Web Scraping with Python: Techniques and Ethics

Presented by Tanya Khanna

WORKSHOP DETAILS

Email: tk759@scarletmail.rutgers.edu

Workshop Materials:

- **Github Link:** https://github.com/Tanya-Khanna/DataScienceWorkshop_Fall-2024_NBL_
- Spring 2024 Workshops: Link

WORKSHOP SCHEDULE

Fall 2024 workshops calender

Introduction to Python Programming	**September 9, 2024; 4 - 5:30 PM**
Advanced Python Programming	**September 16, 2024; 4 - 5:30 PM**
Web Scraping with Python	**September 23, 2024; 4 - 5:30 PM**
Mastering Data Analysis: Pandas and Numpy	**September 30, 2024; 4 - 5:30 PM**
Data Management with Python: SQL and NoSQL	**October 7, 2024; 4 - 5:30 PM**
Python for Visualization and Exploration	**October 14, 2024; 4 - 5:30 PM**
Introduction to Machine Learning: Supervised Learning	**October 21, 2024; 4 - 5:30 PM**
Introduction to Machine Learning: Unsupervised Learning	**October 28, 2024; 4 - 5:30 PM**
Deploying Machine Learning Models	**November 4, 2024; 4 - 5:30 PM**
Ethical AI and Responsible Data Science	**November 11, 2024; 4 - 5:30 PM**

TABLE OF CONTENTS

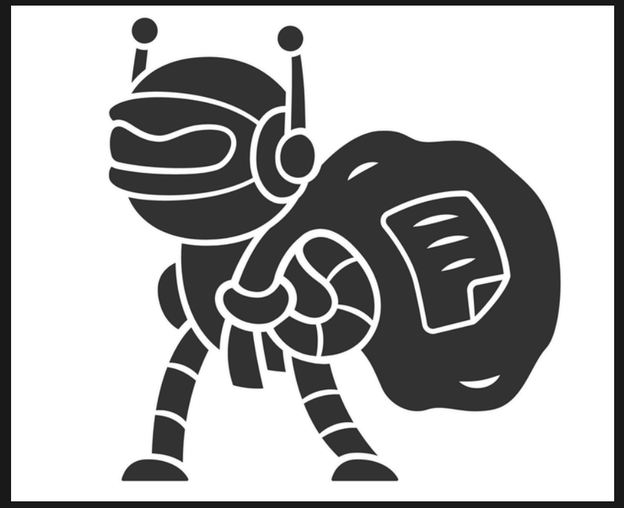
What is Web Scraping?

Web scraping is the process of automatically extracting data from websites using software tools. It works by "reading" the web pages, just like you do in a browser, but instead of looking at the content, the software saves the information in an organized way, like in a spreadsheet or database, for you to analyze later. The content is typically pulled from the HTML of web pages and stored in a structured format like CSV, JSON, or databases for further analysis.

Web scraping sends a request to a website (similar to when you click on a link). Once the website sends back the page, the software looks through the code (HTML) to find the data you want, like prices or text. Then, it pulls that information and organizes it so you can use it.

How Web Scraping Works

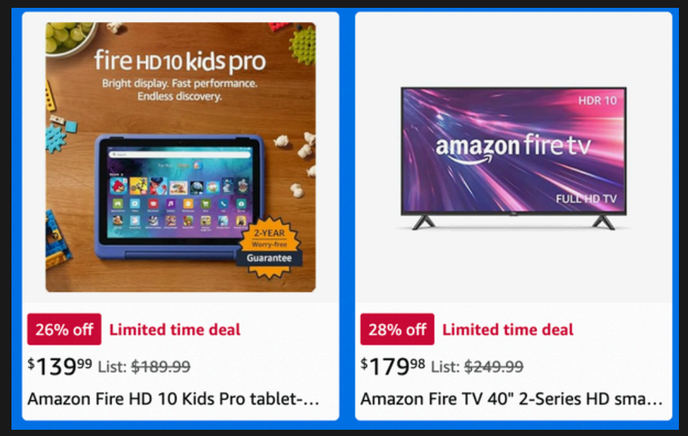
A webpage that has the information you want. The website consists of HTML code, containing the structure and data that will be scraped.



