WebScrapingPython

Web Scraping with PYTHON

View online: https://git.yale.edu/pages/yul-dss/WebScrapingPython/

Download workshop materials: https://yale.box.com/v/WebScrapingPython

OUTLINE:

1. What is web-scraping? What can be scraped?
2. Tools for Scraping
3. Making a GET Request
4. Scraping an HTML Table
5. Scraping links on a page
6. Downloading Files
7. Scrap all text
8. Scraping Paginated Results
9. Resources

What is Web Scraping?

- Fetching and extracting data from websites using software or bots.
- Useful when there is no API for fetching data
- No direct 'Download' of data
- Data trapped on older websites

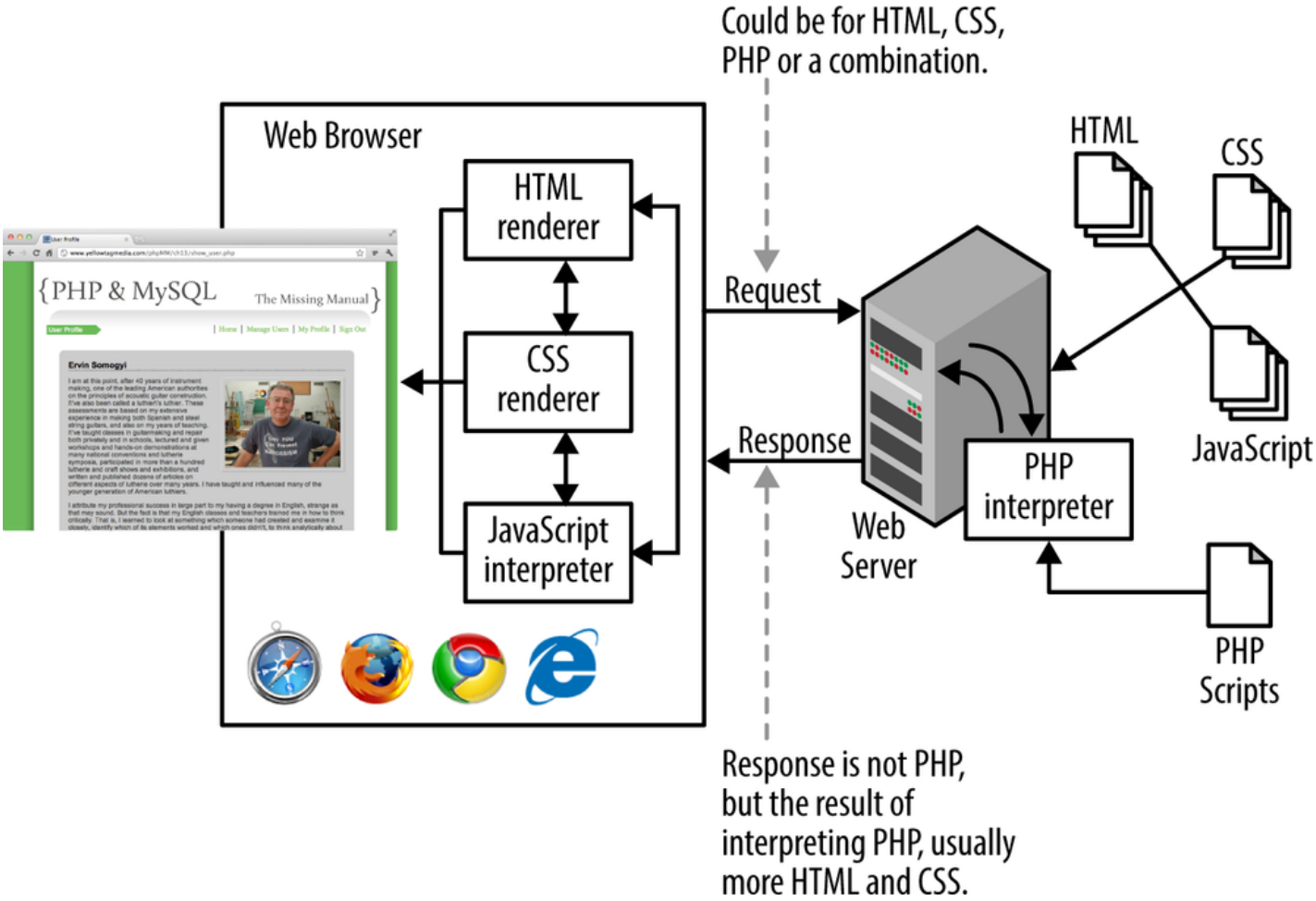
What sites can be scraped?

- Any website can be scraped
- Basic:
    i. Simple HTML pages
    ii. Textual data, HTML tables, hyperlinks
- Advanced:

i. Javascript/AJAX
ii. Password protected
iii. Unstructured Data
iv. Interactive/visualizations

What are we scraping?

- The HTML behind every web page
- [right-click] “View Source”
- Will not interperet content like a browser



Copyright & Fair Use

- Check the site for license

> As per Clause 8, dealing with Restrictions on Content and Use of the Services, of the Twitter Terms of Service, as part of your free access, you are expressly prohibited to access, tamper with, or use non-public areas of the Twitter services; or probe, scan, or test the vulnerability of any system or network or breach or circumvent any security or

> authentication measures; or access or search or attempt to access or search the Services by any means (automated or otherwise) like scraping without the prior consent of Twitter.

- Ask the site owner
- Check withthe Licensing & Copyright Librarian (Joan Emmet)
- Electronic Resources @ Yale:

Appropriate use of licensed resources:

Most of the library's electronic resources are governed by license agreements that limit use to the Yale community or to individuals who are physically present at Yale University Library facilities.

