Python Project for Data Science

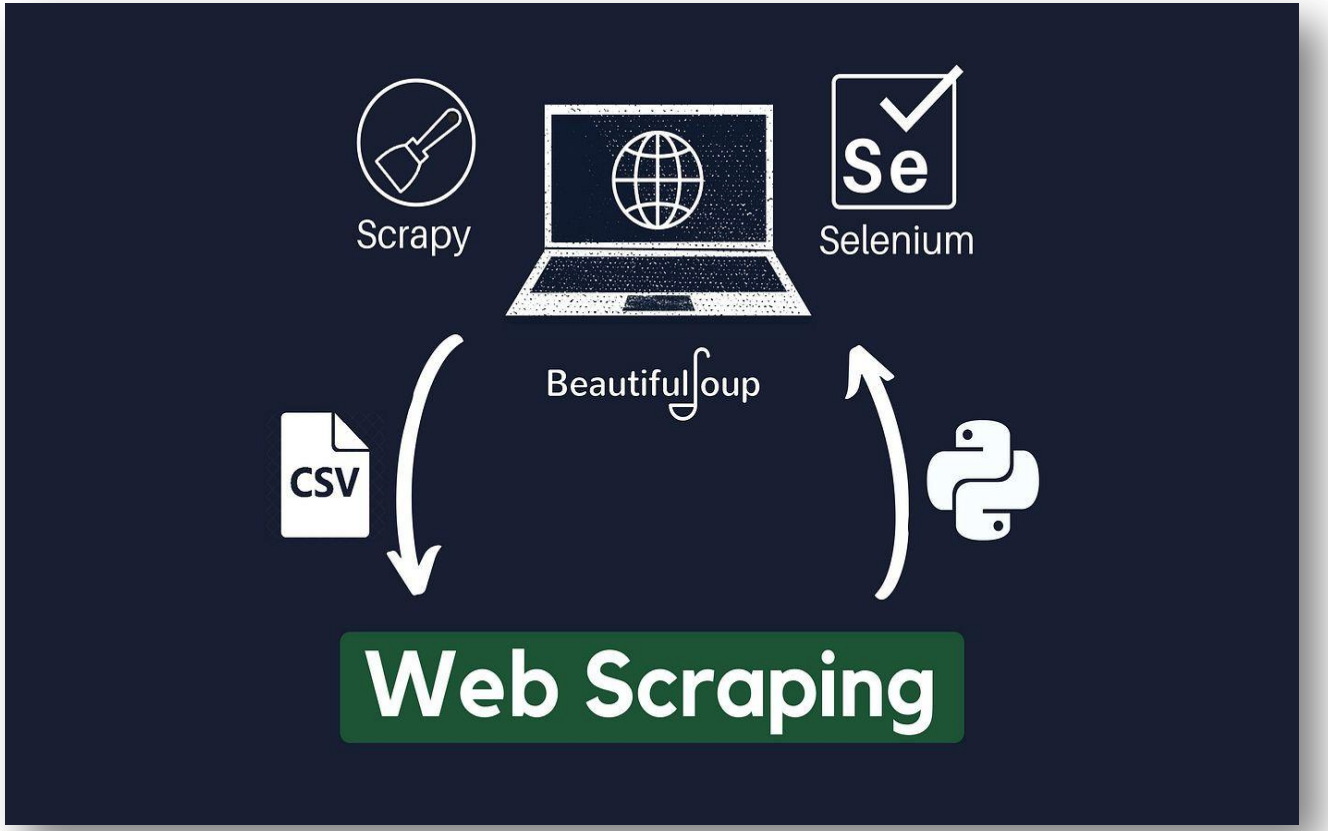
Web Scraping

Course Overview

You will perform specific data science and data analytics tasks such as extracting data, web scraping, visualizing data and creating a dashboard. This project will showcase your proficiency with Python and using libraries such as Pandas and Beautiful Soup within a Jupyter Notebook. Upon completion you will have an impressive project to add to your job portfolio.