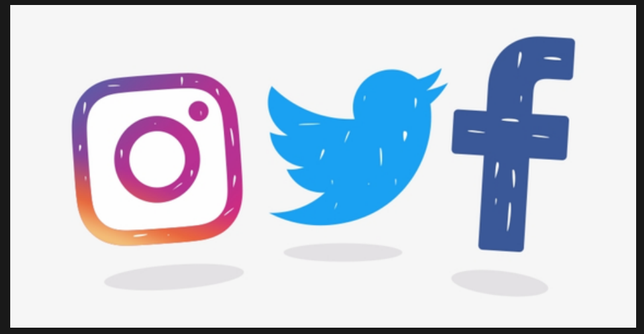
- The scraper sends an HTTP GET request to the website's server to retrieve the HTML file (webpage content).
- The website responds by sending back the entire webpage content, which contains all the page elements, like text, images, and links.
- The HTML content is parsed using libraries to navigate through the elements and tags, identifying the data to extract.
- The scraper identifies the specific HTML tags (e.g., , , <p>) and attributes (e.g., class, id) to extract the desired data.

The scraped data is stored in a structured format such as CSV, JSON, or a database for further analysis or use.

What can you scrape?



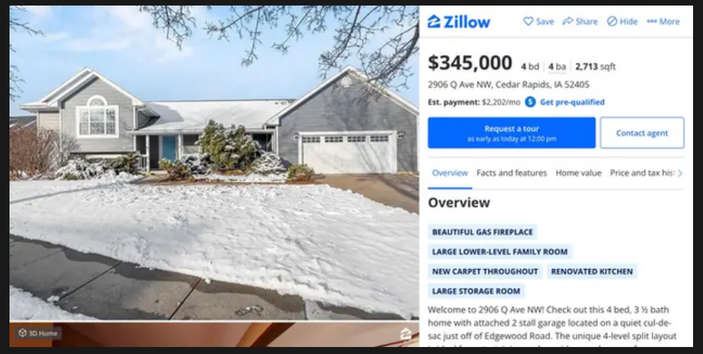
Collect prices from online stores to compare and track changes over time.



Scrape Twitter or Instagram posts to analyze trends or public opinion on a topic.

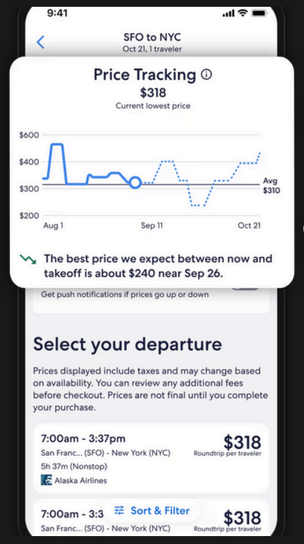
Gather job postings from multiple sites to build a job search tool or analyze hiring trends in a specific industry.

Pull headlines, summaries, or full articles from news websites to track breaking news or run sentiment analysis.



Collect property prices and descriptions from real estate websites to analyze the housing market.

Scrape weather forecasts from sites to build custom weather apps or analyze trends.



Scrape flight prices and schedules to compare airlines or track flight delays and cancellations.

and much more!

Legal Considerations & Terms of Service

- ***Legal Issues:***
  - Web scraping sits in a legal gray area. While scraping publicly available data is generally allowed, violating a website's terms of service (ToS) can lead to legal consequences.
  - Some websites explicitly prohibit scraping, and ignoring these rules may result in a lawsuit or being blocked.
- ***Cases of Legal Scraping:***
  - **HiQ Labs vs. LinkedIn:** HiQ Labs scraped public LinkedIn profiles for data analysis. LinkedIn filed a lawsuit, but the courts ruled in favor of HiQ, citing that publicly accessible data is not restricted by anti-scraping laws.
  - **Craigslist vs. 3Taps:** Craigslist successfully sued 3Taps for scraping and republishing its data. The court ruled that scraping content in violation of the site's ToS was illegal.
- ***Terms of Service (ToS):***
  - Always check a website’s ToS before scraping. Websites often include clauses about data usage and scraping restrictions. Violating these terms can result in legal action.

