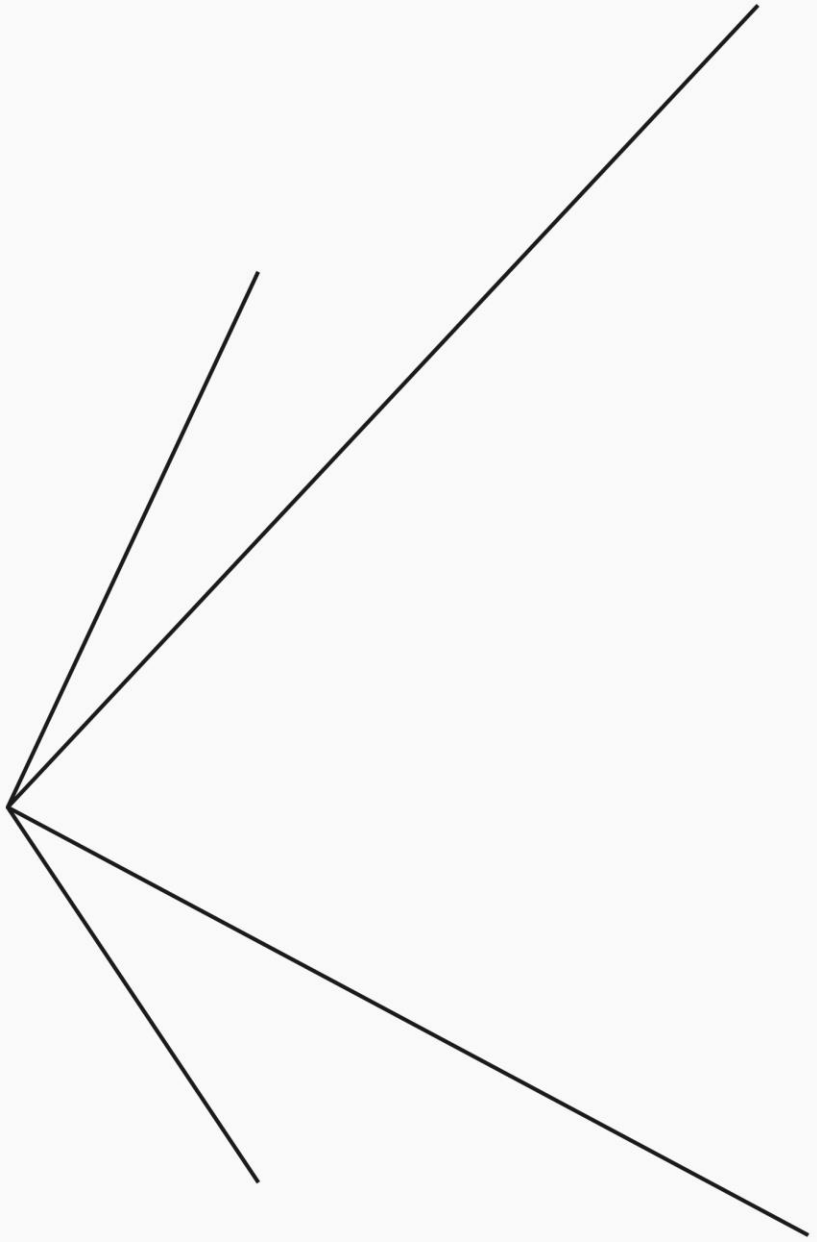
Web Scraping Training for Researchers

Louise Brown



UNIVERSITY OF BIRMINGHAM

Why Web Scraping Training?

2024 IT Needs of Active Research survey

- Web Scraping was the training topic most requested by researchers
- Survey results used to build business case to expand training programme
- Resultant funding enabled RSE time to prepare material
- First instance of course funded by general training budget

UNIVERSITY OF BIRMINGHAM

Review of Existing Carpentries Training

Carpentries incubator Introduction to Web Scraping

1. Very brief introduction to html
2. Uses XPath queries
3. Scraper Chrome extension with custom XPath queries
4. Python and Scrapy library
5. Brief section on legality of web scraping

University of California Santa Barbara Introduction to Web Scraping

- Sections 1-3 and 5 of Carpentries incubator course

UCSB Web Scraping with Python

- HTML overview and parsing using BeautifulSoup package
- Scraping a real website using the requests and BeautifulSoup packages
- Scraping dynamic websites using the Selenium package

