# Paper Information Extraction from HTML Files:

This Python script extracts information from HTML files that contain data related to research papers. The extracted information includes the paper's title, author names, author email, author address, abstract, and keywords.

—----------

## Prerequisites:

Before using this script, make sure you have the following installed:

- Python
- BeautifulSoup ("pip install beautifulsoup4")

—----------

## Usage

1. download this repository to your local machine.
2. Place the HTML files that you want to process in the directory and update the file paths in the code.
3. Open a terminal or command prompt and go to the script's directory.
4. Run the script by the following command: python modularAS.py

This will extract the information from the HTML files and save it to multiple TSV (Tab-Separated Values) files.

—----------

## RAID Methods for Data Storage

For storing the extracted data, you may consider using different RAID (Redundant Array of Independent Disks) methods. Here's a brief overview of some common RAID levels and a recommendation for this project:

RAID 0 (Striping): This RAID level offers increased storage performance by striping data across multiple drives. However, it provides no redundancy, which means data loss in case of a drive failure.

RAID 1 (Mirroring): RAID 1 duplicates data across two drives, providing data redundancy. While it's a secure option, it uses twice the storage capacity for the same amount of data.

RAID 5 (Block-Level Striping with Distributed Parity): RAID 5 combines striping with distributed parity, providing both performance and redundancy. It requires a minimum of three drives.

RAID 10 (Striped Mirroring): RAID 10 combines the features of RAID 1 and RAID 0. It mirrors data while striping for improved performance and redundancy.

## Recommendation

For this project, where data extraction and storage are critical, it's recommended to use RAID 5. RAID 5 provides a balance between performance and data redundancy, making it suitable for storing the extracted paper information. you need at least three drives to implement RAID 5 effectively.

—----------

## Functions

- "read_html_file(file_path)": Reads an HTML file and returns its content.
- "extract_title(soup)": Extracts the paper title from the HTML using BeautifulSoup.
- "extract_author_names(soup)": Extracts author names from the HTML.
- "extract_author_email(soup)": Extracts author emails from the HTML.
- "extract_author_address(soup)": Extracts author addresses from the HTML.
- "extract_abstract(soup)": Extracts the paper's abstract from the HTML.
- "extract_keywords(soup)": Extracts keywords from the HTML.
- "process_html_file(file_path)": Processes an HTML file to extract all relevant information.

- "save_paper_data(paper_data, data_files)": Saves extracted data to multiple TSV files for redundancy.

—----------

## Example

The "main" function in the script demonstrates how to use the provided functions to process HTML files and save the data to TSV files.

—----------

## Authors

- Reihaneh Maarefdoust