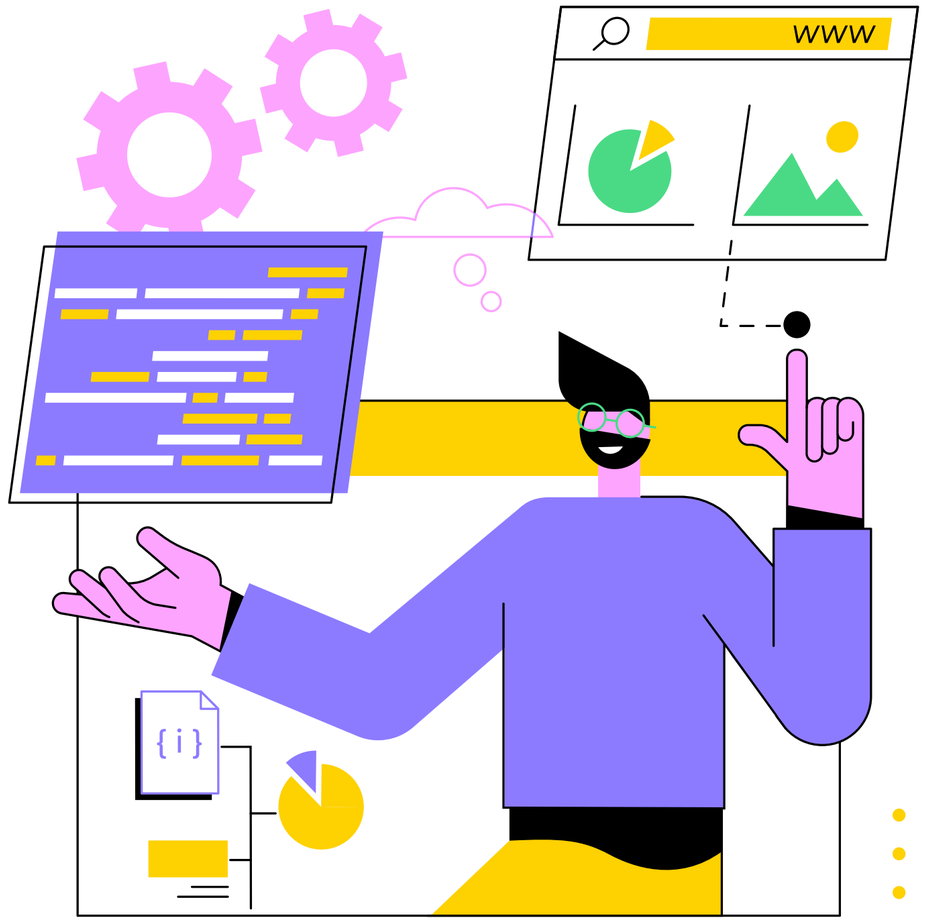
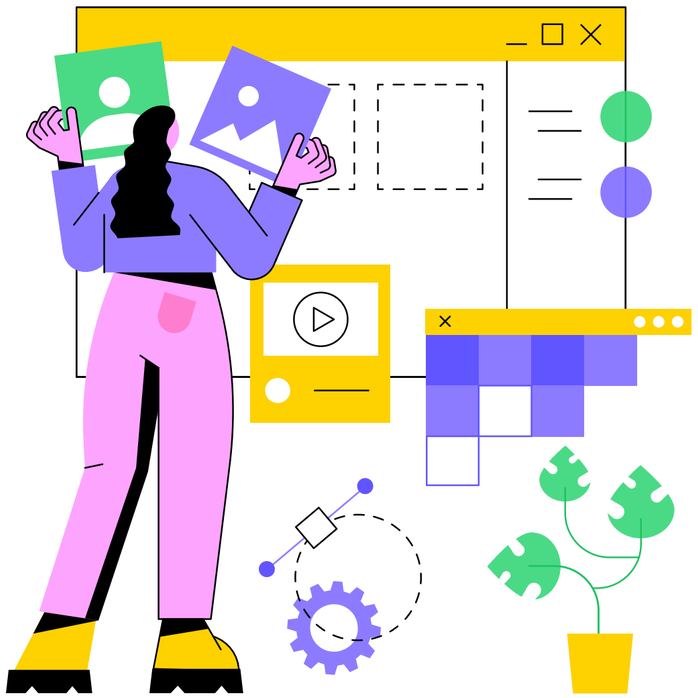
WEB SCRAPING USING PYTHON

By- Chirag Rajput (211IT018)



WHAT IS WEB SCRAPPING?

