Advanced Web Development and Web Scraping
Fall 2018
Assignment #7 – Web Scraping Assignment

Note: For all assignments, *get* the web page using python's request library, and include an appropriate header in the request.

1. Scrape the sample Schedule page (https://gdancik.github.io/CSC-360/data/notes/schedule.html) to output the number of courses being taught and the *total* number of credits, in the following format:

   ```
   Dr. Dancik is teaching 4 courses (12 credits)
   ```

   Note: Your scraper should work for different data (e.g., a different instructor, or a different number of courses. However, the page will always have the same format (e.g., There will be a single table with the list of courses, with columns for Course, Time, and # Credits, in that order).

   Hint #1: In order to extract the instructor name, I recommend using the *split* method (https://www.w3schools.com/python/ref_string_split.asp)

   Hint #2: In order to add the credits, you will need to convert each 'number' to an integer, using the *int* function, e.g., `int('3')` will return 3.

2. Scrape the CS faculty page (http://www.easternct.edu/computerscience/faculty/) to output each faculty member's name (from the table row with the blue background) and office location. Output should appear in the format below:

```
Name                    Office
Dr. Garrett M. Dancik   Science Bldg 257
Dr. Kehan Gao           Science Bldg 254
Tim Hartley             Science Bldg 165
Dr. Jian Lin            Science Bldg 252
Dr. Joel Rosiene        Science Bldg 256
Dr. Sarah Tasneem       Science Bldg 253
Dr. Huan-Yu (Alan) Tu   Science Bldg 255
Ms. Cheryl Le Beau      Science Bldg 168
```

Hint #1: What attribute can you use to identify the faculty tables? For each of these tables, you can get the faculty member's *name* by looking at the first *td* element of the first row. What row number contains the office location?

Hint #2: To format the output, use the following print statement, which uses an *f string* (or formatted string), where "Name" and "Office" are used to specify the column headings, or are replaced with the variables storing the faculty member's name and office location:

```
print(f'{"Name":20}\t{"Office":15}')
```

The *f* is used to denote a formatted string. Strings or variables are placed in curly braces, with colons used to denote the width. In the example above,

```
{"Name":20} – outputs "Name", and will be padded with
              blank spaces so that the width is 20 characters.

      \t    – insert a tab stop between the columns

{"Office":15} – outputs "Office", and will be padded with blank
              spaces so that the width is 15 characters
```

Replace "Name" or "Office" with the name of a variable to output the value of the variable instead.

3. Scrape the title and rating for 5 movies from IMDB, whose links are given below, and construct a bar graph that shows the rating for each movie. Give your graph an informative title. The following links should be used:

    - https://www.imdb.com/title/tt0109830/?ref_=fn_al_tt_1
    - https://www.imdb.com/title/tt0076759/?ref_=fn_tt_tt_1
    - https://www.imdb.com/title/tt0368226/?ref_=nv_sr_2
    - Select another movie from IMDB and include the URL
    - Select another movie from IMDB and include the URL

In order to do this, you should create a list containing the URLs and iterate through the list to scrape the relevant information for each movie. Note that all pages have the same format. **After submitting a request to a page, you must sleep for 1 second so that you do not overburden IMDB's servers.**

Hint: The rating will need to be converted to a float (decimal) using the float function (e.g., `float("3.1")` will return the number 3.1)

Note: The title strings may contain a '\xa0', which is code for a non-breaking space. These do not have to be removed, but if you want to remove them, you can use python's *strip* method.