Issues to be addressed

[Carpentries incubator Introduction to Web Scraping]

1. Very brief introduction to html
2. Uses XPath queries
3. Scraper Chrome extension with custom XPath queries
4. Python and Scrapy library
5. Brief section on legality of web scraping

[UCSB Introduction to Web Scraping]

- Sections 1-3 and 5 of Carpentries incubator course

[UCSB Web Scraping with Python]

- HTML overview and parsing using BeautifulSoup package
- Scraping a real website using the requests and BeautifulSoup packages
- Scraping dynamic websites using the Selenium package

HTML introduction too brief for researchers with no prior knowledge

XPath complicated and difficult to grasp

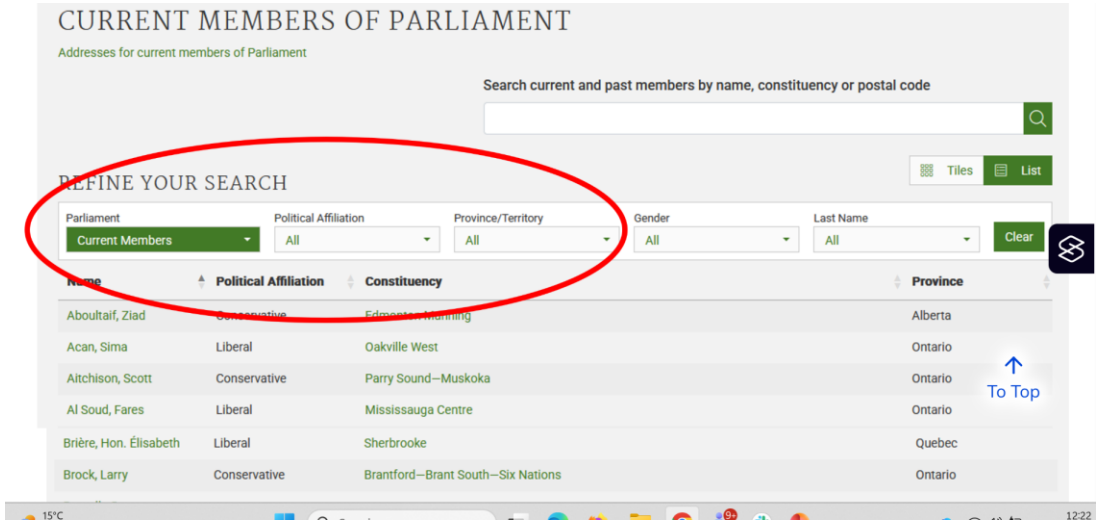
Chrome Scraper extension no longer supported

Information for USA

Too much material for half day course

UoB Introduction to Web Scraping

1. **What is Web Scraping?**
   - Brief overview
   - There may be tools provided to obtain data
2. Anatomy of a web page
3. Manually scrape data using browser extensions
4. Ethics and Legality of Web Scraping

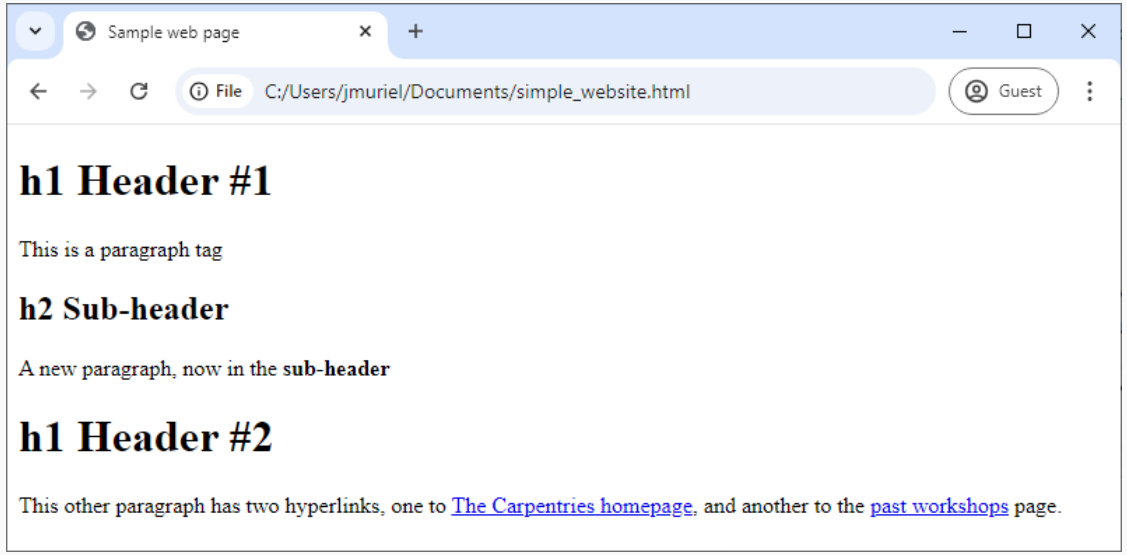



```
(...)
<tr role="row" id="mp-list-id-25446">
  <td data-sort="Allison Dean" class="sorting_1">
    <a href="/members/en/dean-allison(25446)">
      Allison, Dean
    </a>
  </td>
  <td data-sort="Conservative">Conservative</td>
  <td data-sort="Niagara West">
    <a href="/members/en/constituencies/niagara-west(1124)">Niagara West</a>
  </td>
  <td data-sort="Ontario">Ontario</td>
</tr>
(...)
```

