Introduction to Python: Day 4 - Web Scraping

Ryden Butler

Page Source

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My first heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

Page Source

- https://polisci.wustl.edu/people/88/
- View Page Source

Scraping HTML

- `urllib2`
  - web crawler
  - navigates to url
- `BeautifulSoup`
  - parses downloaded HTML

When Is This Useful?

- Info is contained in HTML (not served by JavaScript)
- Encoded HTML follows predictable pattern
- Example:
  https://www.presidency.ucsb.edu/documents/app-categories/press/press-briefings
- Bad Example:
  https://www.oyez.org/cases/2017/17-586

Remote Driver

- We just explored HTML web crawlers
  - They are fast
  - And easy to detect / block
  - Not universal
- Selenium is a "remote driver" for your browser
  - Simulates human actions
  - Harder to track / block
  - Flexible
- Downsides:
  - Slow
  - Requires stable internet
  - Harder to set up

Scraping Tips

- Check the Terms of Service (whether you obey them or not)
- Check a site's source & structure before coding
- I use Google Chrome, which is dev-friendly
- Random time breaks help avoid detection
- Expect your code to break