- **Each user is responsible for ensuring that he or she uses these products solely for noncommercial, educational, scholarly or research use. Systematic downloading, distribution of content to non-authorized users or indefinite retention of substantial portions of information *is strictly prohibited***
- The use of software such as scripts, agents, or robots, is generally prohibited and may result in loss of access to these resources for the entire Yale community.

See Resources for Text & Data Mining

Tools for Web Scraping

- Python:
  - BeautifulSoup –> Modifying, Parsing, and Searching HTML or XML
  - Selenium –> Testing websites; useful for scraping sites with lots of js, interactivity, or log-ins
- Scrapinghub –> create spiders to crawl the web; Scrapy Cloud runs in browser & includes GUI
- OctoParser –> Point & Click web scraping
- OpenRefine
- R
- Excel –> NodeXL

Making a GET Request

- We access the HTML of any website by making a GET request for that site.
- Generally you can copy & paste the full URL from your browser directly into your python code.
- First we save the results of our GET request to the variable r, before printing those results to the screen.

```
import requests

url = "http://web.library.yale.edu"
r = requests.get(url)
print(r)
```

```
<Response [200]>
```

HTML Status Codes

A `<Respose [200]>` means we succesfully connected to the webpage. Read more about HTML status codes here: Status Codes

GETting the HTML

In the last section we got a succesful response from our GET request. Now let's get the actual HTML text for the webpage.

```
import requests

url = "http://web.library.yale.edu"
r = requests.get(url)
print(r.text)
```

- We use `r.text` to access the 'text' or HTML behind the webpage.
- We've successfully scraped our first webpage

Scraping an HTML Table

- We'll be scraping the Yale University Library homepage for a list of libraries and their hours of operation.
- If you open the library site, you can see a small table with the names of library locations and their hours for today.

RESEARCH

- Search Library Catalog (Orbis)
- Search Law Library Catalog (MORRIS)
- Search Articles+
- Search Digital Collections
- Search Finding Aids Database
- Search Borrow Direct
- Search WorldCat
- Research Guides
- Find Databases by Title
- Find eJournals by Title
- Guide to Using Special Collections

SERVICES

- Your Personal Librarian
- Subject Specialists
- Research Support and Workshops
- Citation Tools
- Get It @Yale (Borrow Direct, Interlibrary Loan, Scan & Deliver)
- Course Reserves
- Off-Campus Access
- Digital Services
- EliScholar
- OverDrive: Popular Audio and eBooks
- Bass Media Equipment

TODAY'S HOURS

Bass	8:30am - 9:45pm
Beinecke	9:00am - 5:00pm
CSSSI	8:30am - 7:00pm
Divinity	8:30am - 4:50pm
Haas Arts	8:30am - 5:00pm
Law	8:00am - 10:00pm
Lewis Walpole	8:30am - 4:45pm
Manuscripts & Archives	8:45am - 4:45pm
Medical	7:30am - 10:00pm
Music	8:30am - 4:45pm
Sterling	8:30am - 4:45pm

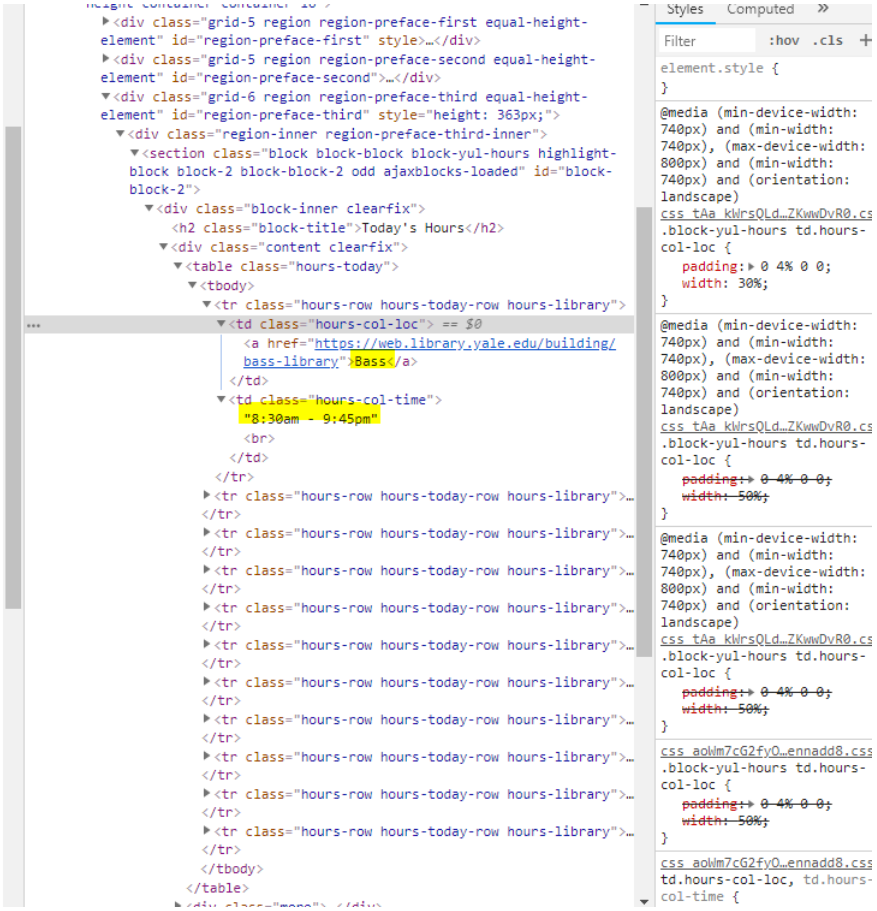
All Library Hours

'Inspect'

TODAY'S HOURS

Bass	8:30am - 9:45pm
Beinecke	9:00am - 5:00pm
CSSSI	8:30am - 7:00pm
Divinity	8:30am - 4:50pm
Haas Arts	8:30am - 5:00pm
Law	8:00am - 10:00pm
Lewis Walpole	8:30am - 4:45pm
Manuscripts & Archives	8:45am - 4:45pm
Medical	7:30am - 10:00pm
Music	8:30am - 4:45pm
Sterling	8:30am - 4:45pm

All Library Hours



- When inspecting the HTML code behind the *table* element, we see that each library location name is stored in a *td* element with the *class="hours-col-loc"*
- The hours of operation are stored in the *td* element directly following the location
- Now that we've identified where the data we want to scrape lives on the page, we can begin to write our code.