- Web scraping refers to the automated process of extracting data from websites.
- It involves retrieving, parsing, and extracting relevant information from HTML or XML web pages from the Browser at user-end.
- Web scraping enables us to collect valuable data from various online sources efficiently and at scale.



WHAT IS WEB SCRAPPING?

- Data collected through web scraping can be used for market research, competitive analysis, sentiment analysis, price monitoring, and more.
- It provides insights and information that can drive decision-making, enhance business strategies, and fuel research projects.

HOW DOES WEB SCRAPPING WORKS?

- Web scraping involves fetching web pages, extracting the relevant data, and storing it for further analysis.
- The process typically includes sending HTTP requests to the target website, receiving HTML or XML responses, and parsing the data to extract the desired information.
- We use APIs Or python Libraries to do so.

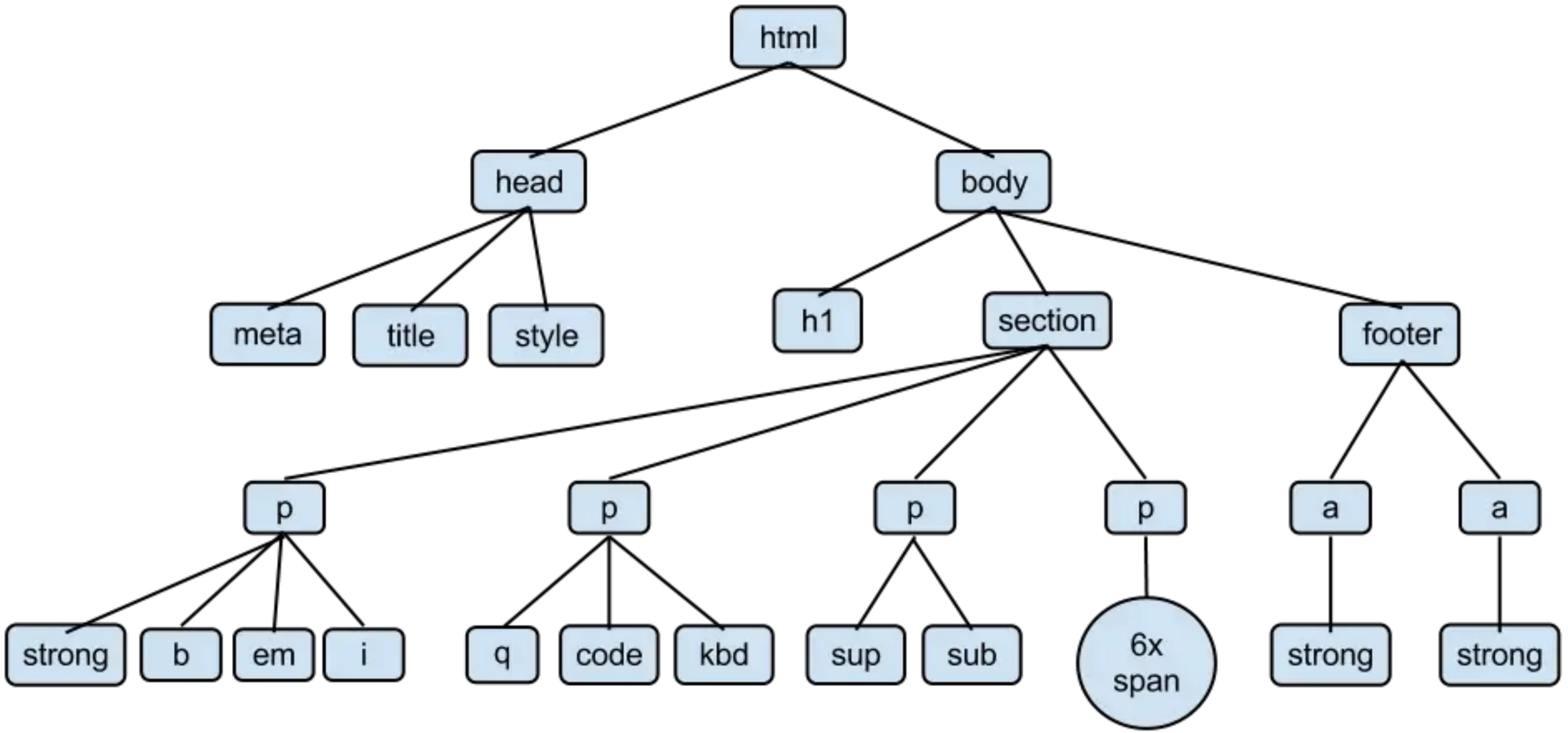
TOOLS USED

- Python is the most commonly used tool for web scrapping.
- other web scraping tools include BeautifulSoup, Scrapy, Selenium, and Requests.
- These tools provide functionalities to parse HTML/XML, navigate web pages, handle dynamic content, and extract data efficiently.

BEAUTIFULSOUP:

- A popular library for parsing HTML and XML documents, providing a simple and intuitive interface for data extraction.
- Beautiful Soup takes the HTML or XML content as input and uses an underlying parser library (such as html.parser, lxml, or html5lib) to convert the content into a parse tree.
