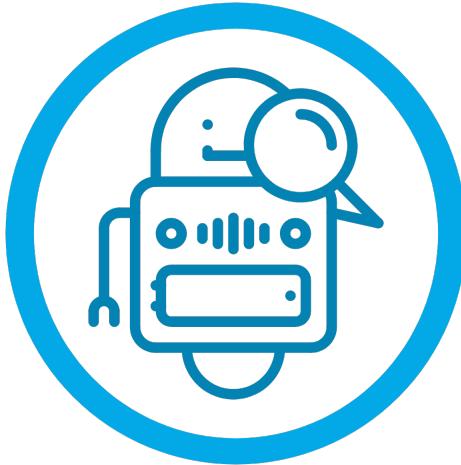


```
 {{ if response.serve_ad and settings.adsenseid: }} {{ pass }}  
 {{ if settings.num_banners > 0 and settings.show_rs_banner: }}  
  
Please Support Runestone  
(/runestone/default/donate?banner={{=banner_num}})  
{{ pass }}
```



1

Learn Web Scraping!



(_images/web_crawler.png)

What is web scraping?

The internet is full of information people would like to collect. Copying and pasting it would take forever! In this ebook, we'll explore an easier way.

Web scraping is the process of getting information from web pages with code.

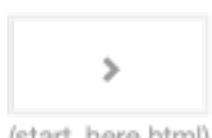
How this ebook works

During this lesson, you will learn code plans used by *real professionals*.

These plans represent the most common patterns in web scraping with the BeautifulSoup library. The plans in this ebook were created after an analysis of 100 web scraping files from Github (<https://github.com/search?l=Python&q=BeautifulSoup%26amp;type=Code>), as well as two interviews with people who use web scraping in their jobs.

Start here

- Before You Start The Activity ([start_here.html](#))



([start_here.html](#))

Two scraping examples

- Scrape all the bb.q Chicken locations (example1.html)
 - Plan 1: Get a soup from a URL (plan1.html)
 - Plan 3: Get info from all tags of a certain type (plan3.html)
 - Plan 5: Print info (plan5.html)
- Get news links from faculty webpages (example2.html)
 - Plan 2: Get a soup from multiple URLs (plan2.html)
 - Plan 4: Get info from a single tag (plan4.html)
 - Plan 5: Print info (plan5.html)



Putting it together

- Code writing activity part 1 (activity-writing-1.html)
- Code writing activity part 2 (activity-writing-2.html)
- Code writing activity part 3 (activity-writing-3.html)
- Code debugging activity (activity-debugging.html)
- Code explaining activity (activity-explaining.html)

End here

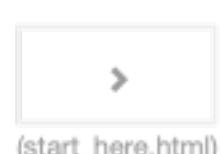
- Before You Complete The Activity - 1 (end_here_comparison.html)
- Before You Complete The Activity - 2 (end_here_usability.html)

You have attempted of activities on this page

© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (<http://runestoneinteractive.org/>) 5.3.0.

| Back to

`{{ if request.application == 'runestone':}} {{ pass }}`



```
{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}  
{{ if settings.num_banners > 0 and settings.show_rs_banner: }} {{ pass }}
```

Please Support Runestone

(/runestone/default/donate?banner={{=banner_num}})
{{ pass }}



Before You Start The Activity

Please rate how much you agree with the each statement. Your responses will be anonymized.

I am interested in learning how to extract data from websites using the web scraping tools such as Beautiful Soup.

- 1. Strongly Disagree
- 2. Disagree
- 3. Somewhat Disagree
- 4. Neither Agree nor Disagree
- 5. Somewhat Agree
- 6. Agree
- 7. Strongly Agree

I am confident in my ability to use web scraping tools such as Beautiful Soup to scrape or parse web pages for data.

- 1. Strongly Disagree
- 2. Disagree
- 3. Somewhat Disagree
- 4. Neither Agree nor Disagree
- 5. Somewhat Agree
- 6. Agree
- 7. Strongly Agree

You have attempted of activities on this page

© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (<http://runestonestrong.org/>) 5.3.0.

| Back to

```
{{ if request.application == 'runestone': }} {{ pass }}
```



(index.html)



(example1.html)

```
{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}
{{ if settings.num_banners > 0 and settings.show_rs_banner: }}
```

Please Support Runestone

(/runestone/default/donate?banner={{=banner_num}})
{{ pass }}



Scrape all the bb.q Chicken locations

Let's say that you want to make a list of all the bb.q Chicken locations. When you go to their website, it turns out that there are a *lot* of locations.

The screenshot shows the bb.q CHICKEN website. At the top, there are navigation links: MENU, ABOUT, FRANCHISING, LOCATIONS (which is underlined), REWARDS, GIFT CARDS, and ORDER ONLINE. Below the navigation, a red banner says "Crispy chicken is closer than you think." A search bar allows users to enter a city, state, or zip code, with an example "ex. Fort Lee, NJ or 07024" and a magnifying glass icon. To the right of the search bar is a large map of North America and Mexico, with numerous red location pins scattered across the United States, Canada, and parts of Mexico. On the left side of the map, there is a sidebar for a specific location: "bb.q Chicken NY K-Town (Chicken & Beer) New York, NY". It lists the address "25 W 32nd St., New York, NY 10001", phone number "(201) 665-3585", and operating hours "Now Open • Closes at 10:00pm". There are also links for "PRIVATE EVENT INQUIRY", "Catering", and "Favorite Location".

(images/bbq_scroll.gif)

If only you could write a little Python to easily collect them all...

It turns out that you can! Run the code below to see what it collects.

```
#Get the webpage
# Load libraries for web
```

This code probably seems a bit complicated. In this ebook, we will break down web scraping into a few common “plans”. This example is made up of three plans. Click on each of them to learn more.