Session Content

- What is Web Scraping?
- How do we do?
- Tools to use
- Ethics for scraping
- Demo

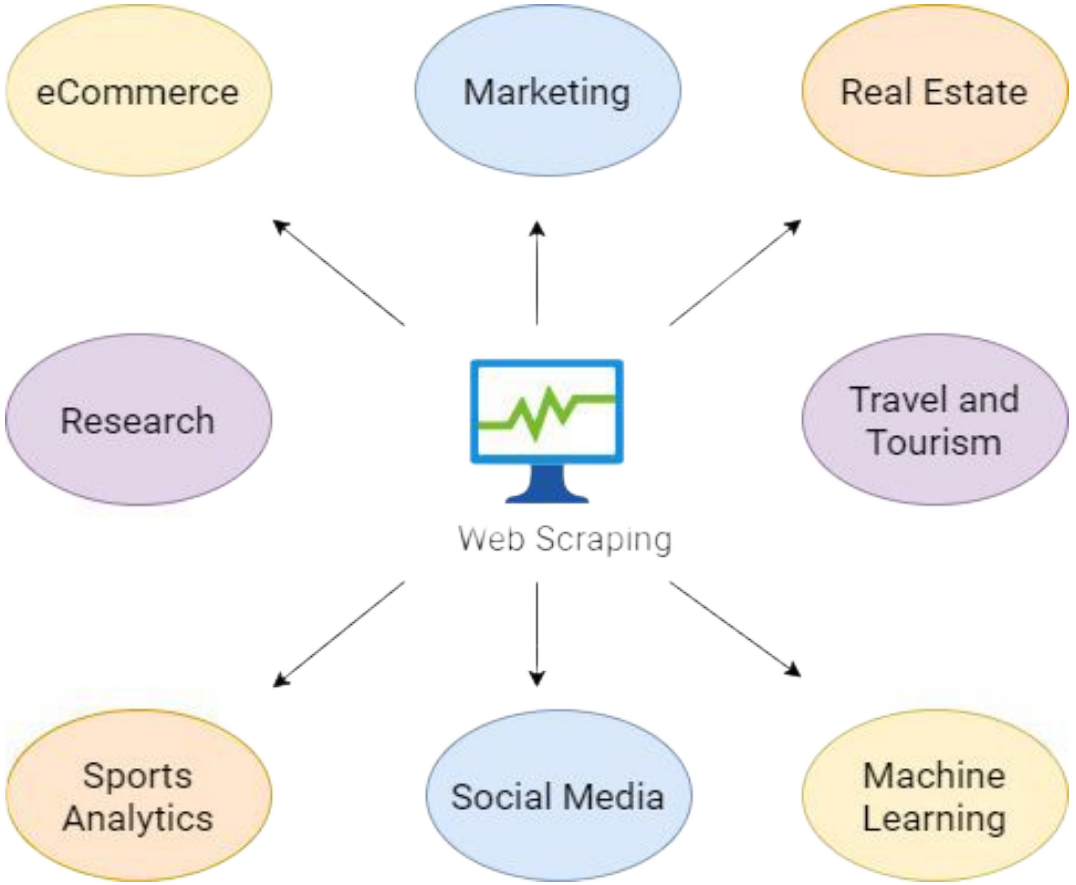


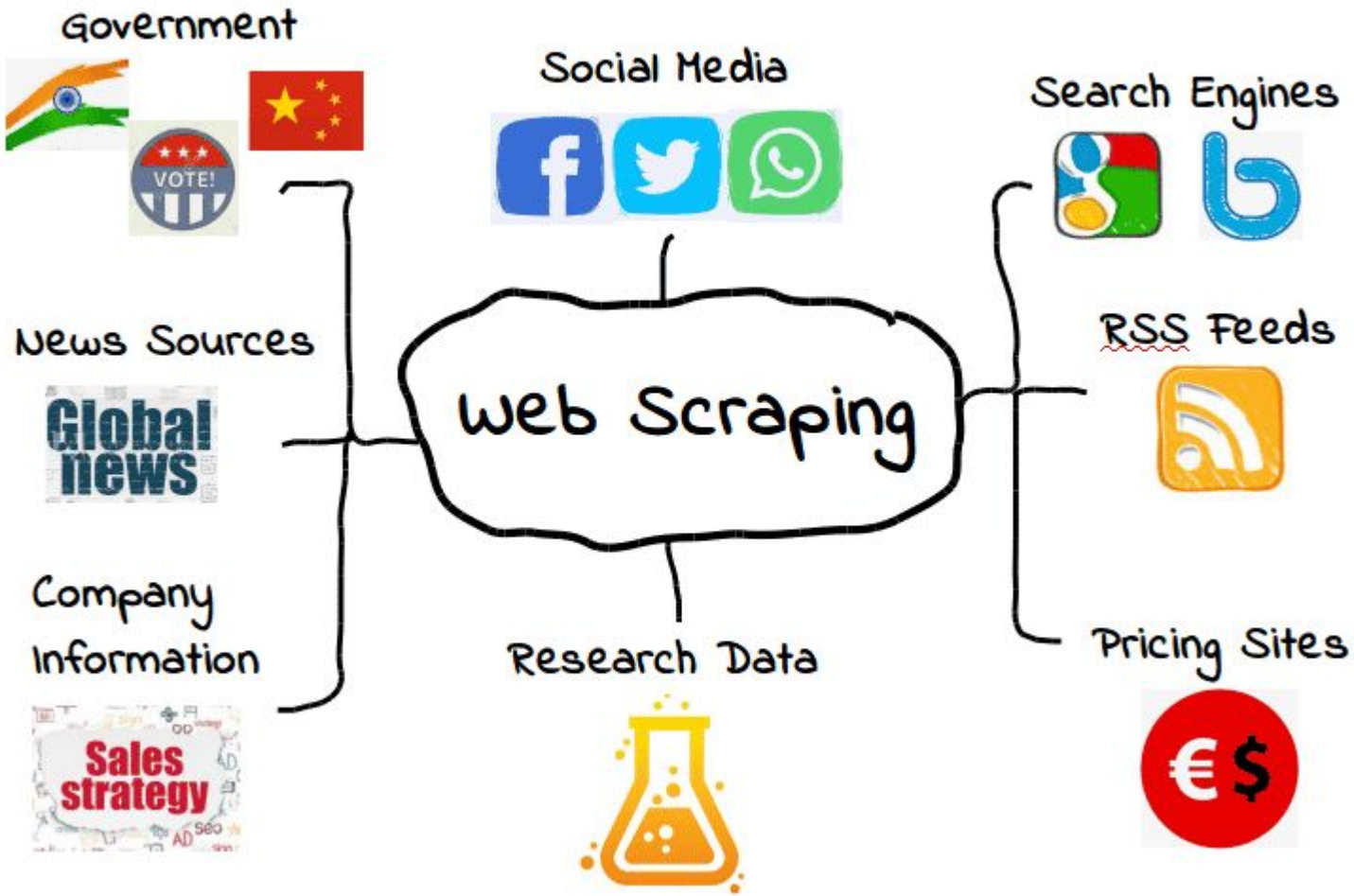
What is Web Scraping?

- **Web scraping:** is technique for gathering data or information on web pages.
- **Web scraping:** is method to extract data from a website that does not have an API, or we want to extract LOT of data which we can not do through an API due to rate limiting.
- Through web scraping we can extract any data which we can see while browsing the web.
- You could revisit your favorite website every time it updates for new information, Or you could write a web scraper to have it

Web Scraping in Real Life

- Extract products information
- Extract job posting and internships
- Extract offers and discount from deal of the day website
- Extract date to make search engine
- Gathering weather data
- Etc.




Government
VOTE!
Social Media
Search Engines
News Sources
Global news
Web Scraping
RSS Feeds
Company Information
Sales strategy
Research Data
Pricing Sites
€ $