- The parser library handles the low-level parsing of the markup language and builds a tree-like structure that represents the document's structure.

PARSE TREE EXAMPLE:



CODE:

```
<html>
    <head>
        <meta>
        <title>Tree</title>
        <style>

        </style>
    </head>
    <body>
        <h1>Heading</h1>
        <section>
            <p>
                <i></i>
                <em></em>
                <b></b>
                <strong></strong>
            </p>
            <p>
                <q></q>
                <code></code>
                <kbd></kbd>
            </p>
            <p>
                <sup></sup>
                <sub></sub>
            </p>
            <p>
                <span></span>
                <span></span>
                <span></span>
                <span></span>
                <span></span>
                <span></span>
            </p>
        </section>
        <footer>
            <a href=""><strong></strong></a>
            <a href=""><strong></strong></a>
        </footer>
    </body>
</html>
```

BEAUTIFULSOUP FUNCTIONS

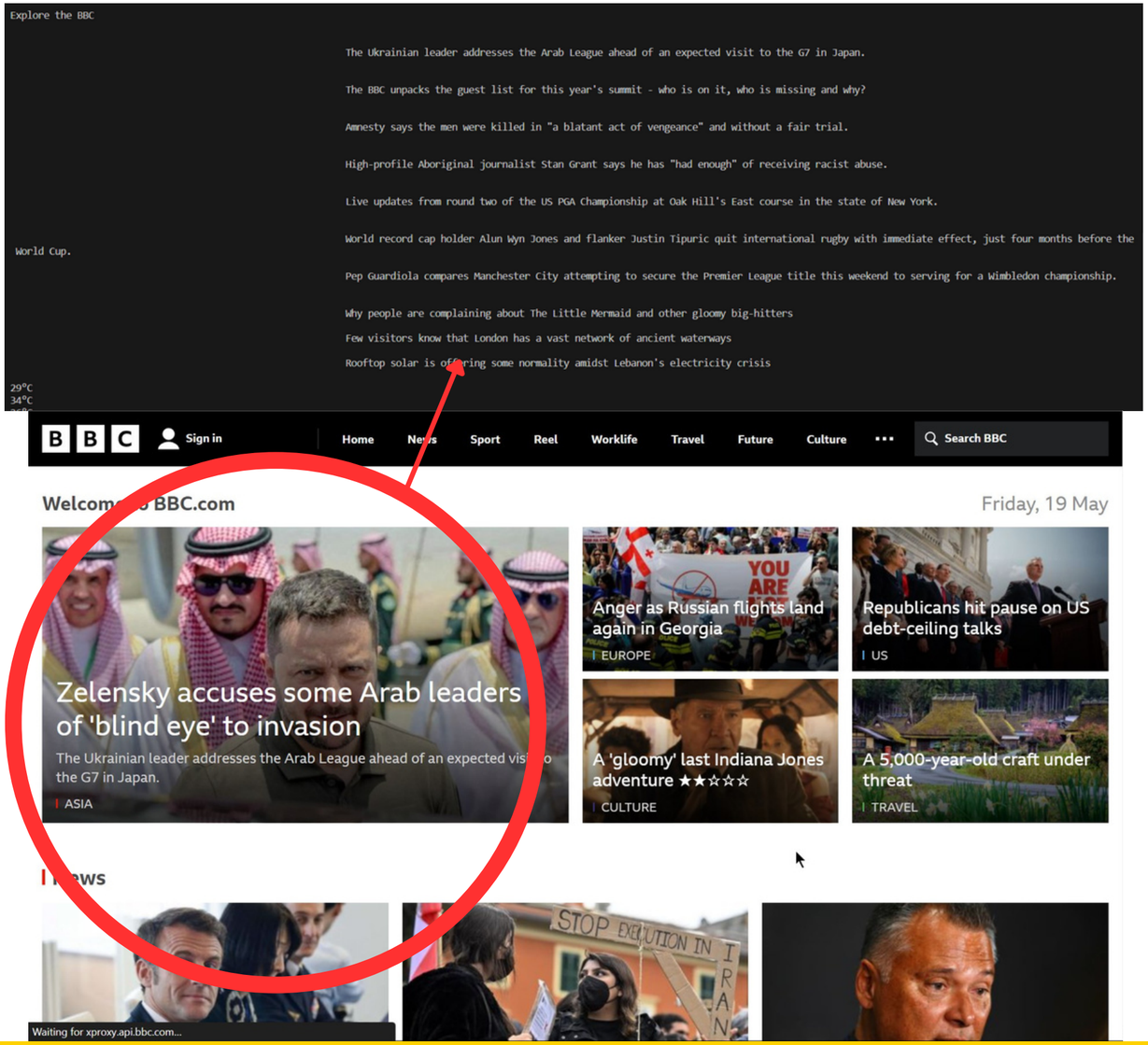
- **Searching by Tag:** find() and find_all()
- **Searching by CSS Selectors:** select() ex-soup.select('.class-name')
- **Accessing Element Attributes:** get() ex-tag.get('href')
- **Navigating the Parse Tree:** parent, contents, next_sibling, and previous_sibling
- **Modifying the Parse Tree:** new_tag(), append(), insert(), replace_with(), and extract()
- **Pretty Printing:** prettify()