Importing Modules

- The first step in any python code is identifying which libraries we are using and importing those. For this section, we will be using three python libraries: csv, BeautifulSoup, requests.

```
import requests, csv
from bs4 import BeautifulSoup
```

Making the soup

- The python library BeautifulSoup allows us to parse our website's HTML before filtering or searching through the results.
- Our code:
  - Loads the variable r into BeautifulSoup
  - Identifies 'html.parser' as the method for parsing the site's HTML.
  - When we print the variable soup, we see the site's raw HTML, the same HTML that we see when we 'View Source' through the browser.

```
url = "http://web.library.yale.edu"
r = requests.get(url)
# print(r.text)

soup = BeautifulSoup(r.text, 'html.parser')
print(soup)
# print(soup.prettify())
```

```
<!DOCTYPE html>

<!--[if lt IE 7 ]> <html lang="en" dir="ltr" class="ie6"> <![endif]-->
<!--[if IE 7 ]>    <html lang="en" dir="ltr" class="ie7"> <![endif]-->
<!--[if IE 8 ]>    <html lang="en" dir="ltr" class="ie8"> <![endif]-->
<!--[if IE 9 ]>    <html lang="en" dir="ltr" class="ie9"> <![endif]-->
<!--[if (gt IE 9)|!(IE)]><!--> <html dir="ltr" lang="en"> <!--<![endif]-->
<head>
<meta content="IE=edge" http-equiv="X-UA-Compatible">
<!--
```

```
      GGGGGGGGGGGG      GGGGGGGGGGG               fGGGGGG
        ;GGGGG.             GGGi                   GGGG
          CGGGG:           GGG                     GGGG
           lGGGGt         GGL                      GGGG
            .GGGGC       GG:                       GGGG
              GGGGG    .GG.        ;CGGGGGGL       GGGG          .LGGGGGGGL
               GGGGG  iGG        GGG:   ,GGGG      GGGG        tGGf     ;GGGC
                LGGGGfGG        GGGG     CGGG;     GGGG       GGGL       GGGGt
                 lGGGGL                  CGGG;     GGGG      CGGGCCCCCCCCCCCCC
                  GGGG                   GGGG,     GGGG      GGGG
                  GGGG              tCGG; CGGG,     GGGG      GGGG
                  GGGG           GGGG     CGGG.     GGGG      GGGGL
                  GGGG          GGGGC     CGGG.     GGGG      :GGGGC          :
                 ,GGGGG.       GGGGG:   .LGGGGG,.tG   GGGG       GGGGGGt,..,fGC
               ,GGGGGGGGGGGGGGf    iGGGGGG   CGGGGC  GGGGGGGGGG       LGGGGGGGC


    -->
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
  <link href="http://web.library.yale.edu/sites/all/themes/yale_omega_base/images/favicon.i
  <meta content="width=device-width, initial-scale=1, maximum-scale=10, minimum-scale=1, us
  <meta content="Get to know the helpful people, explore the unique collections, and see th
  <meta content="Drupal 7 (http://drupal.org)" name="generator"/>
  <link href="http://web.library.yale.edu/" rel="canonical"/>
  <link href="http://web.library.yale.edu/" rel="shortlink"/>
  <title>Yale University Library | The Heart of the University</title>
  <meta content="true" name="HandheldFriendly">
  <meta content="320" name="MobileOptimized">
  <!-- Apple Touch Homescreen images -->
...
  </body>
  </html>
```

Straining the Soup

- We have the HTML for the entire page, so we must filter these results to reach just the elements we want to scrape.
- As we recall, the info we wish to scrape lives in a *td* element with *class="hours-col-loc"*
- Use the `find_all` function built in to BeautifulSoup

```
tableData = soup.find_all('td', class_="hours-col-loc")
print(tableData)
```

```
[<td class="hours-col-loc">Bass</td>, <td class="hours-col-loc">Beinecke</td>, <td class="hou
```

- Now we have a list of HTML elements, but we want the specific text inside the HTML.
- Before we can begin pulling out this text, we need to create a loop in our code. This loop does three things:
  - First it selects a *td* element in our tableData list and saves it as the variable td
  - Then the .text function pulls out only the string of text inside the *td* element (library location) and stores the value in a variable called libraryName
  - Lastly it prints the result.
- Our loop repeats the process again for the next *td* element in our list. We should see the name of each library printed to the screen in its own line.

```
for td in tableData:
    libraryName = td.text
    print(libraryName)
```

```
Bass
Beinecke
CSSSI
Divinity
Haas Arts
Law
Lewis Walpole
Manuscripts & Archives
Medical
Music
Sterling
```

