**SI 507**

**Homework 6:** Web Scraping

**Homework Objectives**

* Understand the basic structure of HTML documents
* Be able to use BeautifulSoup to extract data from web pages without an API

**Supporting Material**

* Beautiful soup documentation <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
* HTML <img> alt Attribute  
  <https://www.w3schools.com/tags/att_img_alt.asp>

**Starter Files**

We have provided you with the following files:

* [hw6\_part1.py](https://www.dropbox.com/s/bydpjndyfyvxk7v/hw6_part1.py?dl=0)
* [hw6\_part2.py](https://www.dropbox.com/s/4yjv6nzblmz02d8/hw6_part2.py?dl=0)
* [hw6\_ec1.py](https://www.dropbox.com/s/bnqu2yej887lfa0/hw6_ec1.py?dl=0)

Please use these python files as a template to add your code. You can chose to use functions or not. If you do chose to use functions, please make sure to call all functions from the main part of your program, so when we run, say, hw6\_part1.py, all outputs should print.

**Part 1 (10 points): Print some ‘alt’ tags**

There are 10 images of cats on the page <http://newmantaylor.com/gallery.html>. Some of them have “alt text,” which is the text that is displayed or spoken when the image can’t be displayed (because of browser limitations, or because someone is using a screen reader). Scrape this page and print out the alt text for each image. If there is no alt text, print “No alternative text provided!”

Your input will be a webpage url (i.e. <http://newmantaylor.com/gallery.html>) that you will pass in when you run the file.

Sample input:

$ python hw6\_part1.py http://newmantaylor.com/gallery.html

Given the current version of the page, which will remain constant until after the deadline, Your output should look like this:

\*\*\*\*\*\*\*\*\*\*\* PART 1 \*\*\*\*\*\*\*\*\*\*\*

------Alt tags------

Waving Kitty 1

No alternative text provided!

Waving Kitty 3

Waving Kitty 4

Waving Kitty 5

Waving Kitty 6

No alternative text provided!

