SI206 Discussion 8

Beautiful Soup

Notes from HW 5

- Only need to use raw string (r'...') when using a special character such as \b
  - Here is a list of Python escape sequence characters https://linuxconfig.org/list-of-python-escape-sequence-characters-with-examples
- Using . * more
  - . matches any character except for line terminators (like \n)
  - * matches previous token zero to unlimited times
  - Useful if you want to match everything till a new line

Beautiful Soup for scraping

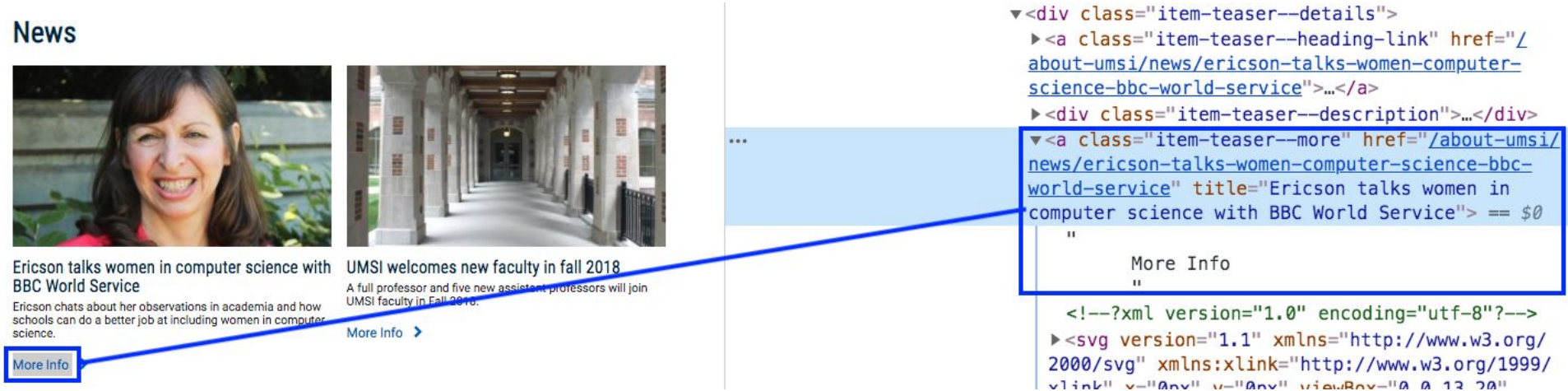
To use the Beautiful Soup module for scraping, you need to create the Beautiful Soup object. There are 3 steps to it:

1. Create a variable that stores the url of website
2. Get the data from the url i.e. `r = requests.get(url)`
3. Create a soup object using the data i.e.
   `soup = BeautifulSoup(r.text, 'html.parser')`

Things to keep in mind with BeautifulSoup

1. soup.find('tag') will return the first tag that matches
2. soup.find_all('tag') will return a list of all the tags that match
3. You can use find and find_all on the tag objects to find children tags!
4. Use the tag_object.attrs to obtain a dictionary of the attributes in a tag object
5. Use the tag_object.get(attr_name) to get a specific attribute

