SI 206 – Data-Oriented Programming Project 2 – Fall 2018

OBJECTIVES:

This assignment will help you gain more experience with Beautiful Soup and processing html files.

FILES:

- The *project2.py* file, which contains some provided code and tests
- *news1.html* – used for testing – a saved version of https://www.michigandaily.com/section/news
- *newsStory1.html* – a saved version of https://www.michigandaily.com/section/ann-arbor/panel-discusses-pros-cons-library-lot-ballot-proposal

WHAT YOU SHOULD SUBMIT TO CANVAS:

- A link to your github account to this project – fill out the top part of the *project2.py* file

Your .py file must run successfully using Python3. We do not grade projects that do not run. Make sure you check that your program runs before submitting it!

TOTAL POSSIBLE POINTS: 300

The project is in multiple parts but each part builds on both the homework you have done previously.

The file also includes unittests, which are the primary determining factor as to how you get points (but we do look at your code, etc).

Make sure your file runs without syntax errors!

You should start this early! Break it up into pieces, write out a plan, do a little bit at a time, and keep running and testing it, adding and committing and pushing to GitHub.

Part 1 –Grab Most Read Headlines

- Complete a function **grab_headlines**.
- **INPUT: soup –** the soup object to work with
- **RETURN VALUE:** a list of strings that represent the headlines of all of the Most Read articles from the Michigan Daily News Page at https://www.michigandaily.com/section/news
- Since the data returned by this text may change daily, you may use the data file *news1.html* to test your function. This is a unittest that tests this function.
- **Successful unit test: 100 points**

Process:

- Open news1.html in a browser window
- Inspect the headlines (there should be a way to inspect the HTML in your browser). Look at the HTML around the headlines to figure out how to find them. Often you need to start with an enclosing tag and work from there.

- Check that the way that you found the headline doesn't find something that you don't want.
- Use beautiful soup to get the headlines and return them as a list of strings.

Part 2 – Return News Headline Dictionary

- Complete the function **get_headline_dict**
- **INPUT: soup** – the soup object to work with
- **RETURN VALUE:** a dictionary with the headlines as the keys and the url for each headline story as the value from the Michigan Daily News Page at https://www.michigandaily.com/section/news
- Since the data returned by this text may change daily, you may use the data file *news1.html* to test your function
- **Successful unit test: 100 points**

Part 3 - Get Page Info

- Complete the function **get_page_info**
- **INPUT:** the **soup** object to work with
- **RETURN VALUE:** a tuple of the story title, date published, author, and the number of paragraphs in the story from one of the full stories that were linked on the https://www.michigandaily.com/section/news page
- Since the data returned by this text may change daily, you may use the data file *newsStory1.hgml* to test your function
- **Successful unit test: 100 points**

Extra Credit – Complete the **find_Mich_stuff** function which must use a regular expression to return a new dictionary with just the items found in part2 that have "U-M" or "Ann Arbor" in the headline. – 10 points