- Next we need to get the hours that each library is open.
- This data lives in an adjacent *td* element which is referred to as a sibling.
- BeautifulSoup allows us to move over to the neighboring element with a function called `.next_sibling`.
  - We create a variable called hours
  - Then we set it equal to `td.next_sibling.text`. Our code will then print this variable.
- If we run the code now, we should see a list alternating between a library's name and the hours of operation.

```
for td in tableData:
    libraryName = td.text
    hours = td.next_sibling.text
    print(libraryName,hours)
```

```
Bass 8:30am - 9:45pm
Beinecke 9:00am - 5:00pm
CSSSI 8:30am - 7:00pm
Divinity 8:30am - 4:50pm
Haas Arts 8:30am - 5:00pm
Law 8:00am - 10:00pm
Lewis Walpole 8:30am - 4:45pm
Manuscripts & Archives 8:45am - 4:45pm
Medical 7:30am - 10:00pm
Music 8:30am - 4:45pm
Sterling 8:30am - 4:45pm
```

Exporting the Data

- Now that we have filtered our results and reached the data we want to scrape from the site, we need to export our results so they can be analyzed.
- We can save our results to a CSV file, which can be opened in excel or any text editor.
  - First, we open a csv document with our python script. The open command will open the file within the first set of quotes. If that file doesn't exist, python will create a file with that name.
  - Next, we set the python function `csv.writer` to a variable `w`. This allows us to begin writing data to the CSV file. (The csv file will 'close' or 'save' at the end of the loop)
  - Then, we must decide what to save to the CSV file

```
with open('yaleLibraryHours.csv', 'w', newline='') as ourCSVdata:
    w = csv.writer(ourCSVdata)
    for td in tableData:
        libraryName = td.text
        hours = td.next_sibling.text
        #print(libraryName,hours)
```

- Before we can write our data to the CSV, we should create headers for our columns:
  - We save the value of our headers as a list. The values in this list represent one row of our final CSV file.
  - The function writerow will write the list called headers to our CSV file.
  - After running the code, we will notice a new file called "yaleLibraryHours.csv" is now in the same folder as our python code. We can open the file and we will see our headers.

```
with open('yaleLibraryHours.csv', 'w', newline='') as ourCSVdata:
    w = csv.writer(ourCSVdata)
    # write header for the csv
    w.writerow(['Library', 'Hours'])
```

```
    for td in tableData:
        libraryName = td.text
        hours = td.next_sibling.text
        #print(libraryName, hours)
```

- We can now add our scraped data to our CSV file. To do this we must write a row of data during each iteration of the for loop.
  - First, we create an empty list called row.
  - Then add our data to the list using row.append
  - Finally, writing the list to our CSV file using the function `w.writerow(row)`.

```
with open('yaleLibraryHours.csv', 'w', newline='') as ourCSVdata:
    w = csv.writer(ourCSVdata)

    # write header for the csv
    w.writerow(['Library', 'Hours'])

    for td in tableData:
        libraryName = td.text
        hours = td.next_sibling.text
        #print(libraryName, hours)
        w.writerow([libraryName,hours])
```

Scraping links from a page

- We want to scrape the links to all the cases from Justia
- Make sure your modules are imported
- Begins the same for most web scraping attempts

```
import requests, csv
from bs4 import BeautifulSoup

url = "https://law.justia.com/cases/federal/district-courts/connecticut/ctdce/2018/"

r = requests.get (url)
# print(r)
# print(r.text)

soup = BeautifulSoup(r.text, 'html.parser')
# print(soup)
```

Find all links & Print

- Using the `find_all` function built in to BeautifulSoup
- The info we want (including links) are each in a *a* element with *class='case-name'*

```
cases = soup.find_all('a', class_='case-name')
print(cases)
```

- Open a CSV file to write data
- Define headers
- Loop through all the divs
- Create a `row` that includes the caseName and link
- Write each `row` to our csv file

```
with open('caseLawLinks.csv', 'w', newline='') as outFile:
    w = csv.writer(outFile)
    for case in cases:
        caseName = case.strong.span.text
        #print(caseName)
        link = 'https://law.justia.com' + case['href']
        #print(link)
        w.writerow([caseName,link])
```

Downloading Files from webpages

- Let's take our list of links found in 'CaseLawLinks.csv' and download the pdf on each page
- Our codes starts by: importing libraries, opening our csv file, and making a get request

```
import csv, requests
from bs4 import BeautifulSoup

with open('CaseLawLinks.csv', 'r') as inFile:
    read = csv.reader(inFile)
    for row in read:
        #print(row)
        url = row[1]
        caseName = row[0]
        #print(url, caseName)
        r = requests.get(url)
        #print(r.text)
```

- We are looping through each row of our csv and saving the values for the url and case name.
- Now we need to make our soup and find the links to the PDF on each page

- The 'PDF Download' link is found in the *a* element with the *class='pdf-icon'*

```
        soup = BeautifulSoup(r.text, 'html.parser')
        #print(soup)
        pdf = soup.find('a', class_='pdf-icon')
        #print(pdf)
        pdfLink = 'https:' + pdf['href']
        #print(pdfLink)
```

Saving the PDF files

- Now that we have the url for the PDF download, we can make another GET request for each PDF
- We first need to use `with open()` to create a filename for each PDF. Since we have the case name save as the variable `caseName`, we can use that for our filename adding `'.pdf'` to the end.
- We use `'wb'` to mean 'write binary'. This write the data to a file and is used when working with non-text data and files

```
        with open(caseName + '.pdf', 'wb') as outFile:
            r2 = requests.get(pdfLink)
            outFile.write(r2.content)
```

- We should end up with one PDF for each link in the same folder as our code

Scraping all text

- Simply scrapes all text within a single website using the built-in `get_text()` function of BeautifulSoup.

```
import requests
from bs4 import BeautifulSoup

url = "https://en.wikipedia.org/wiki/Main_Page"
r = requests.get(url)
print(r)

soup = BeautifulSoup(r.text, 'html.parser')

txt = soup.get_text()
print(txt)
```

