HW6

Reading: Chapter 11

Problems:

Use the usual naming scheme. Make sure you run the doctest on your solutions.

For this assignment we will crawl web pages, collect images, and then write the resulting images to a web page.

1. Copy the file web.py from class (or class notes) into your working folder.
2. Include the following imports at the top of your module (hopefully this is sufficient):

```
from web import LinkCollector  # make sure you did 1
from html.parser import HTMLParser
from urllib.request import urlopen
from urllib.parse import urljoin
from urllib.error import URLError
```

3. Implement a class `ImageCollector`. This will be similar to the `LinkCollector`, given a string containing the html for a web page, it collects and is able to supply the (absolute) urls of the images on that web page. They should be collected in a `set` that can be retrieved with the method `getImages` (order of images will vary). Sample usage:

```
>>> ic =
ImageCollector('http://www2.warnerbros.com/spacejam/movie/jam.
htm')
>>> ic.feed(
urlopen('http://www2.warnerbros.com/spacejam/movie/jam.htm').r
ead().decode())
>>> ic.getImages()
{'http://www2.warnerbros.com/spacejam/movie/img/p-
sitemap.gif', …,
'http://www2.warnerbros.com/spacejam/movie/img/p-
jamcentral.gif'}
```

```
>>> ic = ImageCollector('http://www.kli.org/')
>>> ic.feed( urlopen('http://www.kli.org/').read().decode())
```

```
>>> ic.getImages()
{'http://www.kli.org/wp-
content/uploads/2014/03/KLIbutton.gif',
'http://www.kli.org/wp-content/uploads/2014/03/KLIlogo.gif'}
```

4. Implement a class `ImageCrawler`. Start by copying the `Crawler` developed in class (you can use your version or the version I posted on d2l). Change the name to `ImageCrawler` and modify it so that as each page is read, the images are collected in a set. You don't have to make a lot of much modification, basically just incorporate an `ImageCollector`. The images can be retrieved using the method `getImages`.

```
>>> c = ImageCrawler()
>>>
c.crawl('http://www2.warnerbros.com/spacejam/movie/jam.htm',1,
True)
>>> c.getImages()
{'http://www2.warnerbros.com/spacejam/movie/img/p-
lunartunes.gif', …
'http://www2.warnerbros.com/spacejam/movie/cmp/pressbox/img/r-
blue.gif'}
```

```
>>> c = ImageCrawler()
>>> c.crawl('http://www.pmichaud.com/toast/',1,True)
>>> c.getImages()
{'http://www.pmichaud.com/toast/toast-6a.gif',
'http://www.pmichaud.com/toast/toast-2c.gif',
'http://www.pmichaud.com/toast/toast-4c.gif',
'http://www.pmichaud.com/toast/toast-6c.gif',
'http://www.pmichaud.com/toast/ptart-1c.gif',
'http://www.pmichaud.com/toast/toast-7b.gif',
'http://www.pmichaud.com/toast/krnbo24.gif',
'http://www.pmichaud.com/toast/toast-1b.gif',
'http://www.pmichaud.com/toast/toast-3c.gif',
'http://www.pmichaud.com/toast/toast-5c.gif',
'http://www.pmichaud.com/toast/toast-8a.gif'}
```

5. Implement a function `scrapeImages`: Given a url, a filename, a depth, and Boolean (`relativeOnly`)., this function starts at url, crawls to depth, collects images, and then writes an html document containing the images to filename. This is not hard, use the `ImageCrawler` from the prior step. For example:

```
>>> scrapeImages('http://www2.warnerbros.com/spacejam/

movie/jam.htm','jam.html',1,True)
>>> open('jam.html').read().count('img')
62

>>> scrapeImages('http://www.pmichaud.com/toast/',
'toast.html',1,True)
>>> open('toast.html').read().count('img')
11
```

After running these, the html documents are visible in a browser (the order of images may vary).