

SI100B-23Spring-Project

Part 0: Set Up Environment

- Python Version: 3.7+
- Third Party Library: `pip install -r requirements.txt`

Part 1 (20 points): Official Accounts Crawling

Background

Web scraping is an essential skill for extracting data from websites. It allows you to collect and process vast amounts of information quickly, which can be useful for research, data analysis, and many other applications. In the file processing module, you may often need to acquire data from various sources, and web scraping is an efficient way to achieve that.

https://mp.weixin.qq.com/mp/appmsgalbum?&action=getalbum&album_id=1842688941473251331&scene=173

You can run `python Part1_OfficialAccountsCrawling\mainCrawl.py` to see the result.

This project aims to provide an introduction to web scraping for students with a non-computer science background, focusing on the importance of web scraping in daily learning activities, especially in the context of a file processing module. By completing this project, you will learn how to extract valuable information from web pages using Python, which can be helpful in a variety of real-world scenarios.

Background

Web scraping is an essential skill for extracting data from websites. It allows you to collect and process vast amounts of information quickly, which can be useful for research, data analysis, and many other applications. In the file processing module, you may often need to acquire data from various sources, and web scraping is an efficient way to achieve that.

Project Description

In this project, you will build a simple web scraper to extract information from WeChat official accounts. The scraper will obtain information such as article titles, authors, and publication times.

To complete this project, you need to accomplish the following tasks and understand the purpose of each function:

1. `Change_Params(msgid)` : Update the `params` dictionary with a new `msgid` .
2. `Author_Time_Crawl(page_url)` : Write a function that sends an HTTP GET request to the given URL and parses the HTML content to extract the author and time of the article.
3. `First_Crawl()` : Write a function that sends an HTTP GET request to the first URL (the article collection page) and extracts the first article's title, data-msgid attribute, and URL.
4. Main web scraping logic: Call the `First_Crawl()` function to get the first article's data-msgid, then repeatedly send HTTP GET requests to the `base_url` with updated `params` until the last article is reached. For each article, call the `Author_Time_Crawl()` function to extract the author and time information and print the result.

Implementation Details

1. In the `Change_Params()` function, you need to update the `params` dictionary with the new `msgid`. You can achieve this by setting `params["begin_msgid"]` to the new `msgid`. This function is responsible for updating the parameters used in the HTTP GET request.
2. In the `Author_Time_Crawl()` function, use the `requests` library to send an HTTP GET request to the provided URL. Then, use the `BeautifulSoup` library to parse the HTML content and extract the author and time of the article. This function focuses on extracting the desired information from individual articles.
3. In the `First_Crawl()` function, send an HTTP GET request to the first URL and parse the HTML content using `BeautifulSoup`. Extract the first article's title, data-msgid attribute, and URL, and call the `Author_Time_Crawl()` function with the extracted URL to obtain the author and time information. This function initiates the web scraping process by starting with the first article.
4. In the main part of the script, start the web scraping process by calling the `First_Crawl()` function and update the `params` with the new `msgid`. Send HTTP GET requests to the `base_url` and extract the required information for each article using the `Author_Time_Crawl()` function. The main logic drives the entire web scraping process and orchestrates the functions mentioned above.

Example Output

```
以高质量本科专业建设促创新人才培养  
综合办公室 2023-02-16 22:53:41  
  
喜讯 | 上科大本科生团队获国际超算大赛 IndySCC线上赛道冠军  
信息学院 2022-11-14 23:06:20
```

Todo List:

1. Crawl all articles by `title(6')`, `author(4')` and `time(8')`.
2. Save the crawled information into `excel(6')` in the `specified form(6')`.

Part 2 (35 points): Stack Files Processing

Background

In this project, you will work with different tasks related to processing files and folders. Each task will have different requirements and complexity levels. You will be asked to handle files, folders, and compressed files, as well as perform word counting for text files. In order to complete this project, you'll need to fill in the missing parts of the provided code. The main goal is to process a set of disorganized folders containing files, subfolders, and compressed files, and extract all the contents.