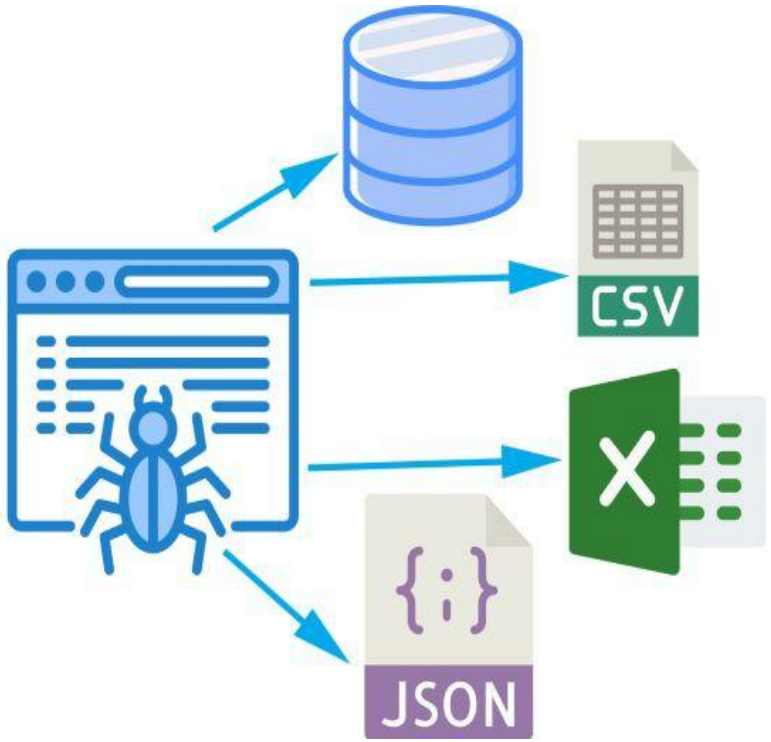
Advanced Web Scraping Vs. API

- Web scraping is not rate limited
- Anonymously access the website and gather data
- Some website don’t have API
- Some data is not accessible through an API

