Web-Scraping Tool README

National Renewable Energy Laboratory, DOE SULI Program
Written by: Lawrence Chen

Setup

- The code requires around 1 hour for 33 chemicals. It is important that the computer that is running the code stays on for the duration of this time. Therefore, alterations to screen saver time settings may be necessary to ensure the code doesn't stop running.
- Downloads:
    - https://googlechromelabs.github.io/chrome-for-testing/ Download Chromedriver using the corresponding zip file below
    - This code uses Jupyter Notebook, Python.
    - Python packages used: pandas, numpy, selenium, webdriver_manager, datetime, requests, re, time, random, math, babel
- Note: The code will open browsers up, such that multiple chrome tabs will open. Headless browser (meaning no chrome tabs will be seen) is an option, but will take significantly longer and will be less reliable.

Code Summary

1. (Code Notebook) takes in an excel file that contains a sheet currently labeled "search_terms" and within that sheet has a column containing the list of chemicals you want to pull results from. The cell creates search terms that can be inputted into google search.
    - Alternative: Define a list of chemical search terms labeled search_terms_list.

2. Generates a tuple of website links to scrape from as well as the corresponding search term that was used to find that site.
3. Scrapes from all the sites using helper functions defined previously. A dataset is produced.
4. Dataset is modified and ready for analysis.

Column Explanations

Final Product: final_df Dataframe.

Some columns includes are not useful (e.g. Price, as USD_Price standardizes for currencies) but are intermediary columns used to produce other columns in the dataset. They are not bolded and labeled with (I), feel free to filter them out when modifying this dataset.

1. **Chemical Name: as titled.**
2. **Search Term: strings that are inputted (searched) into google that return the following website link.**
3. **Website: website link.**
4. **Parent Site: chemical distributor.**
5. **Product Name: product sold by chemical distributor.**
6. **Units: number of products being sold. (e.g.: I'm buying 1 unit of 500mL C60).**
7. Price: (I) raw price collected from site.
8. **USD_Price: number from price (col. 7), standardized.**
9. Currency: (I) original currency form ($, etc.) in col.7.
10. usd: (I) booleans (1 if the currency is already in usd, 0 if the currency is not).

11. category_1_name: (I) among sites that have additional dropdowns like "Quantity", and which vary prices based on this dropdown: we collect the 1st dropdown name.
12. category_1_value: (I) values listed in the 1st dropdown.
13. category_2_name: (I) among sites that have additional dropdowns like "Quantity", and which vary prices based on this dropdown: we collect the 2nd dropdown name.
14. category_2_value: (I) values listed in the 2nd dropdown.
15. quantity_value: (I) original quantity collected. (e.g. 50mg)
16. quantity_number: (I) number in col.15. (e.g. 50)
17. quantity_units: (I) units in col.15. (e.g. mg)
18. **standardized_units_l_g: col.16 number but standardized into liters or grams.**
19. **q_units_standardized: col.17 units standardized into either L or G.**