UNIVERSITY OF BIRMINGHAM

UoB Introduction to Web Scraping

1. What is Web Scraping?
2. **Anatomy of a web page**
   - Introduction to HTML and CSS
   - Using the developer console to see the source code
3. Manually scrape data using browser extensions

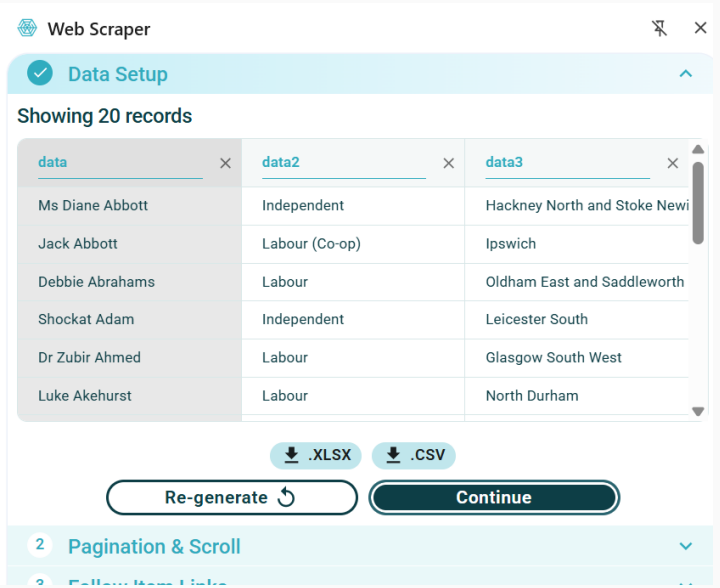


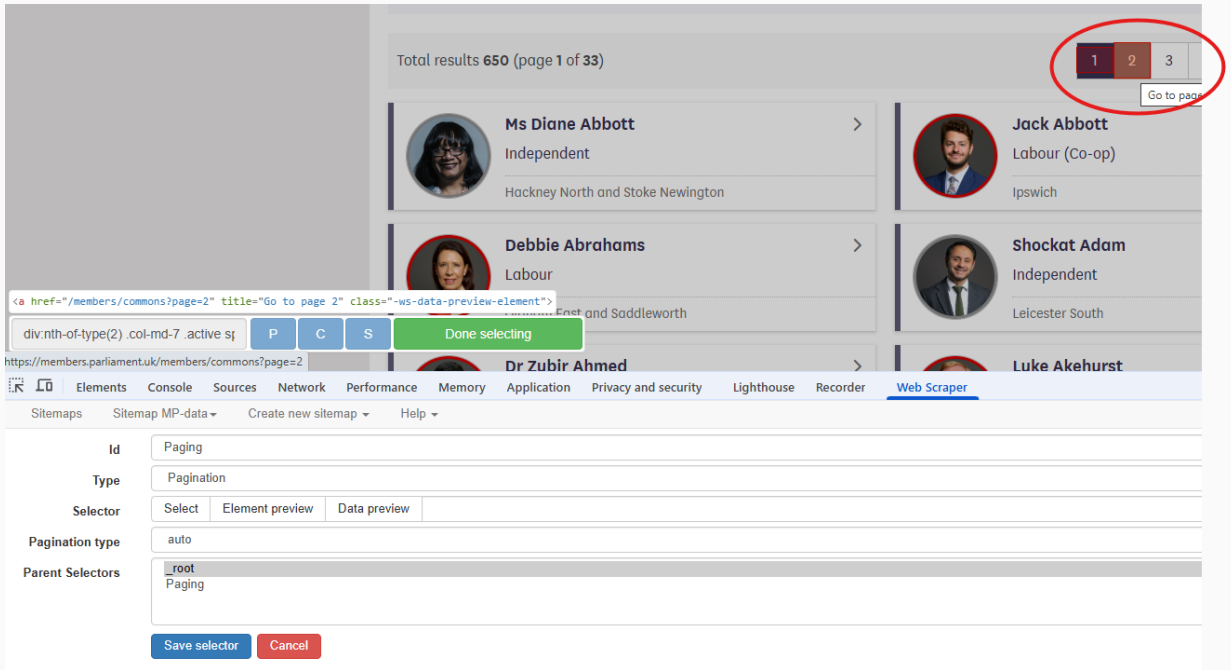

```
<!DOCTYPE html>
<html>
<head>
 <title>Sample web page</title>
</head>
<body>
 <h1>h1 Header #1</h1>
 <p>This is a paragraph tag</p>
 <h2>h2 Sub-header</h2>
 <p>A new paragraph, now in the <b>sub-header</b></p>
  <h1>h1 Header #2</h1>
  <p>
  This other paragraph has two hyperlinks,
  one to <a href="https://carpentries.org/">The Carpentries homepage</a>,
  and another to the
  <a href="https://carpentries.org/workshops/past-workshops/">past workshops</a> page.
 </p>
</body>
</html>
```

UNIVERSITY OF BIRMINGHAM

UoB Introduction to Web Scraping

1. What is Web Scraping?
2. Anatomy of a web page
3. **Manually scrape data using browser extensions**
   - Introduce the Web Scraper extension
   - Automated scraping using the wizard
   - Use with the developer console for more selected scraping
   - Use JQuery to refine selector options
4. Ethics and Legality of Web Scraping

