WEBSCRAPING

WITH BEAUTIFULSOUP AND ETHICS

Introduction

- *Web scraping* is a term for using a program to download and process content from the web.

- *Web scraping* is the process of constructing an agent which can extract, parse, download and organize useful information from the web.

- There is no universal solution for *web scraping* because the way data is stored on each website. Scraping the data from a website requires understanding the website's structure.

Packages

- **webbrowser** Opens a browser to a specific page.
- **requests** Downloads files and web pages from the internet.
- **bs4** Parses HTML, the format that web pages are written in.
- **selenium** Is a tool designed to automate Web Browser scraping.
- **scrapy** a stand-alone ready-to-use data extracting framework. It is also highly customizable, but the learning curve is not smooth.

Beautiful Soup

- BeautifulSoup is a library that allows parsing HTML source code.
- The Request library returns the content of a url but does not automate things like string parsing and error handling.
- BeautifulSoup requires manual handling of many things Scrapy automates, making it harder to use but more configurable.

Opening BeautifulSoup

The bs4.BeautifulSoup() function needs to be called with a string containing the HTML it will parse. The bs4.BeautifulSoup() function returns is a BeautifulSoup object.

```
from urllib.request import urlopen
import requests
from bs4 import BeautifulSoup
html = urlopen('https://climate.nasa.gov/evidence/’)
soup = BeautifulSoup(html, 'html.parser’)
print(soup)
```

Select

- You can retrieve a web page element from a BeautifulSoup object by calling the select()method and passing a string of a CSS selector for the element.

Select()	Description
('div')	All elements named <div>
(‘#author’)	The element with an id attribute of author
('.notice')	All elements that use a CSS class attribute named notice
('div span')	All elements named <span> that are within an element named <div>
('div > span')	All elements named <span> that are directly within an element named <div>
('input[name]')	All elements named <input> that have a name attribute with any value
('input[type="button"]')	All elements named <input> that have anattribute named type with value button

Select Example

```
soup = BeautifulSoup(html, 'html.parser')
pElems = soup.select('p')
for i in range(0,len(pElems)):
    print(str(pElems[i]))

Prints:
            <p>This graph, based on the comparison  ... </p>
            <p><a href="https://climate.nasa.gov/ev ... </p>
            <p>The Earth's climate has changed throu ... </p>
            <p>The current warming trend is of partic  ... </p>
            <p>Earth-orbiting satellites and other tech ... </p>
            …
```

Select Example

```
scripts = soup.select('script')
for i in range(0,len(scripts)):
    print(str(scripts[i]))

Prints:
<script>
 (function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({'gtm.start':new ...
</script>
<script
type="text/javascript">window.NREUM||(NREUM={})...
</script>
```

Span Elements

The <**span**> tag is used to group inline-elements in a document. The <**span**> tag provides no visual change by itself. The <**span**> tag provides a way to add a hook to a part of a text or a part of a document.

```
pElems = soup.select('span’)
print(pElems)
for i in range(0,len(pElems)):
    print(str((spanElem)[i]))
```

Get Example

The get() method for Tag objects makes it simple to access attribute values from an element. The method is passed a string of an attribute name and returns that attribute’s value.

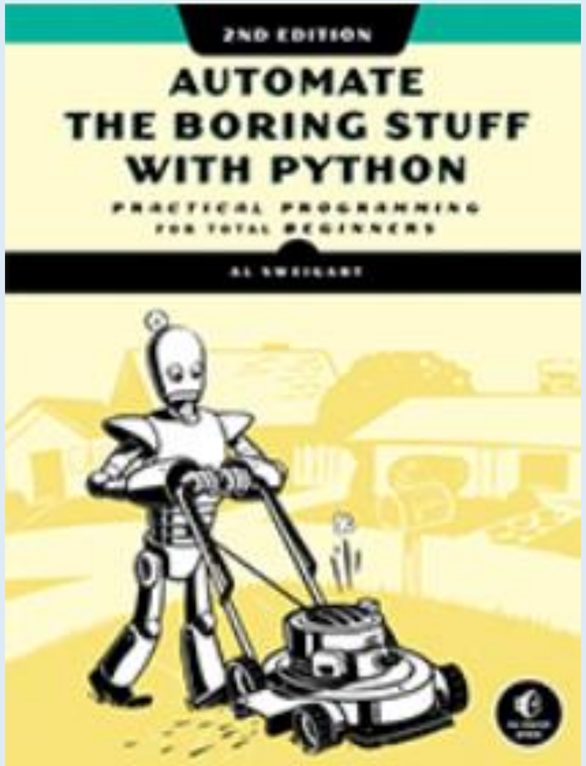
```
spanElem = soup.select('span')[0]
print(str(spanElem))
print(spanElem.get('class’))
```