REQUESTS:

A library for sending HTTP requests, commonly used for fetching web pages and handling authentication.

Web1.py > ...

```python
import requests
from bs4 import BeautifulSoup

url = "https://bbc.com"  # Website We are scrapping
response = requests.get(url)

# Creating a BeautifulSoup object to parse the HTML content
soup = BeautifulSoup(response.text, "html.parser")

# Finding and extracting specific data from the HTML
h1_tags = soup.find_all("h1")
h2_tags = soup.find_all("h2")
p_tags = soup.find_all("p")

#printing the data
for tag in h1_tags:
    print(tag.text)

for tag in h2_tags:
    print(tag.text)

for tag in p_tags:
    print(tag.text)

```



```python
import requests
from bs4 import BeautifulSoup

# Send a GET request to the website
url = "https://www.example.com/table.html"  #URL of the page with the table
response = requests.get(url)

# BeautifulSoup object to parse the HTML content
soup = BeautifulSoup(response.text, "html.parser")

# Finding the table element
table = soup.find("table")

# Extracting data from the table
data = []
for row in table.find_all("tr"):
    row_data = []
    for cell in row.find_all("td"):
        row_data.append(cell.text.strip())
    if row_data:
        data.append(row_data)

# Printing the extracted data
for row in data:
    print(row)
```

To get the data of a web page having a table we can proceed like this.

To check the name we can inspect and find the name of the current page

```python
import requests
from bs4 import BeautifulSoup
import os

# Function to download an image from a URL
def download_image(url, directory):
    response = requests.get(url)
    if response.status_code == 200:
        file_name = url.split("/")[-1]
        file_path = os.path.join(directory, file_name)
        with open(file_path, 'wb') as file:
            file.write(response.content)
        print(f"Image downloaded: {file_name}")
    else:
        print(f"Failed to download image from: {url}")

# Function to scrape images from a website
def scrape_images(url, directory):
    response = requests.get(url)
    if response.status_code == 200:
        soup = BeautifulSoup(response.content, 'html.parser')
        img_tags = soup.find_all('img')

        if not os.path.exists(directory):
            os.makedirs(directory)

        for img in img_tags:
            if 'src' in img.attrs:
                img_url = img['src']
                if img_url.startswith('http'):
                    download_image(img_url, directory)
    else:
        print(f"Failed to scrape images from: {url}")


# Example usage
website_url = "https://thepetwala.com"  #Website to download images from.
download_directory = "E:\Main\College\IT290 Seminar"  # Directory to save the images to.

scrape_images(website_url, download_directory)
```

