CYBR 493A

Chapter 4: Web scraping in Python (Special Topic)

West Virginia University
JOHN CHAMBERS COLLEGE OF
BUSINESS AND ECONOMICS

TOPICS

- Web scraping using different modules

Introduction

- The web has a tremendous amount of data
- In many cases, especially when building software vulnerability detection and classification models, data will need to be collected from the web
- Python is equipped with great tool for web scraping

Web scraping

- All web pages are actually text files with fancy representation:
  - HTML
  - CSS
  - JS/ JSP/ JSF
  - PHP
  - XML
- Technically speaking, we can interact with web pages as text documents

Web scraping libraries

- There are several libraries/ modules Python offers to read scrap web pages:
  - Requests (we will use the highlighted modules for this class)
  - Beautiful Soup
  - LXML
  - Selenium
  - Xpath
- You are free to use whichever library you prefer

Understand web page's structure

- Each web page has a source hidden behind the scene.
- Right click on any page and click on “view page source”
- Generally speaking, each item on the page is defined by a “tag”.
  - Web links
  - Images
  - Fancy text

Understand web page's structure

- We are interested in the tag per se, we are most interested in the data that each tag holds.
- The Request library can automatically locate tags and then we can obtain the data we need and store it locally or in a Database.

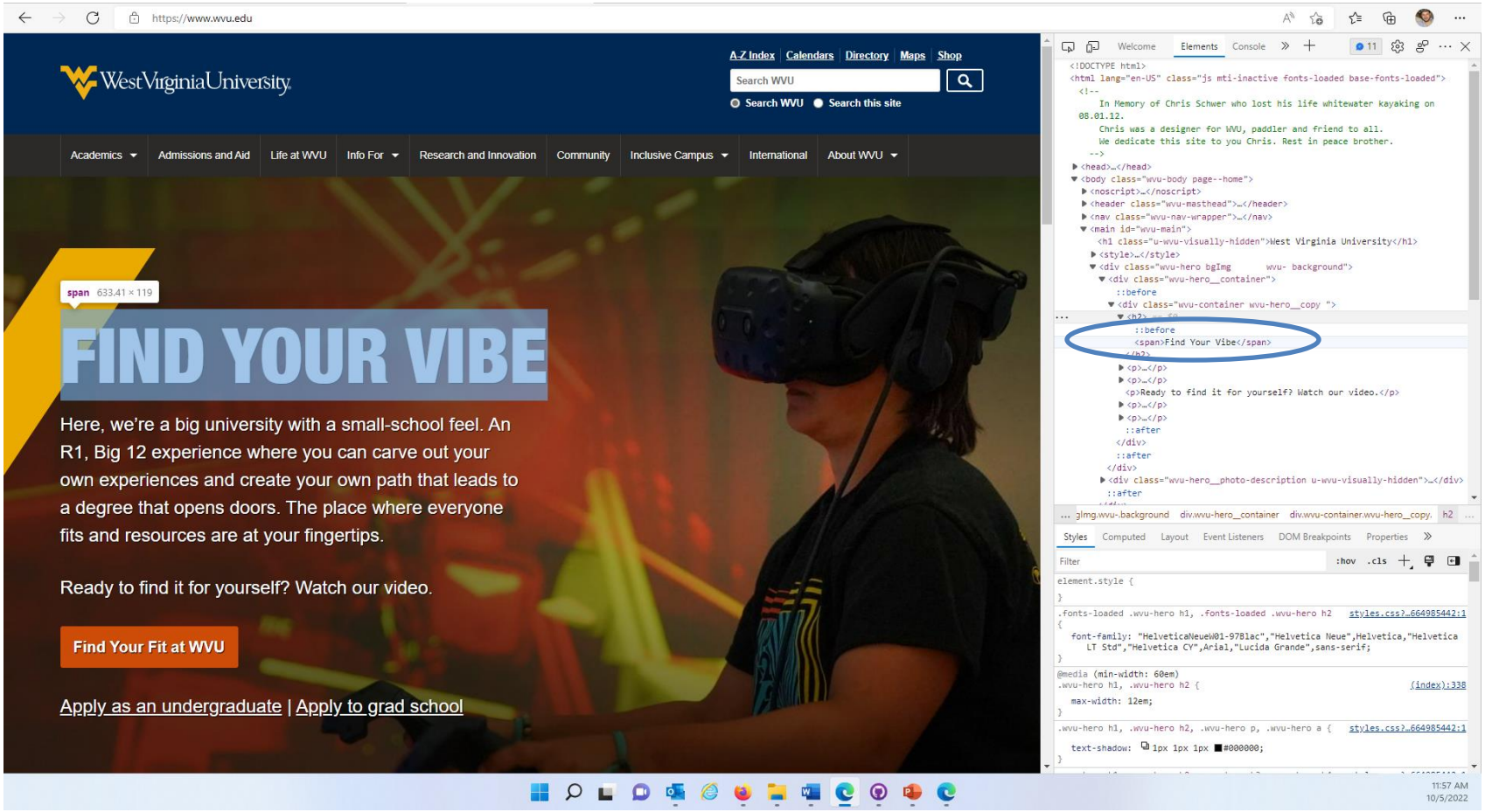
Methodology of web scraping

- In order to extract data from a web page, we need to access the tag the data belongs to.
- Each data element is usually contained within a tag.
- To view the tags, you need to view the source of each page:
  - Right click and click on View Source
  - An easier way is to click on Inspect (preferably Chrome browser)

Methodology of web scraping

- The Inspect feature allows you to view the source code of the page and the data it holds.
- You will need to obtain the tag by tracing its tree.
- See the next slide

Methodology of web scraping



- If we want to extract the "FIND YOUR VIBE" text, then we need to obtain its tag path which is:

//*[@id="wvu-main"]/div[1]/div[1]/div/h2/span and since we need the text, we need to add /text() after span, so the location of that text would be: //*[@id="wvu-main"]/div[1]/div[1]/div/h2/span /text()

- You can copy the Xpath of the full path from the source code: Right Click -> Copy Xpath.