Scraping Paginated Results

- Data spread across multiple pages on a single site
- URLs that follow a pattern through each search page are the easiest to scrape

```
r = requests.get("http://orbis.library.yale.edu/vwebv/search?searchArg=python&searchCode=GKEY
print(r)
```

```
<Response [200]>
```

```
r = requests.get("http://orbis.library.yale.edu/vwebv/search?searchArg=raspberry&searchCode=G
#print(r)

soup =BeautifulSoup(r.text, 'html.parser')
#print(soup)

searchResults = soup.find_all('div', class_='resultListTextCell')
#print(searchResults)
for item in searchResults:
    info = [item.a.text]
    print(info)
```

```
['Raspberry Island Light Station : Apostle Islands National Lakeshore, Bayfield, Wisconsin /
['Building a home security system with Raspberry Pi : build your own sophisticated modular ho
['Exploring the Raspberry Pi 2 with C++ / Warren Gay.']
['Learn to program with Minecraft : transform your world with the power of Python / by Craig
['Raspberry Pi robotic blueprints : utilize the powerful ingredients of Raspberry Pi to bring
['Python playground : geeky projects for the curious programmer / Mahesh Venkitachalam.']
["Maker's guide to the zombie apocalypse : defend your base with simple circuits, Arduino, ar
['Raspberry Pi projects for dummies / Mike Cook, Jonathan Evans, Brock Craft.']
['Raspberry Pi LED blueprints : design, build, and test LED-based projects using Raspberry Pi
['Learn Raspberry Pi 2 with Linux and Windows 10 / Peter Membrey, David Hows.']
["Getting started with Python and Raspberry Pi : learn to design and implement reliable Pytho
['Raspberry Pi Android projects : create exciting projects by connecting Raspberry Pi to your
['Raspberry Pi for dummies / Sean McManus, Mike Cook.']
['Raspberry Pi : user guide / Eben Upton, Gareth Halfacree.']
['Raspberry Pi projects for dummies / by Mike Cook, Jonathan Evans, and Brock Craft.']
['Raspberry Pi robotics essentials : harness the power of Raspberry Pi with Six Degrees of Fr
['Building a Virtual Assistant for Raspberry Pi : The Practical Guide for Constructing a Voic
['Raspberry Pi cookbook / Simon Monk.']
['Raspberry Pi 2 server essentials : transform your Raspberry Pi into a multi-purpose web ser
```

```
['Raspberry Pi by example : start building amazing projects with the Raspberry Pi right out c
['Hello Raspberry Pi! : Python programming for kids and other beginners / Ryan Heitz.']
['Getting started with Raspberry Pi Zero : get started with the smallest, cheapest, and highe
['Python playground : geeky projects for the curious programmer / by Mahesh Venkitachalam.']
["Raspberry Pi computer architecture essentials : explore Raspberry Pi's architecture through
['Exploring the Raspberry Pi 2 with C++ / Warren Gay.']
['Eurasian wren (Troglodytes troglodytes), with raspberry (Rubus idaeus L.), and wood lice (?
['Raspberry Pi cookbook for Python programmers : over 50 easy-to-comprehend tailor-made recip
['Raspberry Pi projects for kids : start your own coding adventure with your kids by creating
['Raspberry Pi for kids for dummies / by Richard Wentk.']
['Make: action : movement, light, and sound with Arduino and Raspberry Pi / Simon Monk.']
['Learn Raspberry Pi 2 with Linux and Windows 10 / Peter Membrey, David Hows ; technical revi
['Programming the Raspberry Pi : getting started with Python / Simon Monk.']
['Raspberry Pi Networking cookbook : connect your Raspberry Pi to the world with this essenti
['Learning internet of things : explore and learn about internet of things with the help of e
['Learn to Program with Minecraft : Transform Your World with the Power of Python / Craig Ric
['Raspberry Pi robotic projects : work through a mix of amazing robotic projects using the Ra
['Building smart homes with Raspberry Pi Zero : build revolutionary and incredibly useful hom
['Smart Internet of Things projects : discover how to build your own smart Internet of Things
['Raspberry Pi for Python programmers cookbook : over 60 recipes that harness the power of th
['Raspberry Pi Electronics Projects for the Evil Genius / Donald Norris.']
['Raspberry Pi : making amazing projects right from scratch! : explore the powers of Raspberr
['Learning computer architecture with Raspberry Pi / Eben Upton, Jeffrey Duntemann, Ralph Rob
['Building a virtual assistant for Raspberry Pi : the practical guide for constructing a voic
['Raspberry Pi for secret agents : turn your Raspberry Pi into a secret agent toolbox with th
['Learn electronics with Raspberry Pi : physical computing with circuits, sensors, outputs, a
['Building the web of things : with examples in Node.js and Raspberry Pi / Dominique D. Guina
['Getting started with Raspberry Pi : getting to know the inexpensive ARM-powered Linux compu
['Yocto for Raspberry Pi : create unique and amazing projects by using the powerful combinati
['Raspberry pi server essentials / Piotr J. Kula.']
['Raspberry Pi home automation with Arduino [electronic resource] : automate your home with a
```

