

Are libraries open right now?

Introduction

Almost every week, my kids ask us this particular question: is library open right now? In this project, I decided to design a program which can provide an answer to this particular question. By employing web scraping techniques provided by BeautifulSoup, data wrangling with pandas, and with the use of the datetime module, I successfully created a program called “webscraper_libraryopentime.py” to provide us a definitive answer to our question.

Approaches

To approach this project, I first have to decide the functional requirements. There are two functional requirements:

1. The program has to determine whether any library is open at the moment.
2. The names of libraries which are open.

To implement these requirements, I first limited the scope of my project to local libraries. In this particular case, I decided that my project will only be for all 8 local JMRL branches. In the coding phase, I split this project into three blocks:

1. Web scraping all 8 libraries, and locate and download information which contains operational hours of all libraries.
2. Clean up these information, and store these information into a .csv file
3. Retrieve the current time, and employ our newly created .csv file to determine whether any libraries are open, and provide a list of libraries which are open.

Results

These three blocks of coding in principle is straightforward, it turned out to be a much harder task than I originally imagined. The problem is that almost each branch uses a slightly different delimiter to display their hours. For example, the central and Gordon libraries have their hours displayed, respectively:

Hours:

Monday - Thursday : 9am - 9pm
Friday - Saturday : 9am - 5pm
Sunday - 1pm - 5pm

(Closed Sundays Memorial Day weekend to Labor Day)

Monday: 9am - 9pm | Tuesday: 9am - 6pm
Wednesday: 12 -9pm | Thursday: 10am - 6pm
Friday - Saturday: 10am - 5pm
Sunday - Closed

Central Branch

Gordon Library

I eventually have to employ the split function with the regular expression. Eventually I defined a function called `scraping_info(url)` to accomplish the web scraping part of design. This function takes url as input, and gives the return data “info” which is a list of two elements. The `info[0]` is a string of the library name, and `info[1]` is a list of data which contains operational hours.

The 2nd block is the data cleaning. The immediate problem was that some weekdays were not explicitly listed, for example Tuesday could be listed as Monday-Wednesday. I reasoned that I can presume the value of later weekdays will be same as the preceding weekdays if not explicitly listed. This is accomplished by the function `getweeklytime(hourinfo)`. It takes the input from the webscraping output `info[1]`, and returns a list of 7 elements (for each weekday). Each element is a list of two float numbers indicating opening and closing hours. Since there is one library that opens at half hours, I have to employ float numbers for this exact reason. In addition, I have to construct another function `hour_range(item)` to convert 30 minutes into 0.5 hours, as well as to convert pm hours into 24 hour expressions. This function takes input of a list of numbers, and returns a list of two float numbers as the opening and closing times.

To conclude the 2nd block, I saved both the name of libraries and their according opening and closing hours into a .csv file called `libraryhour.csv`.

The last block is relatively straightforward. It uses pandas to load `libraryhour.csv` to a dataframe. With the help of datetime module, I was able to retrieve the exact time and the weekday. After locating the exact columns of hours to compare to by the weekday value, we were able to make recursive comparisons to determine whether the library is open, and which ones are open. In this particular block, I used `df.iat[i,j]` to locate an individual element in a pandas dataframe with its indexes.

Conclusions and Outlook

This program is quite practical. Even in this primitive form, the program gives a definitive answer about whether any area libraries are open at the moment. If anyone wants to have a quiet study time in any of these libraries, or just want to use the internet services provided by these libraries, they obviously can use this program to decide which ones are open, and pick one then go.

There are multiple ways which can improve this program. A few examples:

1. A map function. It will allow users to immediately locate the library and locate the closest branch.
2. A link to their book inventories. If users want to borrow a particular book which only a subset of libraries has, a linked inventory will help user to select the right one.
3. I also did not add official holidays in the excluded list of closing dates for libraries, obviously fixing it will enhance the usability.
4. Additionally, a natural extension of this program will be to include additional libraries, i.e., increase the size of the database.

The attached references are probably only a small subset of what I have perused.

References:

1. <https://jmrl.org/>
2. <https://stackoverflow.com/>
3. <https://beautiful-soup-4.readthedocs.io/en/latest/>