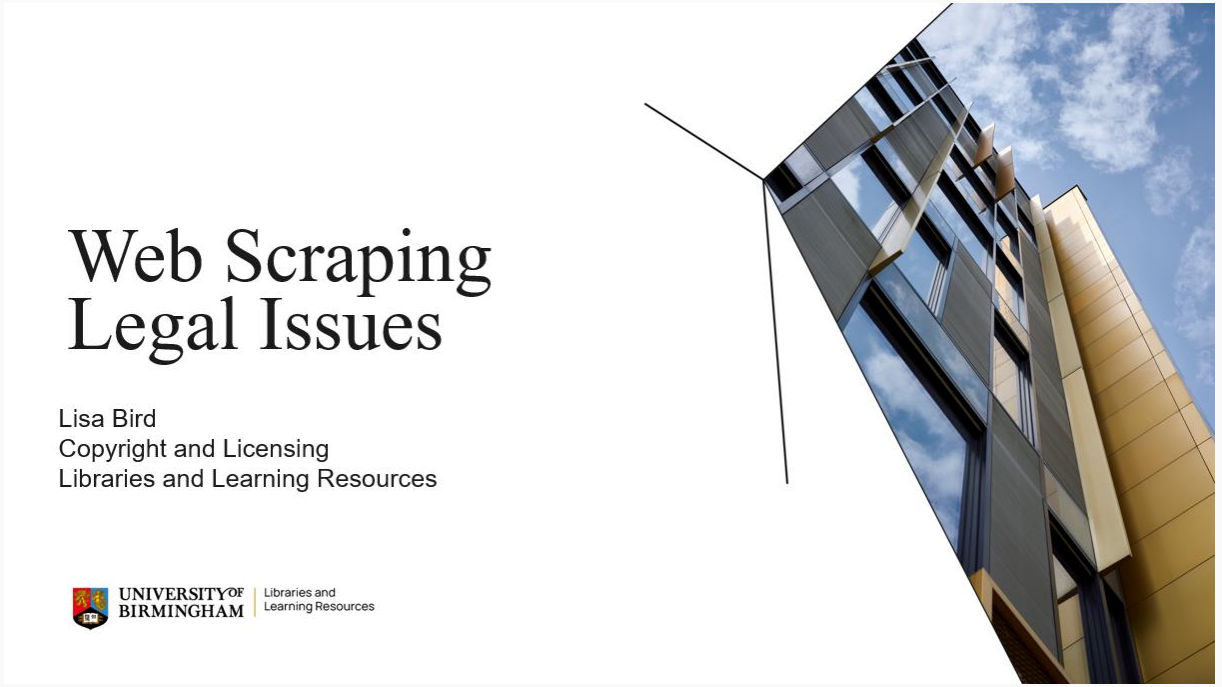




UNIVERSITY OF BIRMINGHAM

UoB Introduction to Web Scraping

1. What is Web Scraping?
2. Anatomy of a web page
3. Manually scrape data using browser extensions
4. **Ethics and Legality of Web Scraping**
   - Understand the basics of UK copyright law.
   - Understand the exception for non-commercial research.
   - Things to consider if not covered by an exception.
   - When a DPIA may be required.
   - Use of Library subscription resources.



UNIVERSITY OF BIRMINGHAM

UoB Web Scraping with Python

1. **Hello Scraping**
   - Quick HTML reminder (same text as intro course)
   - Demonstrate using BeautifulSoup on sample HTML
     - Create a ‘soup’ object
     - Use the find functions to locate specific elements
2. Scraping a real website
3. Dynamic websites

```
from bs4 import BeautifulSoup
import pandas as pd

example_html = """
<!DOCTYPE html>
<html>
<head>
<title>Sample web page</title>
</head>
<body>
<h1>h1 Header #1</h1>
…
</html>
"""
soup = BeautifulSoup(example_html, 'html.parser’)
title = soup.find('title')
headers = soup.find_all('h1')
links = soup.find_all('a')
```

UNIVERSITY OF BIRMINGHAM

UoB Web Scraping with Python

1. Hello Scraping
2. **Scraping a real website**
   - Use the requests package to obtain the HTML for a web page
   - Demonstrate using the developer console to locate the tags to extract
   - Extract data using BeautifulSoup
   - Export to csv or Excel using pandas dataframe
   - Automating page navigation using links and time delays
3. Dynamic websites

```
import requests
import re
from bs4 import BeautifulSoup

# Getting the html from our desired URL as a text string
url = 'https://carpentries.org/workshops/upcoming-workshops/'
req = requests.get(url).text

# Parsing the HTML with BeautifulSoup
soup = BeautifulSoup(cleaned_req, 'html.parser')

