
HTML extractor

Description

Step 1: This script enables you to retrieve the HTML code and all its links from an arbitrary website. [su_tooltip title="Limitations" text=" Please note that some websites/servers implement security measures that prevent scraping. Therefore, the script may not work on all websites." shadow="yes"]?[/su_tooltip]

Step 2: After collecting the links from the HTML code obtained in Step 1, you can extract the code from all the links and merge it into one aggregated code block.

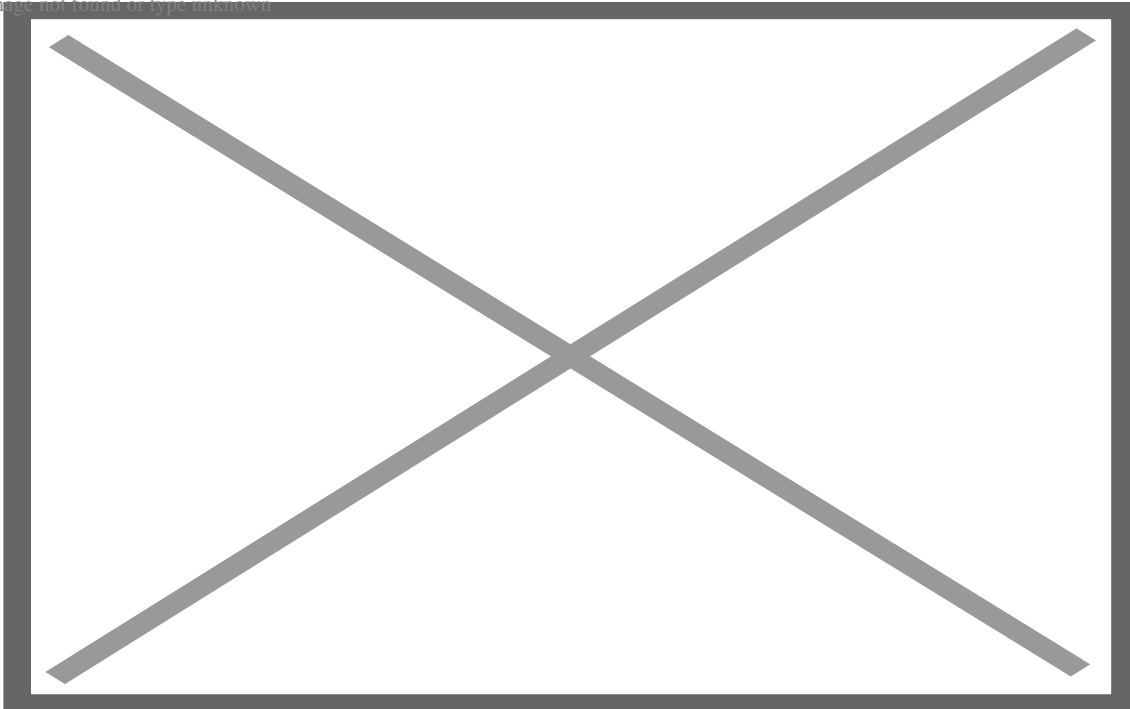
Step 3: In the final step, you can automatically extract all email addresses found in the merged code block obtained in the previous step.

* * *

This script thus allows you to easily extract all email addresses from a webpage and its associated deep links. This is achieved via PHP and JavaScript, i.e., mainly via [DOMXPath::query\(\)](#) and [preg_match\(\)](#). You can then use [Proton Mail](#), for example, to send emails anonymously to the extracted addresses. Of course, VPN and/or [TOR](#) are always recommended when anonymity is important. [Tails OS](#) is even better for anonymous web browsing. No data is stored server-sidedly by the script.

[su_spoiler title="Infographic & usage scenarios" style="fancy" icon="folder-2? anchor="htmlv"]

Image not found type unknown



Usage scenarios

Web Content Aggregation: This script allows you to combine HTML content from multiple web pages into a single code block. This can be useful for creating aggregated content, such as news summaries or data reports from multiple sources.

Data Mining and Analysis: You can use this script to gather data from different web pages and analyze it. For example, you could extract product information from multiple e-commerce sites for competitive analysis.

Email Address Extraction: By combining HTML code from various sources, you can then extract email addresses. This can be handy for building email lists or identifying contact information on websites.

Web Monitoring: You can set up a monitoring system to track changes on specific web pages. By collecting and comparing HTML content, you can detect updates or modifications to websites.

Content Archiving: This script can be used to create archives of web pages. You can periodically collect and store HTML code from different versions of a website to maintain a historical record.

Competitive Intelligence: For businesses, this script can help gather information on competitors' websites, including their products, prices, or news updates.

Research and Data Collection: Researchers can use it to collect data from various sources, such as academic papers, news articles, or social media, for analysis and study.

Automated Testing: In software development, this script can be integrated into automated testing processes to extract and compare HTML code for quality assurance.

Remember that web scraping should always be done within the legal and ethical boundaries of a website's terms of service and relevant laws. Additionally, the purpose of your scraping should respect privacy and data protection regulations. [/su_spoiler] [su_accordion] [su_spoiler title="Step 1: Extract HTML code and all links from a website" open="no" style="default" icon="plus" anchor="" class=""] Enter the URL of a website to extract its HTML code and all links found therein.

[/su_spoiler] [su_spoiler title="Step 2: Extract HTML from links found in Step 1? open="no" style="default" icon="plus" anchor="" class=""] Paste the links obtained in Step 1 in order to extract their respective HTML code (this process might take some time).

[/su_spoiler] [su_spoiler title="Step 3: Extract Email addresses from HTML" open="no" style="default" icon="plus" anchor="" class=""] Paste the HTML code obtained in Step 2 to extract all email addresses. You can then easily send an email to all addresses.

[/su_spoiler] [/su_accordion]

Date Created

17. October 2023

Author

web45