Scraping and downloading images--

The **scrape_images()** function takes a URL of a website and a directory path, send a GET request to the website, parses the HTML content using BeautifulSoup, finds all **<img>** tags, and downloads the images found in those tags using the **download_image()** function which takes a URL of an image and a directory path, downloads the image from the URL, and saves it in the specified directory.

```
PS E:\Main\Projects\WebScraping_Seminar> python -u "e:\Main\Projects\WebScraping_Semina
Image downloaded: Logo.png
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 84770f_93f926c13b8a48b5a459f55ec2236efc~mv2.jpg
Image downloaded: 82fcd3_8d5625845a1d4676b50d97368cf8050e~mv2_d_1735_2000_s_2.jpg
Image downloaded: 82fcd3_8e5daf6e362d432f9f34ff70780c6752~mv2_d_5509_4000_s_4_2.jpg
Image downloaded: 82fcd3_97259c1aab9e4a9d9d9a6d33896ba1ec~mv2_d_3745_2492_s_4_2.jpg
Image downloaded: 11062b_bdb85cc11bbd46c1a25deabb11b96b71~mv2.jpg
Image downloaded: 2d9d0e_655c3e7bd2b44589b09a40a89d0314af~mv2.png
Image downloaded: 2d9d0e_32214c90c5ab4ba7bbba1fb67d72c421~mv2.png
Image downloaded: 2d9d0e_fd7aa412e2a54bb88e97fbe24e0838e3~mv2.png
Image downloaded: 2d9d0e_dfc293082fa14d31a1199508b20dfe00~mv2.png
Image downloaded: 2d9d0e_78002772e0364d538e1a2c3dace0d3e4~mv2.png
Image downloaded: 2d9d0e_f01e557e4e0a4de99f06fa56fa6da594~mv2.png
Image downloaded: 2d9d0e_46123cce44714aceaaeacded2ef5e88d~mv2.png
Image downloaded: 2d9d0e_655c3e7bd2b44589b09a40a89d0314af~mv2.png
Image downloaded: 2d9d0e_32214c90c5ab4ba7bbba1fb67d72c421~mv2.png
Image downloaded: 2d9d0e_fd7aa412e2a54bb88e97fbe24e0838e3~mv2.png
Image downloaded: 2d9d0e_dfc293082fa14d31a1199508b20dfe00~mv2.png
Image downloaded: 2d9d0e_78002772e0364d538e1a2c3dace0d3e4~mv2.png
Image downloaded: 2d9d0e_f01e557e4e0a4de99f06fa56fa6da594~mv2.png
```

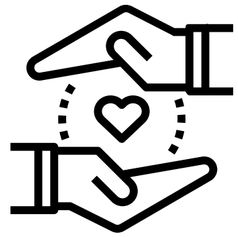
Images getting downloaded in the specified directory

KNOW ABOUT IT

ETHICS

Respecting website Resources

Privacy



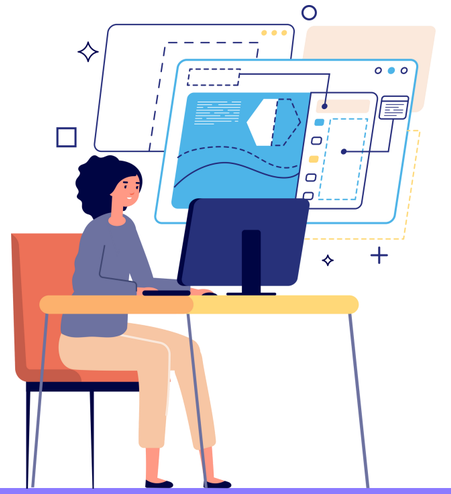
Data Attribution and Copyrights

Ethics

Ethical web scraping involves obtaining data with the owner's permission and ensuring minimal impact on the website's performance.
Respect the website's terms of service and adhere to legal requirements.

Respecting website Resources

When scraping websites, we should consider the resources consumed (bandwidth, server load, etc.). as it can impact the website and may inturn be a loss for the owner.

