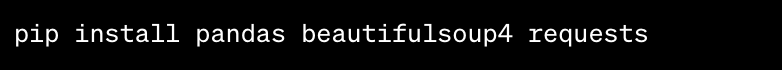
**Instructions to Run the Python Data Extraction Script:**

**1. Installing required dependencies:**

Before running the script, ensure you have the required dependencies installed. You can install them using the following:



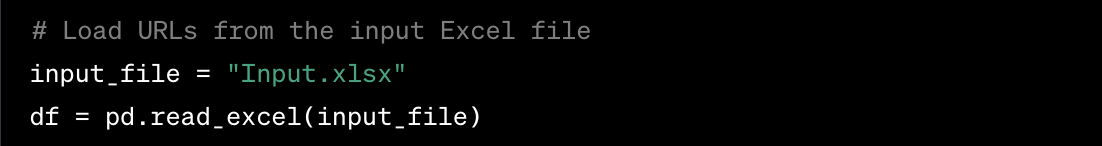
**2. Import Libraries:**

Import necessary libraries (pandas, BeautifulSoup, requests).



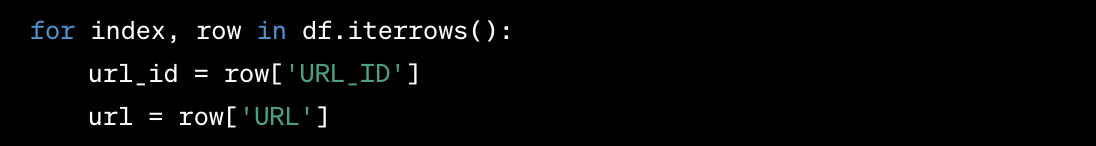
**3. Load URLs from Input Excel File:**

Read the input Excel file (Input.xlsx) using pandas.



**4. Iterate Over Each Row in DataFrame:**

For each row in the dataframe, extract URL\_ID and URL.



**5. Fetch HTML Content from URL:**

Use the requests.get method to fetch the HTML content from the specified URL.



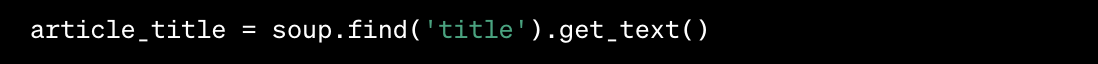
**6. Parse HTML Content:**

Parse the HTML content using BeautifulSoup to create a soup object.



**7. Extract Article Title:**

Find and extract the title of the article using soup.find('title').get\_text().



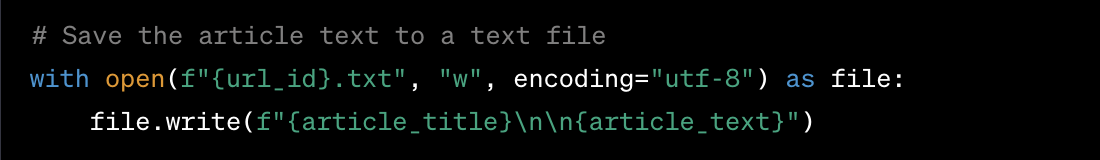
**8. Extract Article Text:**

Find all paragraph (<p>) tags and join the text inside them to extract the article text.



**9. Save Article Text to Text File:**

Write the article title and text to a text file named {url\_id}.txt.

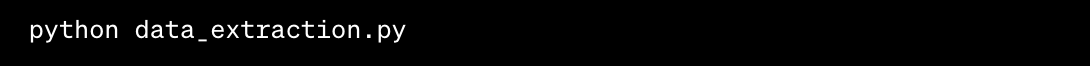


**Dependencies:**

* **pandas**: For data manipulation and handling.
* **beautifulsoup4**: For HTML parsing.
* **requests**: For making HTTP requests.

**Run the Script:**

1. Open a terminal or command prompt.
2. Navigate to the directory containing the script.
3. Run the script using the following command:



**Instructions to Run the Python Text Analysis Script:**

**Step 1: Install Dependencies**

Before running the script, make sure you have the required libraries installed. Open your terminal or command prompt and execute the following commands:



This will install the necessary libraries for data manipulation (pandas), natural language processing (nltk), and syllable counting (pronouncing).

**Step 2: Download NLTK Resources**

The script uses the NLTK library, and it requires additional resources for tokenization. Uncomment the following line in your script:



After uncommenting, save the script. This step needs to be done only once.

**Step 3: Prepare Input Data**

Create an input Excel file (Input.xlsx). Ensure it contains at least two columns: 'URL\_ID' and 'URL'. Each row should represent a unique URL you want to analyze.

**Step 4: Prepare Master Dictionaries**

Ensure you have two master dictionaries:

* **Positive words**: MasterDictionary/positive-words.txt
* **Negative words:** MasterDictionary/negative-words.txt

These dictionaries should contain a list of positive and negative words, respectively, with each word on a new line.

**Step 5: Prepare Stop Words**

In the 'StopWords' directory, include custom stop word files. Each file should follow the naming convention "StopWords\_something.txt". Words in these files will be excluded during the analysis.

**Step 6: Define Functions**

* **load\_dictionary**: Loads a set of words from a given file.
* **load\_stop\_words**: Loads custom stop words from files in a specified directory.
* **clean\_text**: Tokenizes and removes custom stop words from the text.
* **calculate\_sentiment\_scores**: Calculates positive, negative, polarity, and subjectivity scores.
* **calculate\_readability**: Calculates various readability metrics using word and sentence counts.
* **calculate\_personal\_pronouns**: Counts personal pronouns (I, we, my, ours, us) in the text.
* **calculate\_avg\_word\_length**: Calculates the average word length in the text.
* **analyze\_text\_file**: Orchestrates the analysis for a given text file.

**Step 7: Prepare Text Files**

Create text files for each URL\_ID, named as "URL\_ID.txt". Each file should contain the text you want to analyze.

**Step 8: Define the Main Function**

* **main()**: The main function that orchestrates the entire process.
* Loads positive and negative dictionaries, custom stop words, and input URLs.
* Iterates through each URL, reads the associated text file, and analyzes it.
* Appends the analysis results to a list.
* Converts the list to a DataFrame and saves it to an Excel file.

**Running the Python Script:**

1. **Navigate to Script Directory:**

Open a terminal or command prompt and navigate to the directory where your text\_analysis.py script is located using the cd command.

1. **Run the Script:**

Execute the script using the following command:



or, if you're using Python 3:



1. **Wait for Execution:**

The script will start running, and you'll see the output on the console. Depending on the script's complexity and the amount of data being processed, it may take some time to complete.

1. **Check Output:**

Once the script completes its execution, check the specified output file (in this case, "Output.xlsx") for the results.

**Dependencies :**

* **os**: Provides a way of using operating system-dependent functionality.
* **pandas**: Library for data manipulation and analysis.
* **nltk**: Natural Language Toolkit library for natural language processing tasks.
* **codecs**: Module providing support for reading and writing encoded data.
* **pronouncing**: Library for counting syllables in words.