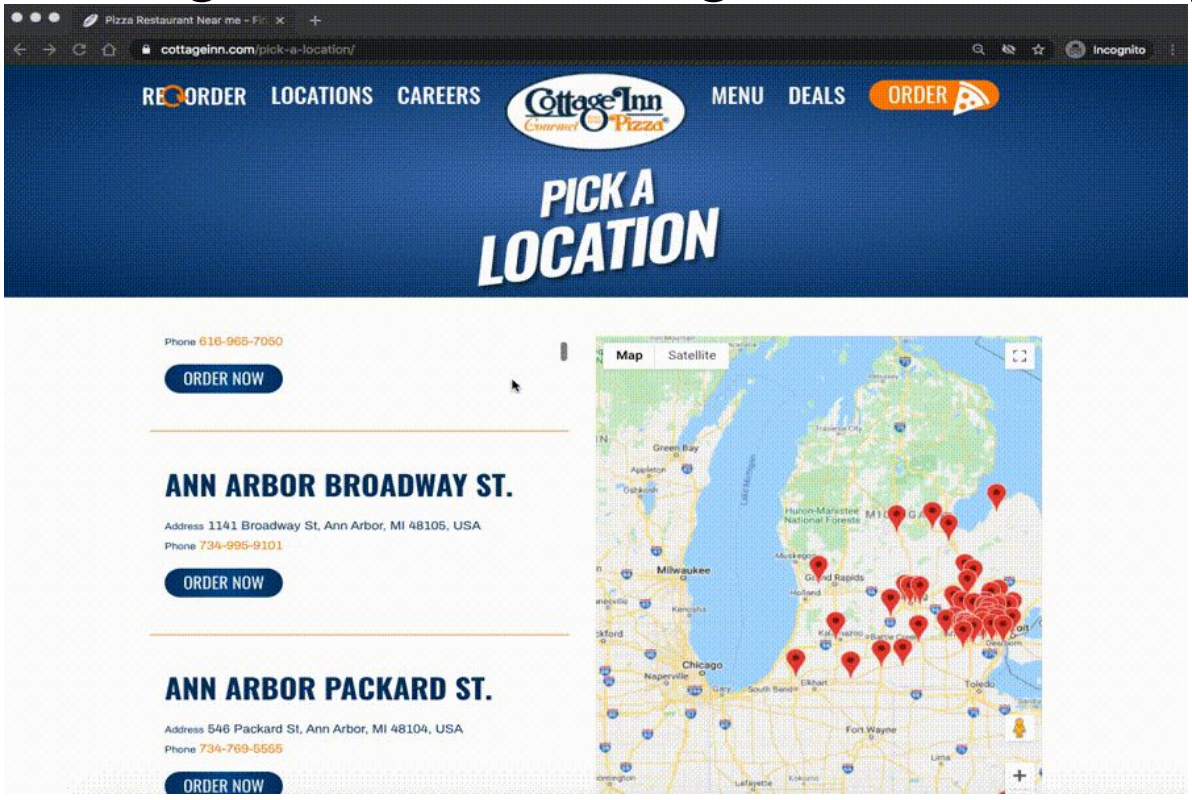
Getting info from a single tag



Since that tag is the first of its type on the page, we can use the soup.find()

```
# Get first tag of a certain type from the soup
tag = soup.find('a', class_='item-teaser--more')
# Get info from tag
info = tag.get('href')
```

Getting info from all tags of a certain type



We see that we need to get info from all the h3 tags from the webpage. The *text* in those tags has the information we need!

```
# Get all tags of a certain type from the soup
tags = soup.find_all('h3')
# Collect info from the tags
collect_info = []
for tag in tags:
    # Get info from tag
    info = tag.text
collect_info.append(info)
```