Looping through pages

- We can continue to loop through the search results by changing the URL after each loop
- The `recPointer` variable will increase by 50 each loop.
- What's wrong with this code?

```
import requests
from bs4 import BeautifulSoup

recPointer = 0

while recPointer <= 150:
        url = "http://orbis.library.yale.edu/vwebv/search?searchArg=raspberry&searchCode=GKE
        r = requests.get(url)
        #print(r)
```

```
        recPointer += 50
        soup = BeautifulSoup(r.text, 'html.parser')
        searchResults = soup.find_all('div', class_="resultListTextCell")

        for item in searchResults:
                info = [item.a.text]
                print(info)
```

```
['Raspberry Island Light Station : Apostle Islands National Lakeshore, Bayfield, Wisconsin /
['Building a home security system with Raspberry Pi : build your own sophisticated modular ho
['Exploring the Raspberry Pi 2 with C++ / Warren Gay.']
['Learn to program with Minecraft : transform your world with the power of Python / by Craig
['Raspberry Pi robotic blueprints : utilize the powerful ingredients of Raspberry Pi to bring
['Python playground : geeky projects for the curious programmer / Mahesh Venkitachalam.']
["Maker's guide to the zombie apocalypse : defend your base with simple circuits, Arduino, ar
['Raspberry Pi projects for dummies / Mike Cook, Jonathan Evans, Brock Craft.']
['Raspberry Pi LED blueprints : design, build, and test LED-based projects using Raspberry Pi
['Learn Raspberry Pi 2 with Linux and Windows 10 / Peter Membrey, David Hows.']
["Getting started with Python and Raspberry Pi : learn to design and implement reliable Pytho
['Raspberry Pi Android projects : create exciting projects by connecting Raspberry Pi to your
['Raspberry Pi for dummies / Sean McManus, Mike Cook.']
['Raspberry Pi : user guide / Eben Upton, Gareth Halfacree.']
['Raspberry Pi projects for dummies / by Mike Cook, Jonathan Evans, and Brock Craft.']
['Raspberry Pi robotics essentials : harness the power of Raspberry Pi with Six Degrees of Fr
['Building a Virtual Assistant for Raspberry Pi : The Practical Guide for Constructing a Voic
['Raspberry Pi cookbook / Simon Monk.']
['Raspberry Pi 2 server essentials : transform your Raspberry Pi into a multi-purpose web ser
['Raspberry Pi by example : start building amazing projects with the Raspberry Pi right out o
['Hello Raspberry Pi! : Python programming for kids and other beginners / Ryan Heitz.']
['Getting started with Raspberry Pi Zero : get started with the smallest, cheapest, and highe
['Python playground : geeky projects for the curious programmer / by Mahesh Venkitachalam.']
["Raspberry Pi computer architecture essentials : explore Raspberry Pi's architecture through
['Exploring the Raspberry Pi 2 with C++ / Warren Gay.']
['Eurasian wren (Troglodytes troglodytes), with raspberry (Rubus idaeus L.), and wood lice (I
['Raspberry Pi cookbook for Python programmers : over 50 easy-to-comprehend tailor-made recip
['Raspberry Pi projects for kids : start your own coding adventure with your kids by creating
['Raspberry Pi for kids for dummies / by Richard Wentk.']
['Make: action : movement, light, and sound with Arduino and Raspberry Pi / Simon Monk.']
['Learn Raspberry Pi 2 with Linux and Windows 10 / Peter Membrey, David Hows ; technical revi
['Programming the Raspberry Pi : getting started with Python / Simon Monk.']
['Raspberry Pi Networking cookbook : connect your Raspberry Pi to the world with this essenti
['Learning internet of things : explore and learn about internet of things with the help of e
['Learn to Program with Minecraft : Transform Your World with the Power of Python / Craig Ric
['Raspberry Pi robotic projects : work through a mix of amazing robotic projects using the Ra
['Building smart homes with Raspberry Pi Zero : build revolutionary and incredibly useful hom
['Smart Internet of Things projects : discover how to build your own smart Internet of Things
['Raspberry Pi for Python programmers cookbook : over 60 recipes that harness the power of th
['Raspberry Pi Electronics Projects for the Evil Genius / Donald Norris.']
['Raspberry Pi : making amazing projects right from scratch! : explore the powers of Raspberr
```

```
['Learning computer architecture with Raspberry Pi / Eben Upton, Jeffrey Duntemann, Ralph Rob
['Building a virtual assistant for Raspberry Pi : the practical guide for constructing a voic
['Raspberry Pi for secret agents : turn your Raspberry Pi into a secret agent toolbox with th
['Learn electronics with Raspberry Pi : physical computing with circuits, sensors, outputs, a
['Building the web of things : with examples in Node.js and Raspberry Pi / Dominique D. Guina
['Getting started with Raspberry Pi : getting to know the inexpensive ARM-powered Linux compu
['Yocto for Raspberry Pi : create unique and amazing projects by using the powerful combinati
['Raspberry pi server essentials / Piotr J. Kula.']
['Raspberry Pi home automation with Arduino [electronic resource] : automate your home with a
<Response [200]>
['Raspberry Pi user guide [electronic resource] / Eben Upton and Gareth Halfacree.']
['Börzsöny-vidéki málnatermelő táj gazdaságnéprajza / Bali János.']
['Raspberry Pi IoT Projects : Prototyping Experiments for Makers / John Shovic.']
['Learn Electronics with Raspberry Pi : Physical Computing with Circuits, Sensors, Outputs, a
['Raspberry Pi IoT projects : prototyping experiments for makers / John C. Shovic.']
['Penetration testing with Raspberry Pi : learn the art of building a low-cost, portable hack
['Baisée : roman / Marie Raspberry.']
['Looking backward at us / William Raspberry.']
['Arduino Music and Audio Projects [electronic resource] / by Mike Cook.']
['Program the Internet of Things with Swift for iOS [electronic resource] / by Ahmed Bakir, G