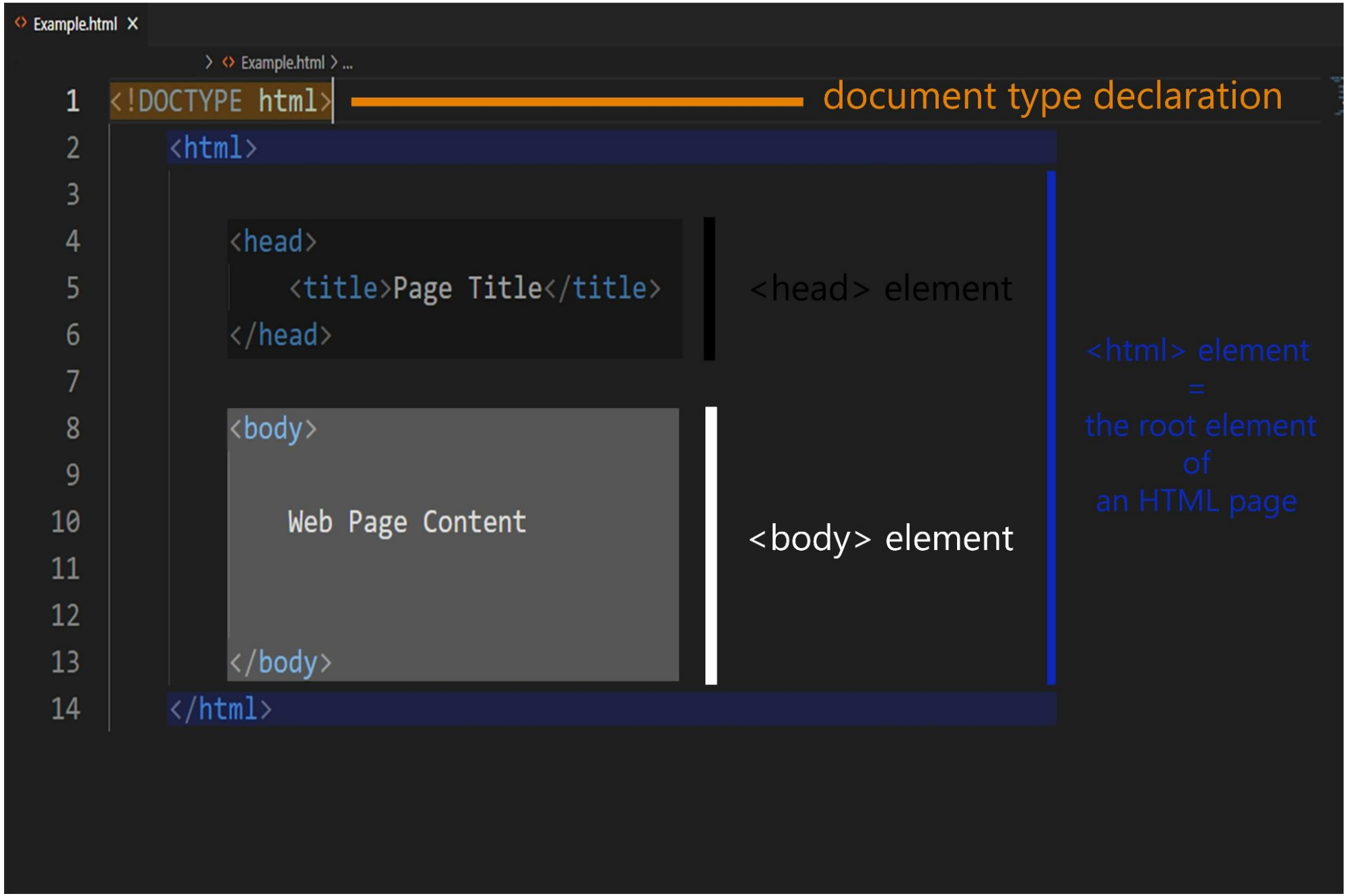
Workflow

Web scraping follows this workflow:

1. Get the website – using HTTP library
2. Parse the html document – using any parsing library
3. Store the results – either a db , csv, txt file etc.



Term	Definition (Simple Explanation)
Web Scraping	The process of automatically extracting data from websites.
HTML	The structure/language websites are built with – like the skeleton of a webpage.
Tag	An HTML element like `<h1>`, `<p>`, or `<div>` used to define different parts of a webpage.
Element	A single component of an HTML page (like a quote, image, or button).
Attribute	Extra information inside an HTML tag (like `class`, `id`, `href`).
Class	A type of attribute used to group HTML elements with similar styles or purposes.
ID	A unique identifier for one specific HTML element.
Parser	A tool that reads and analyzes HTML content (e.g., `html.parser` in BeautifulSoup).
BeautifulSoup	A Python library used to parse HTML and extract specific data.
requests	A Python library used to send HTTP requests and receive website content.
HTTP Request	A message sent to a web server asking for a specific webpage or data.
Response	The data or webpage the server sends back after an HTTP request.
GET Request	A type of HTTP request used to ask for data from a server (usually a webpage).


Example.html
Example.html > ...
1 <!DOCTYPE html>
document type declaration
2 <html>
3
4 <head>
5 <title>Page Title</title>
6 </head>
<head> element
7
8 <body>
9
10 Web Page Content
11
12
13 </body>
<body> element
14 </html>
<html> element
=
the root element
of
an HTML page

Html Tags

Tag	Description
<html> ... </html>	Declares the Web page to be written in HTML
<head> ... </head>	Delimits the page's head
<title> ... </title>	Defines the title (not displayed on the page)
<body> ... </body>	Delimits the page's body
<h *n*> ... </h*n*>	Delimits a level *n* heading
...	Set ... in boldface
...	Set ... in italics
...	Center ... on the page horizontally
...	Brackets an unordered (bulleted) list
...	Brackets a numbered list
...	Brackets an item in an ordered or numbered list
 	Forces a line break here
<p>	Starts a paragraph
<hr>	Inserts a horizontal rule
<img src="...">	Displays an image here
 ... 	Defines a hyperlink

⚠️ Mandatory ⚠️

📚 You *must* watch and study the following resources if you don't know Html before.

1. **Learn HTML In One Video**

   (Complete introduction to HTML in a simple and practical way)

2. **HTML Tags Reference**

   (Understand the most important HTML tags and their usage)

HTML Page Structure

```
<!DOCTYPE html>
<html>

<head>
  <title>Page Title</title>
</head>

<body>
  <h2>Heading Content</h2>
  <p>Paragraph Content</p>
</body>

</html>
```

