Documentation:

HTML Web Scraping Project on BookstoScrape Website

Objective of This Project

The purpose of this project is to scrape book titles and their corresponding prices from the website Books to Scrape using Python. The scraped data is then stored in a structured format (a pandas Data Frame) for further analysis or use. This project demonstrates how to gather and process data from a web page programmatically.

Python and Its Libraries

This project utilizes Python and the following libraries and they are:

1. **requests:**
   - A library to send HTTP requests and retrieve the HTML content of web pages.
2. **beautifulsoup4:**
   - This is a parsing library that is used to navigate, search, and modify the HTML information in the web pages.
   - It provides tools to extract elements from the HTML using CSS selectors or tags.
3. **pandas:**
   - This library is used for data manipulation and analysis.
   - The extracted data from website is organized into a DataFrames for handling and storing the data easily.

How Does It Work Internally?

1. **Step 1: Fetching the Web Page**
   - The requests library sends an HTTP GET request to the website URL.
   - The HTML content of the web page is downloaded and stored in the response object.
2. **Step 2: Parsing the HTML Content**
   - The HTML content from the response object is passed to BeautifulSoup for parsing.
   - BeautifulSoup creates a tree structure of the HTML, allowing us to locate specific elements like book titles and prices.

3. **Step 3: Extracting Data**
    - Using BeautifulSoup’s methods like find_all and find, the script identifies and extracts:
        - Book titles from the title attribute of ” <a> tags” inside “<h3> tags”.
        - Prices from the “<p> tag” with the class price_color.
    - The extracted data is stored in two separate Python lists btitles and bprices.
4. **Step 4: Structuring Data**
    - The extracted lists are combined as a DataFrame.
    - Each book's title and price is correspond to a row in the DataFrame, making the data easily accessible for analysis.
5. **Step 5: Displaying the Results**
    - The final DataFrame is printed to the output terminal, showing the structured data scraped from the website.