# Finding all third-level headers and doing a formatted print
h3_by_tag = soup.find_all('h3')
print("Number of h3 elements found: ", len(h3_by_tag))
for n, h3 in enumerate(h3_by_tag):
    print(f"Workshop #{n} - {h3.get_text()}")
```

UoB Web Scraping with Python

1. Hello Scraping
2. Scraping a real website
3. **Dynamic websites**
   - Use the Selenium package to access dynamically loaded elements in a web page
   - Use of ‘headless’ mode to run browser without opening visible window
   - Recap of scraping pipeline

```
from selenium import webdriver
from selenium.webdriver.common.by import By

# Open a Chrome web browser driven by Selenium
driver = webdriver.Chrome()

# Go to a specific website
driver.get("https://www.scrapethissite.com/pages/ajax-
javascript/")

# Find 2015 element button and click button
button_2015 = driver.find_element(by=By.ID, value="2015")
button_2015.click()

# Wait for table to load
sleep(3)

# Retrieve page HTML
html_2015 = driver.page_source

# Close web browser
driver.quit()
```

Please use/contribute

Repos for the courses are on the bham-carpentries GitHub:

Introduction to web scraping repo

Web scraping with Python repo

Note:

- The setup instructions are Birmingham specific
- The links in the legal section include Birmingham library links
  - Would advise collaborating with your library to deliver this section
- The Python course is still too long!

Please use/contribute

Repos for the courses are on the bham-carpentries GitHub:

Introduction to web scraping repo

Web scraping with Python repo

Note:

- The setup instructions are Birmingham specific
- The links in the legal section include Birmingham library links
  - Would advise collaborating with your library to deliver this section
- The Python course is still too long!

UNIVERSITY OF BIRMINGHAM

Any Questions?