Web Scraping

CST 205

What is web scraping?

- Extracting data from websites
    - May be against the terms of use of some websites.

Sample applications

- Price data
- Business profiles and reviews
- Job openings
- Academic research (historical weather, etc.)

Python's `urllib`

- `urllib` is Python's library for handling web page addresses
  - Part of Python's standard library
  - Was called `urllib2` in Python 2
- Split into three submodules:
  - `urllib.request`
  - `urllib.parse`
  - `urllib.error`

Aside

- When we used the Spotify API, we used the following endpoint:
  - `v1/search?q={search_string}&type=artist`
  - spaces in search_string should be encoded with hex code `%20`.
- We can do this using `urllib.parse.quote`:

```
urllib.parse.quote('Justin Bieber')

    'Justin%20Bieber'
```

result

Steps to use `urllib.request`

1. Copy a web page address from your browser address bar

2. In a Python program, execute the following:

```
from urllib.request import urlopen

# Use the web page you chose here:
my_site = "https://thespaces.com/"
html = urlopen(my_site)

# Print out a portion of the HTML
print(html.read()[100:150])
```

Uh oh, errors!

- Certain sites might *try* to block web scraping
  - `urllib.error.HTTPError: HTTP Error 403: Forbidden`
- Usually possible to get around this by sending a **user agent**

```python
from urllib.request import Request, urlopen

# Use the web page you chose here:
my_site = "https://thespaces.com/"

req = Request(my_site, headers={'User-Agent': 'Mozilla/5.0'})

html = urlopen(req)

# Print out a portion of the HTML
print(html.read()[50_000:50_100])
```

Aside: Byte literals

- urllib outputs the web page as bytes
- Byte (or bytes) literals always begin with `'b'` or `'B'`
- Can convert these to Python strings
  - `b'hello'.decode('utf-8')`
    - UTF-8 is a unicode format

Making sense of the HTML

- Regular expressions are not the answer
    - Regular expressions can only match **regular** languages
    - HTML is not a regular language
    - HTML parsing requires matching opening and closing tags.

Beautiful Soup

- https://www.crummy.com/software/BeautifulSoup
- Installation (with virtual environment activated):
  - `pip install beautifulsoup4`
  - `pip install lxml`
- How to import:
  - `from bs4 import BeautifulSoup`
- Example: GitHub Gist