Syntax

- First, you need to obtain the web page Tree, see the Web_Scraping.py file on our Repo.

```python
Web scrapping in Python
This is an example of scrapping a web page in Python using requests and lxml

"""
import requests
from lxml import html


def get_web_tree(link):
    """
    This method gets a web page from the specified url, and returns a tree of all elements in the page
    :param link: The webpage to access and process
    :return: The tree element created from the page
    """
    # Welcome message
    print('Obtaining the page: ', str(link))
    # get the page
    page = requests.get(link)
    # get the elements from the page
    page_tree = html.fromstring(page.content)
    # return the tree of the web page
    return page_tree
```

Syntax

- Second, include the path to the data you need to obtain. Please refer to the Web_Scraping_Main.py file on our Repo

```
print(str(len(first_div)))
find_your_vibe = main_tree.xpath('//*[@id="wvu-main"]/div[1]/div[1]/div/h2/span/text()')
print(find_your_vibe)
```

- There you go ☺

Documentation

- Please look at the documentation of the lxml library for more info.
- Also : https://www.guru99.com/selenium-webtable.html

Scraping Software Vulnerabilities

- We have been using github to share and submit our codes.
- In big project, there is another system that keeps track of software bugs:
  - Bug Tracking Systems:
    - JIRA
    - Bugzilla

Scraping Software Vulnerabilities

- Bug Tracking Systems allow users/ developers to submit bug reports to report issues with these software.
- We will look at Bugzilla and collected some bug reports
  - Red Hat Bugzilla Main Page
  - Specialized search for certain bugs

Link needed for later

- https://bugzilla.redhat.com/buglist.cgi?bug_status=__closed__&bug_status=CLOSED&classification=Red%20Hat&classification=Fedora&f1=short_desc&f2=alias&f3=longdesc&j_top=OR&limit=0&o1=anywordssubstr&o2=anywordssubstr&o3=anywordssubstr&order=priority%2Cbug_severity&product=Fedora&product=Red%20Hat%20Enterprise%20Linux%202.1&product=Red%20Hat%20Enterprise%20Linux%203&product=Red%20Hat%20Enterprise%20Linux%204&product=Red%20Hat%20Enterprise%20Linux%205&product=Red%20Hat%20Enterprise%20Linux%206&product=Red%20Hat%20Enterprise%20Linux%207&product=Red%20Hat%20Enterprise%20Linux%208&query_format=advanced&resolution=WONTFIX&resolution=DEFERRED&resolution=CURRENTRELEASE&resolution=RAWHIDE&resolution=ERRATA&resolution=UPSTREAM&resolution=NEXTRELEASE&short_desc=CVE-&short_desc_type=anywordssubstr&v1=CVE-&v2=CVE-&v3=CVE-

Launchpad: Ubuntu Bug Tracking System

- Ubuntu has its own but tracking system, Launchpad:
  - Tracks bug
  - Cybersecurity Vulnerabilities

Launchpad pre-defined search

- Visit this link, and familiarize yourself with the structure of the page
- https://bugs.launchpad.net/ubuntu/+bugs?field.searchtext=&field.status%3Alist=EXPIRED&field.status%3Alist=CONFIRMED&field.status%3Alist=TRIAGED&field.status%3Alist=INPROGRESS&field.status%3Alist=FIXCOMMITTED&field.status%3Alist=FIXRELEASED&field.importance%3Alist=UNKNOWN&field.importance%3Alist=UNDECIDED&field.importance%3Alist=CRITICAL&field.importance%3Alist=HIGH&field.importance%3Alist=MEDIUM&field.importance%3Alist=LOW&field.importance%3Alist=WISHLIST&field.information_type%3Alist=PUBLIC&field.information_type%3Alist=PUBLICSECURITY&field.information_type%3Alist=PRIVATESECURITY&field.information_type%3Alist=USERDATA&assignee_option=any&field.assignee=&field.bug_reporter=&field.bug_commenter=&field.subscriber=&field.structural_subscriber=&field.component-empty-marker=1&field.tag=&field.tags_combinator=ANY&field.status_upstream-empty-marker=1&field.has_cve.used=&field.omit_dupes.used=&field.omit_dupes=on&field.affects_me.used=&field.has_no_package.used=&field.has_patch.used=&field.has_branches.used=&field.has_branches=on&field.has_no_branches.used=&field.has_no_branches=on&field.has_blueprints.used=&field.has_blueprints=on&field.has_no_blueprints.used=&field.has_no_blueprints=on&search=Search&orderby=-importance&memo=75&start=0

Labeling vulnerable bugs in Ubuntu

- Each bug has a bug report on Launchpad
- As a viewer, how would you know whether a bug report relates to a security-related bug (i.e., vulnerability)?
  - Examine some bugs provided in the main link of bugs
  - Examine the bug label