Homework 7

Web scrapping

Given March 5th – Due March 17th

Exercise 1. I'm feeling lucy Google search

Whenever I search a topic on Google, I don't look at just one search result at a time. By middle-clicking a search result link (or clicking while holding ctrl), I open the first several links in a bunch of new tabs to read later. I search Google often enough that this workflow—opening my browser, searching for a topic, and middle-clicking several links one by one—is tedious. It would be nice if I could simply type a search term on the command line and have my computer automatically open a browser with all the top search results in new tabs. Let's write a script to do this.

This is what your program should do:

- Gets search keywords from the command line arguments.
- Retrieves the search results page.
- Opens a browser tab for each result.

This means your code will need to do the following:

- Read the command line arguments from sys.argv.
- Fetch the search result page with the requests module.
- Find the links to each search result.
- Call the webbrowser.open() function to open the web browser.

Save your program as *lucky.py*.

Tips:

- Sometimes the html response we get is different from the browser (because we are not a browser). One thing you could do is provide a User Agent as follows. But, still you could get different html pages with that too.

```
userAgent = "Mozilla/5.0 (X11; CrOS i686 2268.111.0) AppleWebKit/536.11
(KHTML, like Gecko) Chrome/20.0.1132.57 Safari/536.11"
headers = {'User-Agent': userAgent}
res = requests.get(url, headers=headers)
```

- If you are still getting different html pages, you could save your html response of requests module to a file. Then, open that file with your browser and inspect that.

- By looking at the browser's address bar after doing a Google search, you can see that the result page has a URL like https://www.google.com/search?q=SEARCH_TERM_HERE
- After doing a Google search for Beautiful Soup, you can open the browser's developer tools and inspect some of the link elements on the page. They look incredibly complicated. It doesn't matter. You just need to find the pattern that all the search result links have.
- You will probably need to do a combination of two types:

Table 11-2: Examples of CSS Selectors

Selector passed to the `select()` method	Will match . . .
`soup.select('div')`	All elements named `<div>`
`soup.select('#author')`	The element with an `id` attribute of `author`
`soup.select('.notice')`	All elements that use a CSS `class` attribute named `notice`
`soup.select('div span')`	All elements named `<span>` that are within an element named `<div>`
`soup.select('div > span')`	All elements named `<span>` that are *directly* within an element named `<div>`, with no other element in between
`soup.select('input[name]')`	All elements named `<input>` that have a `name` attribute with any value
`soup.select('input[type="button"]')`	All elements named `<input>` that have an attribute named `type` with value `button`

So something like soup.select('input[name] > span')

Exercise 2. Image Site Downloader

Write a program that goes to a photo-sharing site like Flickr or Imgur, searches for a category of photos, and then downloads all the resulting images.

Tips:

- Being "url" the url of an image you may want to use the following code to download an image. Note that this method will download the image in the current directory.

```
from urllib.request import urlretrieve

urlretrieve(url, "filename.jpg")
```

If the exercise does not say which name to save the file, save your code files as *hm7_name_surname_ex_num.py*, where *num* is the exercise number 1, 2, etc.

Comment everything so we know you wrote the code! On top of your files write this multiline comment with your information:

”””

Homework 7, Exercise 1 (or 2…)
Name
Date
Description of your program.

”””