['Critical Dietary Factors in Cancer Chemoprevention [electronic resource] / edited by Mohamm
['Exploring the Raspberry Pi 2 with C++ [electronic resource] / by Warren Gay.']
['Learn Raspberry Pi 2 with Linux and Windows 10 [electronic resource] / by Peter Membrey, Da
['Knowledge Management in Organizations [electronic resource] : 10th International Conference
['Mitsitam Cafe cookbook : recipes from the Smithsonian National Museum of the American India
['Three taverns; a book of poems, by Edwin Arlington Robinson ....']
['Valentine stomp [sound recording].']
['Raspberry vinegar / Joan Fern Shaw.']
['Reflections : a history of Arelee and the districts of Balmae, Dreyer, Golden Valley, Light
['Welcome home ; Raspberry ; The Lucky ones : three plays / by Tony Marchant.']
['Anthology of Austrian drama / edited with a historical introduction by Douglas A. Russell.'
['Anthology of Austrian drama / edited with a historical introduction by Douglas A. Russell.'
['Public prosecutor and other plays / Fritz Hochwälder ; introd. by Martin Esslin.']
['Raspberry Reich : a novel / by Wolf Mankowitz.']
['Deploying Raspberry Pi in the Classroom [electronic resource] / by Guy Hart-Davis.']
['Beginning FPGA: Programming Metal [electronic resource] : Your brain on hardware / by Aiker
['Flowers of Skye / K.M.T.']
['Windows 10 for the Internet of Things [electronic resource] / by Charles Bell.']
['Proceedings of the International Conference on Data Engineering and Communication Technolog
['Raspberry Pi IoT Projects [electronic resource] : Prototyping Experiments for Makers / by J
['Building a Virtual Assistant for Raspberry Pi [electronic resource] : The practical guide f
['Beginning Ruby [electronic resource] : From Novice to Professional / by Peter Cooper.']
['Learn Electronics with Raspberry Pi [electronic resource] : Physical Computing with Circuit
['Color Adjustment.']
['Proceedings of the International Conference on Recent Cognizance in Wireless Communication
['Program the internet of things with Swift for iOS / Ahmed Bakir, Gheorghe Chesler, Manny de
['Proceedings of Fifth International Conference on Soft Computing for Problem Solving [electr
['Measurement, Modelling and Evaluation of Dependable Computer and Communication Systems [ele
['Fritzing for inventors : take your electronics project from prototype to product / Simon Mo
['Proceedings of the International Conference on Soft Computing Systems [electronic resource]
['Raspberry ice cream war : a comic for young people on a peaceful Europe without frontiers /
```

```
['Raspberry jam / by Carolyn Wells, ; with frontispiece in color by Gayle Hoskins.']
['Raspberry hut : and other Ukrainian folk tales retold in English / edited by Danny Evanishe
['Mezzotints / Henry Miller ; with an historical introduction by Roger Jackson.']
['Fantastic tales / I.U. Tarchetti ; edited and translated by Lawrence Venuti ; with original
['Raspberry growing in New York State / D. K. Ourecky and J. P. Tomkins.']
['Raspberry tree : and other poems of sentiment and reflection / by Stoddard King.']
['How to make more than one hundred summer and winter drinks : wholesome, refreshing, cooling
['Treatise on the improved culture of the strawberry, raspberry, and gooseberry : designed to
['American kitchen gardener; containing practical directions for the culture of vegetables. A
['Translocation of carbohydrates in the Cuthbert raspberry ... by Charles J. Engard...']
['Linux Sound Programming [electronic resource] / by Jan Newmarch.']
['Custom Raspberry Pi Interfaces [electronic resource] : Design and build hardware interfaces
['Python, PyGame and Raspberry Pi Game Development [electronic resource] / by Sloan Kelly.']
['PHP Beyond the Web [electronic resource] / by Rob Aley.']
['Raspberry Pi GPU Audio Video Programming [electronic resource] / by Jan Newmarch.']
['Raspberry Reich [videorecording] / Strand Releasing in association with Jürgen Brüning Film
['Missing pages : black journalists of modern America : an oral history / Wallace Terry.']
['Raspberry room / Alison Lohans ; with illustrations by Gillian Newland.']
["Fairy's pie."]
['Ripe fruit.']
['Misbranding of lemon, raspberry, and strawberry extracts [microform]']
['Pilot raspberry and blackberry loss adjustment standards handbook [electronic resource] / U
['Autobiography of Harry Secombe.']
['Joe Satriani anthology : guitar.']
['Peace in the house : tales from a Yiddish kitchen / by Faye Moskowitz.']
['Complete manual for the cultivation of the strawberry; with a description of the best varie
['Book of fruits, being a descriptive catalogue of the most valuable varieties of the pear, a
['Mother in modern story / edited by Maud Van Buren and Katharine Isabel Bemis ... .']
['Proposal to study the effectiveness of raspberry leaf tea in reducing or relieving nausea w
['Five shots and a funeral / by Dashiell Loveless, a.k.a. Tom Fassbender and Jim Pascoe ; ...
['Complete Fats Waller. Vol. IV [sound recording].']
['Intravenus mind presents John Giorno [sound recording].']
['Estimated probabilities, volumes, and inundation area depths of potential postwildfire debr
['Raspberry byturus : Byturus unicolor / [W.H. Goodwin].']
['Patch Dynamics of Spruce-Fir forests During a Spruce Budworm Outbreak in Maine (Tree Mortal
["Rip-roarin' ragtime [electronic resource] / David Jasen."]
['Soviet medicine : culture, practice, and science / editors, Frances L. Bernstein, Christoph
["Ladies' assistant for regulating and supplying the table [electronic resource] : Being a co
["Ladies' assistant for regulating and supplying the table [electronic resource] : Being a co
['William Empson [electronic resource] : against the Christians / John Haffenden.']
['Lovely, raspberry : poems / Aaron Belz.']
['Puyallup and the Puyallup Valley : the natural home of the red raspberry, State of Washingt