1. Find the tag description and use that as an argument in soup.find() or soup.find_all()

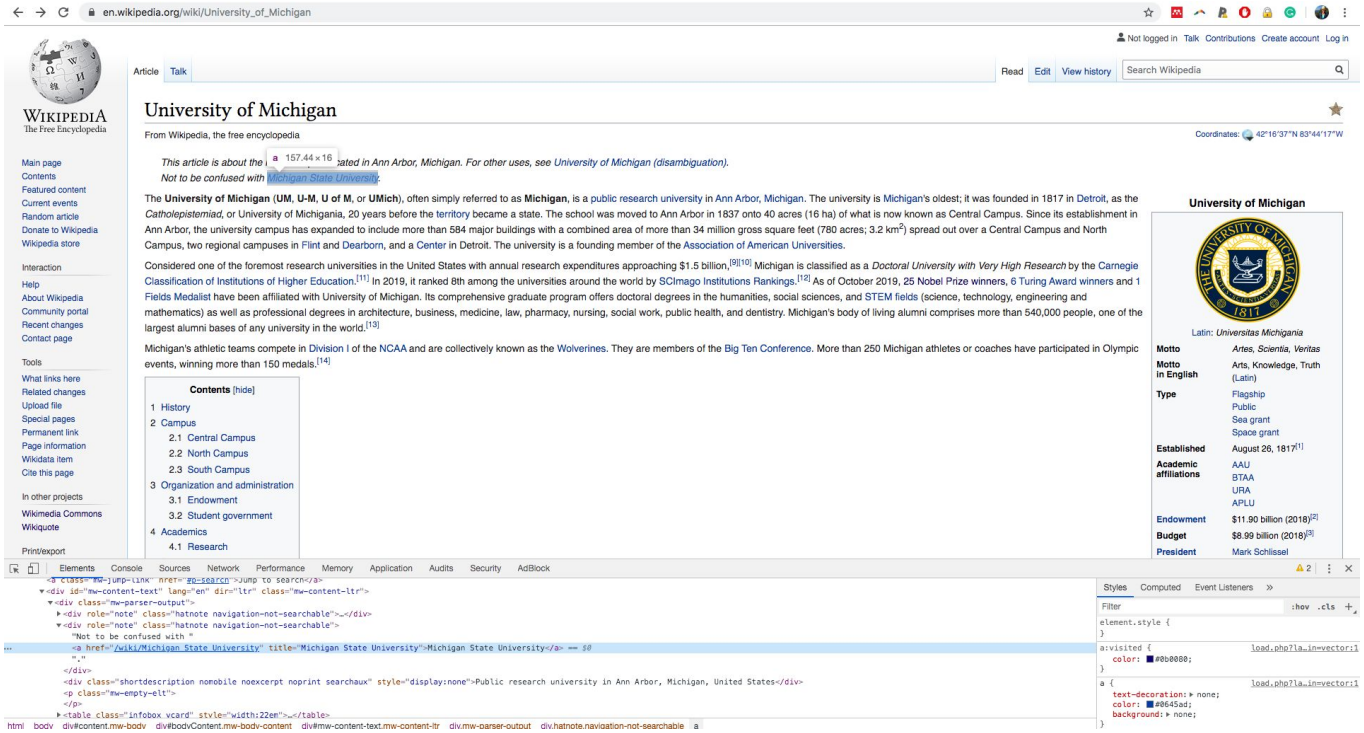
What you see when you inspect		Tag description in the code
`<p>`	->	`'p'`
`<h3>`	->	`'h3'`
`<div class="comment">`	->	`'div', class_='comment'`
`<span style="X5e72;">`	->	`'span', style='X5e72;'`
`<a class="css4z" href="/orders">`	->	`'a', class_='css4z'`

2. Determine if you want to get text from a tag, or a link from a tag

The info you want		What you put in the code
The tag's text	->	`text`
The tag's link	->	`get('href')`

Use Dev Tools!

Right click on an element you want to know more about and choose ‘Inspect’.



Scraping Wikipedia

We will use BeautifulSoup to get some data from https://en.wikipedia.org/wiki/University_of_Michigan

Task 1: Create a BeautifulSoup object

Task 2: Get the URL that links to list of American universities with Olympic medal wins. The clickable link can be found near the end of third paragraph in the introduction of the University of Michigan page.

HINT: You will have to add https://en.wikipedia.org to the URL retrieved using BeautifulSoup

Scraping Wikipedia

Task 3: Get the details from the box titled "College/school founding" in the University of Michigan Wikipedia page. Get all the college/school names and the year they were founded and organize that information into key-value pairs of a dictionary.

Organize the details into a dictionary as shown below:

```
{'College of Literature, Science, and the Arts': '1841',
 'School of Medicine': '1850',
 .
 .
 'School of Kinesiology': '1984'}
```

APPENDIX - Tips

1. We can filter tags by their attributes by passing additional arguments to the *find()* or *find_all()* methods. For instance, if I only want to get *a* tags that link to Google, I could do:
   a. `soup.find_all('a', href='https://www.google.com')`
2. Remember that you need to use *class_* instead of *class* in *find* or *find_all* because class is a reserved word in Python.
3. When trying to decide how you want to grab a particular tag, remember that in HTML a *class* is typically assigned to multiple tags while an *id* is unique.
   a. Sometimes a tag may have multiple classes separated by a space. Do not treat these all as one class.