DOCKET ALARM

# Docket Alarm Programming Test

## Instructions

Below are programming questions to be answered using python. Some additional instructions:

- Use python 2.7
- Answer each question in a separate a python file (*e.g.*, q1.py, q2.py, ...)
- When finished, create a private github repository, push the five answers to the repository, and give access to @speedplane and @mglindblom.
- Each program should run on the command line with no arguments
- Only use built-in python libraries and (optionally) BeautifulSoup.
- Add comments to your code as if you were working with a larger team
- If you have trouble with any question, skip it and go on to the next.
- Test your program to make sure it works, if you can't get it to work, explain why

DOCKET ALARM

# Test Questions

**1. Basic Programming Test**:

a. List the libraries or framework you have used creating a python web crawler. In 200 words or less, describe the working flow.
b. Explain difference between python list and tuple.
c. Write a program that prints numbers from 1 to 1000 on each line. But for numbers that are multiples of 7 print "Docket" instead of the number. For multiples of 6 print "Alarm" instead of the number. For numbers that are multiples of both 6 and 7 print "Docket Alarm".
d. Explain what a Python generator is. Modify the answer to part "c" to utilize a generator.

**2. Basic Internet Programming**: Write a program that prints the IP address of the computer that it is being run on. If the computer is not connected to the internet, it should print "not connected"

**3. Single Scraper (one Virginia case)**

a. Write a scraper which can download and parse the page here
b. The program should output the Appellant name, Appellee name, CAV record number, and date received. The format should be in JSON and saved to a filename q3.json. The exact format of the JSON file is not important.

**4. Batch Scraper (multiple Virginia cases)**

a. Build on top of the previous answer to create a program that downloads 50 cases sequentially. Note the form of the URL, to scan through caseids 23800 to 23850.
b. Output the result into a JSON format into a file named q4.json. The JSON format should be a list, where each list item is in the same form of the JSON object you created in question 3.

**5. Form Submission (Maryland)**

a. Create a program that downloads the page located here and saves it to the file q5-1.html.
b. Then have the program click the checkbox, and select continue, download the resulting case search page and save it into a file named q5-2.html

**6. Advanced Scraper**

a. You should have received the HTML files q6-1.html, q6-2.html, and q6-3.html
b. Open each HTML file in a web browser. After scrolling down, you should see an "Actions" table, with headings "Viewed" and "Date", among several others.
c. Create a program that opens each file and creates a corresponding file called q6-1.json, q6-2.json, and q6-3.json, respectively.
d. Each JSON file should be a list of dictionaries, where each entry corresponds to the values in the table. So for example, the first entry in q6-1.html should be:

```
[{
    "Viewed" : "",
    "Date": "11/05/2004",
    "Action Text": "EXHIBITS DESTROYED",
    "Disposition": "Not Applicable",
    "Image: "",
}, ...]
```

If there are HTML links in any of the tables, add the link to the structure in a way that makes sense to you.