Prints:

```
        <span class="menu_icon"></span>
        ['menu_icon']
```

File Scraping

- https://automatetheboringstuff.com/2e/chapter12/


2ND EDITION
AUTOMATE THE BORING STUFF WITH PYTHON
PRACTICAL PROGRAMMING FOR TOTAL BEGINNERS
AL SWEIGART

Webscraping Ethics

- Web Scraping is a technique to extract large amounts of data from websites. The data is extracted and saved to a local file.

- Data displayed by most websites is viewed using a web browser. Web browsers do not offer functionality to save a copy of this data.

- The only option then is to manually scrape the data.

Bots

- An Internet bot, is a software application that runs automated scripts over the Internet. Bots perform tasks that are simple and repetitive, at a much higher rate than a human.

- The largest use of bots is web scraping and spidering, in which an automated script fetches, analyzes and files information.

- More than half of all web traffic is made up of bots.

Scraping Scenario

By automating the process of gathering a lot of data quickly is considered stealing?

- For example, one website sells company data. The website charges a fee to export the data on a selected list of companies such as email, phones, address.
- This is information can be accessed though by simply clicking on the company's profile. This information can be easily scraped from their website.

Ethical Questions

The ability to scrape large amounts of data in a short time, brings with it ethical questions:

- Can I take this data?
- Can I republish this data?
- Am I overloading the website’s servers?
- What can I use this data for?

Terms of Service

If a website clearly states that web scraping is not allowed, you must respect that.

“You may only use or reproduce the Content for your own personal and non-commercial use. The framing, scraping, data-mining, extraction or collection of the Content of the Sites in any form and by any means whatsoever is strictly prohibited. Furthermore, you may not mirror any material contained on this site.”

Unethical Webscraping

- Can be used for Identity theft: Social media profiles and data in them can be scraped using data scraping techniques. People with malicious intentions can do this for identity theft and similar illegal acts.

- Used for Spamming: Spamming can be termed as one of the most annoying things we have ever come across on the internet.

- Encourage Plagiarism: It’s not wrong to collect content, but reproducing it anywhere without the permission from its creators is unethical.

Denial-of-Service Attacks

- A distributed denial-of-service (DDoS) attack occurs when multiple machines are operating together to attack one target.

- Attackers take advantage of security vulnerabilities or device weaknesses to control numerous devices using command and control software.

- Once in control, an attacker can command their botnet to conduct DDoS on a target. In this case, the infected devices are also victims of the attack.

Laws Concerning Web Scraping

- Computer Fraud and Abuse Act
- Copyright Infringement
- Trespass to Chattel (degrade the website)
- Crawl Rate (too much traffic downgrades website access time)
- Violating Terms of Service
- Going beyond Public Content

Three Famous Webscraping Cases

- **Craigslist vs 3Taps:** The case between Craigslist and 3Taps set a number of precedents regarding the legality of data scraping, as well as the right of businesses to deny access to publicly available data. It involved three companies: Craigslist, 3Taps, and PadMapper.

- **LinkedIn vs hiQ:** LinkedIn's dispute with hiQ Labs, a data scraping business from Silicon Valley that is not at all dissimilar to 3Taps, has echoes of the above case. The dispute is very similar in nature, revolving around whether LinkedIn can prevent the startup from accessing data that is publicly available across LinkedIn.

- **Ryanair vs PR Aviation:** This case was argued in the European Court of Justice but is the same situation as both of the above. PR Aviation enabled users to compare flight prices and was scraping Ryanair's servers for data. Unlike the US courts, the EUCJ was swift in arriving at its decision. Ryanair had argued that the scraping was a violation of the ToS, as well as a copyright infringement.

Summary

- To avoid legal problems, maintain a fine balance between the tendency to scrape under all circumstances and the respect for the website’s norms.

- If you violate any of the norms that the website has established, you are exposing yourself to legal complications.

- The webscraping laws in the U.S. permit webscraping.