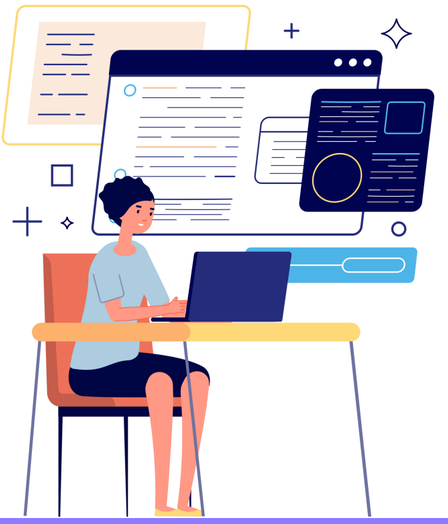


Privacy

Ensure that any scraped data is used responsibly and securely, following applicable privacy guidelines.
It can be Illegal to scrap personal user data from websites

Data Attribution and Copyrights

Proper attribution should be provided when using published data scrapped from website or using copyrighted data.



```python
import praw
reddit = praw.Reddit(
    client_id="5wF9x_\           _7Cw",
    client_secret="p9\           Hvkw2N3WxKG_VgsA",
    user_agent="seminar",
    username="us     322",
    password="ch     ajput"
)

subreddit = reddit.subreddit("Python")

for post in subreddit.hot(limit=5):
    print(post.title)
    print()
print("The top 10 hottest posts in machine learning subreddit are->")
hot_posts = reddit.subreddit("MachineLearning").hot(limit=10)
for posts in hot_posts:
    print(posts.title)
print()
print("The top 10 hottest posts in climate change subreddit are->")
climate = reddit.subreddit("climatechange").top(limit=10)
for posts in climate:
    print(posts.title)
print()
print("The top 10 recent commentors in the subreddit damnthatsinteresting are->")
for comment in reddit.subreddit("Damnthatsinteresting").comments(limit=10):
    print(comment.author)
```

Example of Scrapping Reddit using Reddit APP API, PRAW (Python Reddit API Wrapper) is a Python library that provides a way to interact with the Reddit API.

Data that was fed into making chatgpt was also partially taken from Reddit using web scraping.

Output

```
PS E:\Main\Projects\WebScraping_Seminar> python -u "e:\Main\Projects\WebScraping_Seminar\Reddit.py"
Sunday Daily Thread: What's everyone working on this week?

Thursday Daily Thread: Python Careers, Courses, and Furthering Education!

I'm open-sourcing the Python interface for JOY OF PROGRAMMING - a programming game where you automate all kinds of machines, robots, drones and more.

Propan - is a new python framework for building messaging services

NoDoze - smart sleep for Linux computers

The top 10 hottest posts in machine learning subreddit are->
[D] Simple Questions Thread
Reminder: Use the report button and read the rules!
[R] Tree of Thoughts paper
[D] Is there a theory of Deep Learning?
[P] Best image classifier architecture right now
[P] Testing different popular GPT tokenizers
[D] Over Hyped capabilities of LLMs
[D] Generative vs embedding models
[N] Daily Papers by Hugging Face
[D] Conflicting gradients in multiple heads

The top 10 hottest posts in climate change subreddit are->
I'm afraid climate change is going to kill me! Help!
This news article from 108 years ago
Equip me in the debate on global warming
They knew. They knew a hundred years ago.
Surface area of solar panels required to power entire U.S.
Calvin & Hobbes captured the generational divide over Climate Change... in 1987.
Not much to talk about I guess
14,000 scientists warn of "untold suffering" if we fail to act on climate change
Costa Rica will run on more than 98% renewable energy for fifth consecutive year
Countries with the Largest CO2 Pollution Over Time

The top 10 recent commentors in the subreddit damnthatsinteresting are->
Bigtex6786
RAMENBELLY
darknessinthere
Hodaka
Brodman_area11
silentbassline
TheIndomitableMass
AldousCuckskey
TomiShinoda
Blackletterdragon
PS E:\Main\Projects\WebScraping_Seminar>
```

- Once we have scraped the data it is stored.
- This stored data can be then used with Python libraries like Matplotlib, seaborn, and Plotly to create visuals.
- This can serve as the dataset for ML algorithms and can be used to build AI bots.
- Data scraping is a powerful tool and should be used ethically.

Thank you