Web Scraping Mini Project

Tutorials Freak Scraper

August 17, 2025

Description

This Python script demonstrates a beginner-friendly web scraping project using the `requests` and `BeautifulSoup` libraries. It fetches the homepage of `tutorialsfreak.com`, parses its HTML content, and explores various BeautifulSoup methods to access tags, extract navigable strings, find elements by class, and retrieve image sources. The code is organized with comments to reflect the original notebook's structure, making it easy to understand and execute. To run, install dependencies with `pip install requests beautifulsoup4` and execute `python web_scraping_mini_project.py`. Note: Web scraping should respect the website's `robots.txt` and terms of service.

Consolidated Code

```
# Web Scraping Mini Project: Scraping tutorialsfreak.com
# This script demonstrates basic web scraping using requests and BeautifulSoup.
# It fetches the page, parses HTML, accesses tags/strings, and finds elements.

import requests
from bs4 import BeautifulSoup

# Fetch the webpage
web = requests.get("https://www.tutorialsfreak.com/")
print(web)  # Expected: <Response [200]>

# Parse the content with BeautifulSoup
soup = BeautifulSoup(web.content, "html.parser")

# Section: Accessing Tags
# Demonstrates how to access specific HTML tags.
tag = soup.html
print(type(tag))  # Expected: <class 'bs4.element.Tag'>

tag = soup.p
print(tag)  # First <p> tag

tag = soup.a
```

```
print(tag)  # First <a> tag

tag = soup.h1
print(tag)  # First <h1> tag

tag = soup.h2
print(tag)  # First <h2> tag

# Section: Navigable Strings
# Extracts text content from tags.
tag = soup.p.string
print(tag)  # Text from first <p>

tag = soup.h1.string
print(tag)  # Text from first <h1>

tag = soup.h2.string
print(tag)  # Text from first <h2>

# Section: BeautifulSoup Methods
# Explores soup properties and search methods.
print(soup.name)  # Expected: '[document]'

print(soup.title)  # <title> tag

print(soup.find('h1'))  # First <h1>

print(soup.find_all('h1'))  # All <h1> tags

print(soup.find('p'))  # First <p>

print(soup.find_all('p'))  # All <p> tags

# Section: Comments and Prettifying
# Handles strings and formatted HTML output.
com = soup.p.string
print(com)  # Same as navigable string

print(soup.p.prettify())  # Prettified <p> tag

# Section: Finding Elements by Class
# Finds a specific div by class (fixed typo in class name).
class_data = soup.find("div", class_="bg-white border-0 p-3 card-
    footer")
print(class_data)  # The matching div element

# Extract all image sources
img = soup.find_all('img')
for i in img:
    print(i.get("src"))  # Prints all image src attributes
```

Listing 1: web_scraping_mini_project.py