Waving Kitty 8

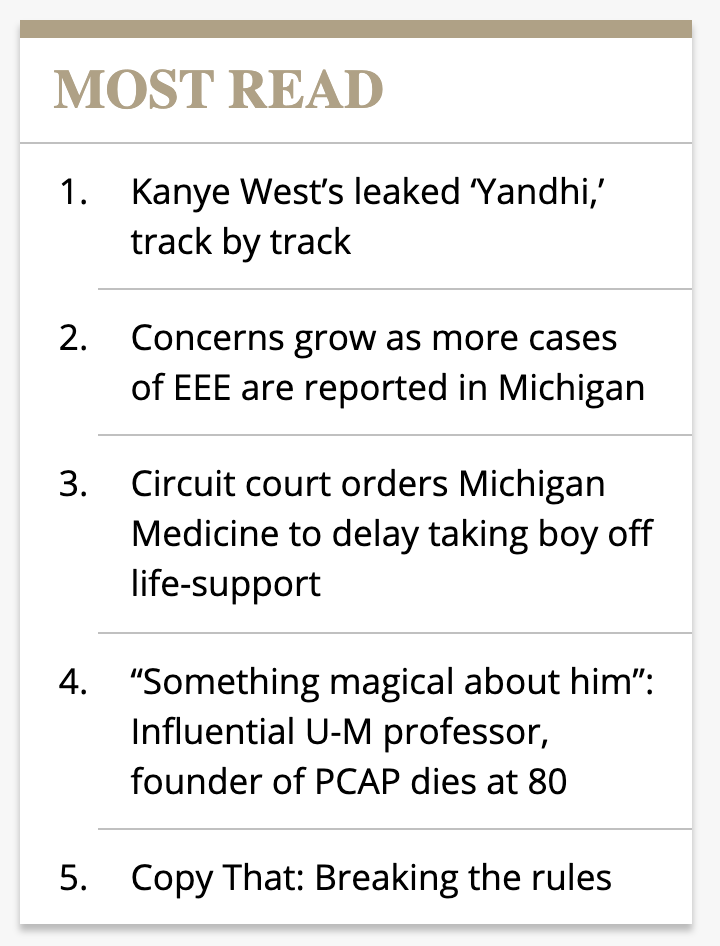
Waving Kitty 9

Waving Kitty 10

We may test your code on a different version of gallery.html or on a different website (a different url) that has different alt text. For example, it may be that the 8th image is missing alt text and the 7th images has the alt text “Waving Kitty 7.”, or completely different alt texts. So you shouldn’t hardcode the website url and you code should work for websites with different structures. (in fact, you may want to try your program on some other websites just to make sure it works.)

**Part 2 (10 points): Scrape Michigan Daily**

For this problem, you will need to inspect the Michigan Daily page (<https://www.michigandaily.com/>) to figure out how to extract the “Most Read” headlines. It’s the part of the page that looks like this (as of 12:35 pm, Oct. 8, 2019):



And it should not surprise you to learn that the output from a program that scrapes these headlines should print out (as it did at 1:05 pm, Oct. 8, 2019):

Sample input:

→ python3 hw6\_part2.py

\*\*\*\*\*\*\*\*\*\*\* PART 2 \*\*\*\*\*\*\*\*\*\*\*

Michigan Daily -- MOST READ

Kanye West’s leaked ‘Yandhi,’ track by track

Concerns grow as more cases of EEE are reported in Michigan

Circuit court orders Michigan Medicine to delay taking boy off life-support

“Something magical about him”: Influential U-M professor, founder of PCAP dies at 80

Copy That: Breaking the rules

Your code will be graded by pulling the current Most Read headings at the time of grading and comparing them to your output.

**\*\*\*Important Note\*\*\***: By default, Michigan server will refuse connections from the python request library. To get this part of the assignment to work, you will need to tell requests to identify itself as a regular browser by changing the User-Agent string it sends to the Michigan Daily web server. You do this by calling requests with the following code:

user\_agent = {'User-agent': 'Mozilla/5.0'}

html = requests.get("https://www.michigandaily.com", headers=user\_agent).text

you should now be able to read in the web page and find the data you need.

**Extra Credit 1 (2 points): Michigan Daily Top 5 for News, Sports and Arts**

Utilizing a similar approach to part 2, scrape the Michigan Daily to extract the top 5 headlines for News, Sports and Arts for that day. By top 5, we are referring to the first 5 headlines.

Your output should look like this (as of 2:10 pm, October 8, 2019):

Sample input:

$ python hw6\_ec1.py

\*\*\*\*\*\*\*\*\*\*\* EXTRA CREDIT 1 \*\*\*\*\*\*\*\*\*\*\*

Top Headlines

Top 5 Headlines: news

Dingell, state politicians address climate concerns at town hall

Supreme Court civil rights litigator talks upcoming LGBTQ rights cases

Philbert discusses tenure policy revisions, arts initiative at SACUA

Van Jones talks DEI, importance of collaboration for success

Panel discusses James Foley, the safety of American hostages

Top 5 Headlines: sports

Tien Le: Are you faster than a hockey player?

Harbaugh advocates NCAA reform but stays against California law

Michigan power play refreshed under new system

In Champaign, Michigan set to face its past in Brandon Peters

Big shoes to fill: Howard brings new energy to Michigan basketball

Top 5 Headlines: arts

We didn’t need 'Joker'

Lessons in the overuse of power with Grupo Corpo

Wilco’s ‘Ode to Joy’ lacks… well, joy

Publish Our Love: Kid Cudi

‘Almost Family’ is banal and disorganized

When grading, we will check the top 5 headlines for each sector (news, sports, and arts), and check if your program outputs the same headlines.

**What to turn in on Canvas**

* hw6\_part1.py
* hw6\_part2.py
* (optional) hw6\_ec1.py