INTERACTS WITH WEB ELEMENTS (BUTTONS, FIELDS, LINKS).

03

EXTRACTS DATA FROM THE PAGE.

PREREQUISITES:

INSTALL SELENIUM: `pip install selenium`

DOWNLOAD WEBDRIVER (E.G., CHROMEDRIVER).

SETTING UP SELENIUM IN PYTHON

STEPS TO START:

1. Import the library:
```
from selenium import webdriver
```

2. Initialize the WebDriver:
```
driver = webdriver.Chrome()
driver.get("https://example.com")
```

3. Locate elements using methods like:

- find_element_by_id
- find_element_by_name
- find_element_by_xpath

```
from selenium import webdriver

driver = webdriver.Chrome()
driver.get("https://example.com")
print(driver.title)
driver.quit()
```


LOCATING WEB ELEMENTS IN SELENIUM
OVER TO WORKSHOP

BEST PRACTICES AND ETHICS


End of Presentation
ANY QUESTIONS?
memecreator.org

THANK YOU