['Rip-roarin ragtime [electronic resource] / David Jasen.']
['Recipe Club : a tale of food and friendship / Andrea Israel & Nancy Garfinkel ; recipes in
['Growing raspberries / prepared by Northeastern Region, Agricultural Research Service.']
['Promise to ourselves : a journey through fatherhood and divorce / Alec Baldwin ; with Mark
['Raspberry Pi user guide [electronic resource] / Eben Upton and Gareth Halfacree.']
['Learn Raspberry Pi with Linux [electronic resource] / Peter Membrey, David Hows.']
['Fats Waller. Volume 3, 1934-1936, Rhythm and romance [sound recording].']
['Learn Raspberry Pi with Linux [electronic resource] / by Peter Membrey, David Hows.']
['First fifteen years. Vol 2 [sound recording] / Bok, Muir & Trickett.']
```

```
['High-Tech and Micropropagation II [electronic resource] / edited by Y. P. S. Bajaj.']
['Book of fruits [electronic resource] : being a descriptive catalogue of the most valuable v
["Carolan's welcome [sound recording] : Harp music of Ireland / Carol Thompson and friends."]
['Yearbook of Agriculture, 1937. United States Department of Agriculture. [electronic resourc
['Report of the Commissioner of Patents for the year 1861. Agriculture. [electronic resource]
['Writer: Carole King [sound recording].']
['"Fats" 1935-37 [sound recording] / Fats Waller & his rhythm.']
['Fats Waller [sound recording].']
['Raspberry Island Light Station : Apostle Islands National Lakeshore, Bayfield, Wisconsin /
['Building a home security system with Raspberry Pi : build your own sophisticated modular ho
['Exploring the Raspberry Pi 2 with C++ / Warren Gay.']
['Learn to program with Minecraft : transform your world with the power of Python / by Craig
['Raspberry Pi robotic blueprints : utilize the powerful ingredients of Raspberry Pi to bring
['Python playground : geeky projects for the curious programmer / Mahesh Venkitachalam.']
["Maker's guide to the zombie apocalypse : defend your base with simple circuits, Arduino, ar
['Raspberry Pi projects for dummies / Mike Cook, Jonathan Evans, Brock Craft.']
['Raspberry Pi LED blueprints : design, build, and test LED-based projects using Raspberry Pi
['Learn Raspberry Pi 2 with Linux and Windows 10 / Peter Membrey, David Hows.']
["Getting started with Python and Raspberry Pi : learn to design and implement reliable Pytho
['Raspberry Pi Android projects : create exciting projects by connecting Raspberry Pi to your
['Raspberry Pi for dummies / Sean McManus, Mike Cook.']
['Raspberry Pi : user guide / Eben Upton, Gareth Halfacree.']
['Raspberry Pi projects for dummies / by Mike Cook, Jonathan Evans, and Brock Craft.']
['Raspberry Pi robotics essentials : harness the power of Raspberry Pi with Six Degrees of Fr
['Building a Virtual Assistant for Raspberry Pi : The Practical Guide for Constructing a Voic
['Raspberry Pi cookbook / Simon Monk.']
['Raspberry Pi 2 server essentials : transform your Raspberry Pi into a multi-purpose web ser
['Raspberry Pi by example : start building amazing projects with the Raspberry Pi right out o
['Hello Raspberry Pi! : Python programming for kids and other beginners / Ryan Heitz.']
['Getting started with Raspberry Pi Zero : get started with the smallest, cheapest, and highe
['Python playground : geeky projects for the curious programmer / by Mahesh Venkitachalam.']
["Raspberry Pi computer architecture essentials : explore Raspberry Pi's architecture through
['Exploring the Raspberry Pi 2 with C++ / Warren Gay.']
['Eurasian wren (Troglodytes troglodytes), with raspberry (Rubus idaeus L.), and wood lice (?
['Raspberry Pi cookbook for Python programmers : over 50 easy-to-comprehend tailor-made recip
['Raspberry Pi projects for kids : start your own coding adventure with your kids by creating
['Raspberry Pi for kids for dummies / by Richard Wentk.']
['Make: action : movement, light, and sound with Arduino and Raspberry Pi / Simon Monk.']
['Learn Raspberry Pi 2 with Linux and Windows 10 / Peter Membrey, David Hows ; technical revi
['Programming the Raspberry Pi : getting started with Python / Simon Monk.']
['Raspberry Pi Networking cookbook : connect your Raspberry Pi to the world with this essenti
['Learning internet of things : explore and learn about internet of things with the help of e
['Learn to Program with Minecraft : Transform Your World with the Power of Python / Craig Ric
['Raspberry Pi robotic projects : work through a mix of amazing robotic projects using the Ra
['Building smart homes with Raspberry Pi Zero : build revolutionary and incredibly useful hom
['Smart Internet of Things projects : discover how to build your own smart Internet of Things
['Raspberry Pi for Python programmers cookbook : over 60 recipes that harness the power of th
['Raspberry Pi Electronics Projects for the Evil Genius / Donald Norris.']
['Raspberry Pi : making amazing projects right from scratch! : explore the powers of Raspberr
['Learning computer architecture with Raspberry Pi / Eben Upton, Jeffrey Duntemann, Ralph Rob
['Building a virtual assistant for Raspberry Pi : the practical guide for constructing a voic
```

```
['Raspberry Pi for secret agents : turn your Raspberry Pi into a secret agent toolbox with th
['Learn electronics with Raspberry Pi : physical computing with circuits, sensors, outputs, a
['Building the web of things : with examples in Node.js and Raspberry Pi / Dominique D. Guina
['Getting started with Raspberry Pi : getting to know the inexpensive ARM-powered Linux compu
['Yocto for Raspberry Pi : create unique and amazing projects by using the powerful combinati
['Raspberry pi server essentials / Piotr J. Kula.']
['Raspberry Pi home automation with Arduino [electronic resource] : automate your home with a
```

Resources

- “Web Scraping with Python” by Richard Lawson http://hdl.handle.net/10079/bibid/12646583
- StackOverflow https://stackoverflow.com/
- Programming Historian