Ethical Scraping Guidelines

- **Respect the ToS:** Always follow a website's terms of service.
- **Check robots.txt:** Respect the directives in the robots.txt file.
- **Use proper intervals:** Don’t overwhelm a website with too many requests at once.
- **Cite your sources:** If you’re using data from a scraped website, give proper credit.
- **Use API when Available**: Prefer using official APIs if available, as they are designed for data access.
- **Data Accuracy**: Maintain the integrity and accuracy of the data being scraped.
- **Avoid Personal Data**: Steer clear of scraping personal or sensitive information without explicit permission.

*Non-compliance with web scraping laws can lead to serious consequences. **Legal action**, such as lawsuits or cease-and-desist orders, may be taken against violators, potentially resulting in hefty **financial penalties** or compensation to affected parties. Beyond legal repercussions, non-compliance can severely **damage a company’s reputation**, eroding customer trust and business relationships. **Operational disruptions**, including business interruptions from legal proceedings, are common, and **key partnerships or contracts may be terminated.** Offending companies may also face **access denial**, where websites block their IP addresses, cutting off future data access. Additionally, **regulatory scrutiny** may increase, subjecting the company to more oversight.*

Robots.txt

What is robots.txt?

robots.txt is a file that websites use to communicate with web crawlers and other automated tools. It defines which sections of the website can or cannot be accessed by crawlers.

Ethical Implications

While robots.txt is not legally binding, respecting its rules is considered best practice for ethical web scraping. Websites use this file to protect sensitive areas or prevent overwhelming their servers with traffic from scrapers.

Robots.txt File Example

Visit any website and type /robots.txt at the end of the domain name to view its file.
Example:
https://www.example.com/robots.txt.

Any path listed after the Disallow: directive indicates sections of the website that are off-limits
The Allow: directive specifies the areas that the bot is allowed to scrape

Ethical Web Scraping - Best Practices and Tools

- Avoid scraping personal or sensitive data without permission.
- Use caching to avoid repetitive requests to the same pages. Caching means saving the results of your previous scraping so that you don’t have to request the same page again if the data hasn’t changed. This reduces the load on the website and speeds up your scraper since it doesn’t need to keep downloading the same content. For example, if you scraped product prices yesterday and they haven’t changed today, you can reuse the saved data instead of scraping the page again.Set your scraper to send appropriate headers and user-agent strings to mimic normal browser activity.
- Rotate IP addresses to avoid being blocked for excessive requests. Websites track the number of requests coming from a single IP address (your computer's unique online identifier). If too many requests come from one IP in a short time, the website may block you. By rotating IP addresses, you spread your requests across different addresses, making it look like the traffic is coming from multiple users and reducing the risk of being blocked. This is like taking turns with different devices to avoid overwhelming a system with too many requests from one source.
- By adding small pauses (delays) between your requests, you prevent overwhelming the server with too much traffic at once. This helps you avoid being detected and blocked, and it ensures the website can function properly for other users.

TOOLS: Selenium, BeautifulSoup, Scrapy (A web scraping framework that is ideal for scalable and responsible scraping. It provides built-in features for handling requests, following links, and managing data pipelines in an efficient and ethical manner), Rotating Proxies (Services like ScraperAPI provide rotating proxies to avoid IP bans. These proxies automatically rotate your IP address for each request, helping you scrape large volumes of data without being blocked by the server).

Web Scraping Best Practices

- **Use Headers and User Agents:**
  - Set your scraper to send appropriate headers and user-agent strings to mimic normal browser activity.
- **Monitor Rate Limits:**
  - Be mindful of rate limits set by the server to avoid being banned.
- **Check for API Availability:**
  - Use APIs whenever available. They provide structured access to data and typically have clear guidelines on usage.
- **Error Handling:**
  - Implement error handling in your scraper to manage cases where a website is down or unresponsive.
- **Log Your Activity:**
  - Keep logs of your scraping activities to track issues and respect rate limits.