Tells version of HTML

HTML Root Element

Used to contain page HTML metadata

Title of HTML page

Hold content of HTML

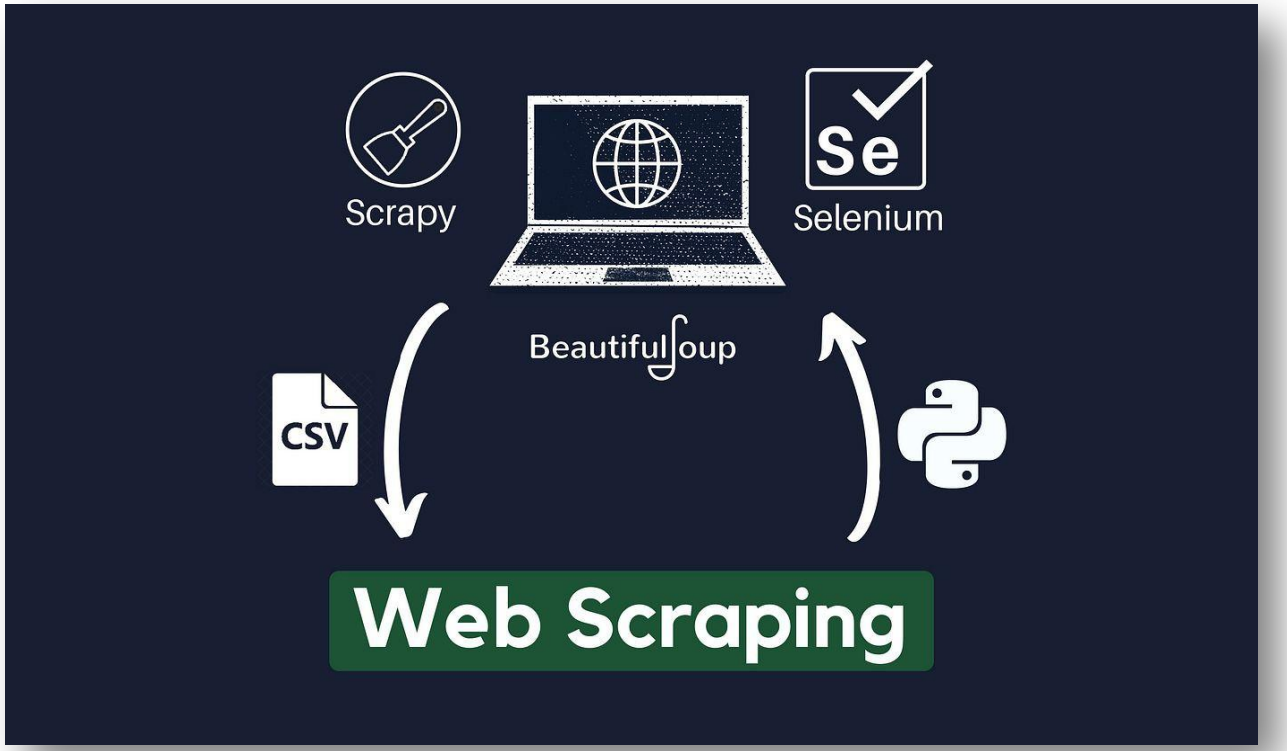
HTML headling tag

HTML paragraph tag

https://jekso.github.io/scrapping-example/index.html

Libraries

- BeautifulSoup (bs4)
- Selenium
- scrapy **Framework**
- Pandas and requests



Feature	BeautifulSoup	Selenium	Scrapy
Type	HTML parser library	Browser automation tool	Full-featured web scraping framework
Best For	Simple static websites	Interactive or JavaScript-heavy websites	Large-scale scraping projects and crawlers
JavaScript Support	No	Yes	No (requires extra tools like Splash)
Speed	Fast	Slower (due to real browser rendering)	Very fast and optimized for crawling
Ease of Use	Easy and beginner-friendly	Relatively easy, but setup required	Steeper learning curve
Installation	Lightweight (only needs BeautifulSoup + Requests)	Requires browser drivers (e.g., ChromeDriver)	Requires full Scrapy setup
DOM Interaction (Click, Scroll)	Not supported	Fully supported	Not supported
Built-in Crawling Features	No (manual handling required)	No	Yes (spiders, pagination, URL rules)
Data Export Options	Manual (e.g., CSV, JSON using pandas)	Manual	Built-in support for JSON, CSV, XML
Suitable For	Small tasks, quick scripts	Medium-scale, interactive scraping	Enterprise-level scraping, large-scale crawling

Is Web Scraping Legal?

In short, the action of web scraping is not illegal. However, some rules need to be followed. Web scraping is illegal when non-publicly available data is extracted.

https://github.com/KOrfanakis/Web_Scraping_With_Python

Questions & Answers