Project Description

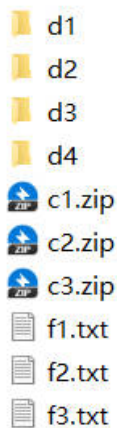
This script defines a set of classes and functions to handle file and folder operations, including compressed files. The script can be broken down into three sections:

`tools.py` - This file contains utility functions for working with paths, files, and folders. It includes functions for checking if a path exists, if it's a folder, or if it's a compressed file, as well as functions for manipulating paths and copying files.

compressed_file.py - This file defines the CompressedFile class, which inherits from the File class and provides functionality for handling compressed files. It includes methods for decompressing compressed files in various formats (zip, tar, rar, gz, 7z), although some of these methods are not yet implemented. When a CompressedFile object is created, it checks if the provided path is a valid compressed file and initializes the necessary attributes such as file name, path, password (if applicable), and compression method.

file.py - This file defines the File class, which is a base class for working with files. It includes methods for transferring files to a new location, and it provides a string representation of the file object for easy printing.

folder.py - This file defines the Folder class, which provides functionality for working with folders. It includes methods for unpacking the folder's contents, including decompressing any compressed files and recursively processing any subfolders. The Folder class also has a method for counting the occurrences of words in text files within the folder.



Todo List:

Task 1 (7 points): Handle only files (no word counting required).

Task 2 (7 points): Handle only files and perform word counting for text files.

Task 3 (7 points): Handle files and single-layer folders (word counting required).

Task 4 (7 points): Handle files, single-layer folders, and compressed files (with either no contents or only files inside). Perform word counting for text files.

Task 5 (7 points): Using all above words counting, find the passwords to solve the password-protected compressed file.

Part 3 (15 points): Functional Perfection

As we can see, the part two's function is not abundant enough, we need to make it more perfect by doing these following tasks. And you may get extra points by finishing the following tasks:

Task 6 (7 points): Handle multi-layer folders (word counting required).

Task 7 (8 points): Handle multi-layer folders and compressed files (which may contain files, folders, and compressed files). Perform word counting for text files.

Part 4 (30 points): Live coding

After the project submission, we will have a live coding session with all the teams, where you will use the code you wrote for parts 1-3 to solve a new problem. Make sure you're familiar with the code you submit and the libraries you used, and also make sure you're comfortable with the basic Python programming techniques we showed in lecture.

SI100B-23Spring-Project

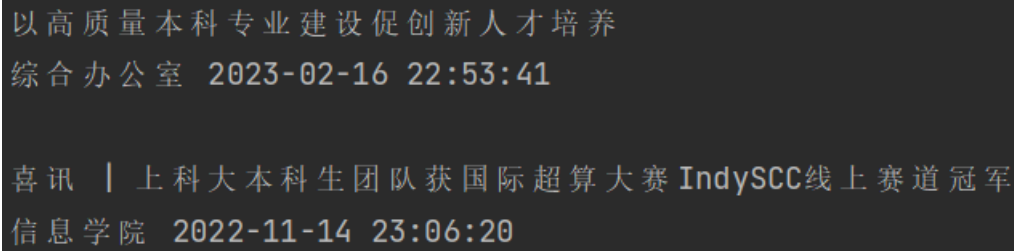
Part 0: Set Up Environment

- Python Version: 3.7+
- Third Party Library: `pip install -r requirements.txt`

Part 1 (20 points): Official Accounts Crawling

https://mp.weixin.qq.com/mp/appmsgalbum?&action=getalbum&album_id=1842688941473251331&scene=173

You can run `python Part1_OfficialAccountsCrawling\mainCrawl.py` to see the result.



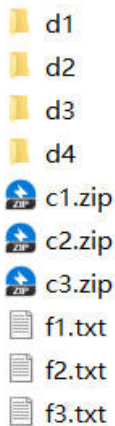
以高质量本科专业建设促创新人才培养
综合办公室 2023-02-16 22:53:41

喜讯 | 上科大本科生团队获国际超算大赛 IndySCC线上赛道冠军
信息学院 2022-11-14 23:06:20

Todo List:

1. Crawl all articles by `title(6')`, `author(4')` and `time(8')`.
2. Save the crawled information into excel(6') in the `specified form(6')`.

Part 2 (35 points): Stack Files Processing



d1
d2
d3
d4
c1.zip
c2.zip
c3.zip
f1.txt
f2.txt
f3.txt

Part 3 (15 points): Functional Perfection

Part 4 (30 points): Live coding
