*****THE QUESTION THAT I WILL BE ATTEMPTING IS QUESTION 1*****

Please open the .ipynb file named Final Assessment Code.ipynb on jupyter notebook.

Before beginning on the web scrapping process, we would have to install the necessary parsers, packages etc.

Also note that for some bigger websites like YouTube or Facebook etc., they might have their own public API for scrapping of their data. Some big websites are against the idea of using manual means to scrape their data so just watch out for that because we do not want to get into any trouble!

****

However, one big issue with web scrapping is that there is no universal solution for web scraping because the way data is stored on each website is usually specific to that site. In fact, if you want to scrape the data, you need to understand the website's structure and either build your own solution or use a highly customizable one.

Hence, keeping that in mind, I do not aim to present you with a universal web scrapping program, but rather I would present to you my thought process and the different steps that I take to successfully scrape data off a website.
****

For this problem, we assume that the website has no public API to use to scrape their data. My approach to this problem is basically to inspect the website and the HTML code on how the website has saved its data, and then extract valuable information that I want and then save it into a csv file.

Refer to the .ipynb file named **Final Assessment Code.ipynb** for the code that is translated from the steps below. Also note that I will be using a mac so slight variations in code might apply.

Initialization
Step 1: Launch mac terminal

#when we run pip with sudo, it also means we run setup.py with sudo.
# (aka running Python code from the internet as the root)
Step 2: Enter "sudo pip install bs4", key in password whenever the terminal prompts

Step 3: Enter "sudo pip install lxml", key in password whenever the terminal prompts Step 4: Enter "sudo pip install requests", key in password whenever the terminal prompts

#now that all our required packages and parsers are installed successfully, we can go ahead and launch python.

Step 5: Key in "python" in the terminal and you should see this.

```
Chuas-MacBook-Air-3:~ huarenxn$ python
Python 3.6.4 |Anaconda, Inc.| (default, Jan 16 2018, 12:04:33)
[GCC 4.2.1 Compatible Clang 4.0.1 (tags/RELEASE_401/final)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
#to check if python is working, simply do something like key in 2+2 in the console and yes, it should return the output of 4.
```

Step 6: Launch Jupyter notebook. We will be using that for our code editor.

Step 7: Import all the relevant packages that we have installed earlier.

Now that we have initialized the environment, it is time to move on to the body of the program.

Body
Step 1: Create a new csv file which will be used to save the information that we scrape from the web

Step 2: Save the URL link into a variable as a string.

Step 3: We shall use for loops to iterate and gather all the information that we want. Next question now is, where and what information do we want? From the task, we want to find out famous spots where movies are filmed, eg. the names of the places.

Step 4: Inspect the website. (For mac users, double click the browser page and click "Inspect".)

Step 5: Look over the HTML code and find out which part of it contains the name of the famous spots.

Step 6: Request for the URL and parse it in the lmxl format. (refer to actual code) Step 7: Select the part of the HTML code that nests the information that we want.

Step 8: We can now decide what we want to do with the data. For example, we can store it in a csv file, an excel spreadsheet, or even just a simple list.

Conclusion

Ultimately, we realise that every website is unique and has its information stored in different formats. In my short 48hours I have only explored these few formats, and I strongly believe that there is way more formats for me to uncover and explore.