Web Scrapping

Web scraping is the process of extracting and parsing data from websites in an automated fashion using a computer program. It's a useful technique for creating datasets for research and learning. Follow these steps to build a web scraping project from scratch using Python and its ecosystem of libraries.

Guidelines for Web Scrapping

Pick a website and describe your objective:

1. Browse through different sites and pick on to scrape.
2. Identify the information you'd like to scrape from the site. Decide the format of the output CSV file.
3. Summarize your project idea and outline your strategy in a Juptyer notebook.

Use the requests library to download web pages:

1. Inspect the website's HTML source and identify the right URLs to download.
2. Download and save web pages locally using the requests library.
3. Create a function to automate downloading for different topics/search queries.

Use Beautiful Soup to parse and extract information:

1. Parse and explore the structure of downloaded web pages using Beautiful soup.
2. Use the right properties and methods to extract the required information.
3. Create functions to extract from the page into lists and dictionaries.
4. (Optional) Use a REST API to acquire additional information if required.

Create CSV file(s) with the extracted information:

1. Create functions for the end-to-end process of downloading, parsing, and saving CSVs.
2. Execute the function with different inputs to create a dataset of CSV files.
3. Verify the information in the CSV files by reading them back using Pandas.

Document and share your work:

1. Add proper headings and documentation in your Jupyter notebook.
2. Publish your Jupyter notebook to your Github profile.
3. (Optional) Write a blog post about your project and share it online.