(start_here.html)

(plan1.html)

- Plan 1: Get a soup from a URL (plan1.html)
- Plan 3: Get info from all tags of a certain type (plan3.html)
- Plan 5: Print info (plan5.html)

Plan 1: Get a soup from a URL

```
# Load libraries for web scraping
from bs4 import BeautifulSoup
import requests
# Get a soup from a URL
url = 'https://web.archive.org/web/20250309231002/https://bbqchicken.com/location'
r = requests.get(url)
soup = BeautifulSoup(r.content, 'html.parser')
```

(/ns/books/published/cs102web/plan1.html)

**Plan 3: Get info from all tags of a certain type**

```
# Get all tags of a certain type from the soup
tags = soup.find_all('h4')
# Collect info from the tags
collect_info = []
for tag in tags:
    # Get info from tag
    info = tag.text
    collect_info.append(info)
```

(/ns/books/published/cs102web/plan3.html)

Plan 5: Print the info

```
# Print the info
print(collect_info)
```

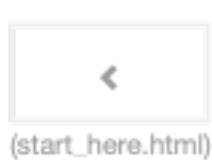
(/ns/books/published/cs102web/plan5.html)

You have attempted of activities on this page

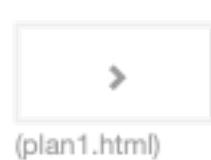
© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (<http://runestoneinteractive.org/>) 5.3.0.

| Back to

{{ if request.application == 'runestone':}} {{ pass }}



(start_here.html)



(plan1.html)

```
 {{ if response.serve_ad and settings.adsenseid: }} {{ pass }}  
 {{ if settings.num_banners > 0 and settings.show_rs_banner: }}  
  
Please Support Runestone  
(/runestone/default/donate?banner={{=banner_num}})  
{{ pass }}
```



Plan 1: Get a soup from a URL

Plan 1: Example

The first step in web scraping is getting information from a webpage. To use the BeautifulSoup web scraping library, we have to put the webpage into something called a *soup*.

Here is the code for getting a **soup** from the **bb.q Chicken locations page**.

Goal: Get a soup from one webpage

```
# Load libraries for web scraping
from bs4 import BeautifulSoup
import requests

# Get a soup from a URL
url = 'https://web.archive.org/web/20250309231002/https://bbqchicken.com/location'
r = requests.get(url)
soup = BeautifulSoup(r.content, 'html.parser')
```

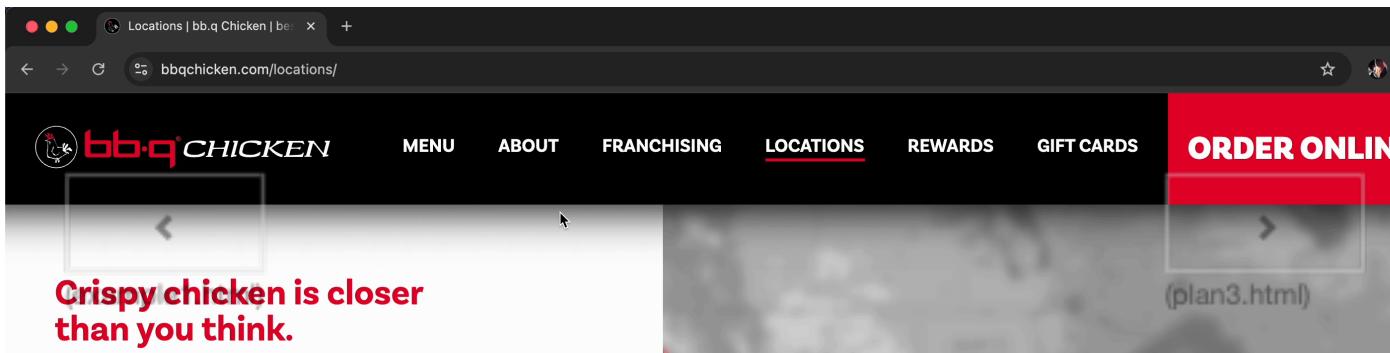
Plan 1: When to use this plan

Use this plan when you want to scrape **one webpage**.

Plan 1: How to use this plan

Replace the URL with the URL of the website you want to scrape.

A URL is a web address, like you see in your web browser. It should be complete (starting with http:// (http://) or https:// (https://)). In this plan, a URL should be surrounded by quotes (' ').



(_images/bbq_URL.gif)



Plan 1: Exercises

p1-1: If you wanted to get a soup from the Illini Union Bookstore homepage instead of the bb.q Chicken locations page, which part(s) of the code below would you change? Click on those part(s) of the code. Check out the example of this plan above to identify the area that should be changed.

```
# Load libraries for web scraping
from bs4 import BeautifulSoup
import requests

# Get a soup from a URL
url = 'https://bbqchicken.com/locations/'
r = requests.get(url)
soup = BeautifulSoup(r.content, 'html.parser'):
```

p1-2: Fill in the plan in order to get a soup from the University of Illinois Urbana-Champaign wikipedia page.

```
# Load libraries for web scraping
from bs4 import BeautifulSoup
import requests

# Get a soup from a URL
url = 
r = requests.get(url)
soup = BeautifulSoup(r.content, 'html.parser')
```

Note

[Click here to go back to the bb.q Chicken example \(/ns/books/published/cs102web/example1.html\)](#)

You have attempted of activities on this page

© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (<http://runestoneinteractive.org/>) 5.3.0.

| Back to

`{{ if request.application == 'runestone':}} {{ pass }}`



(example1.html)



(plan3.html)

```

{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}
{{ if settings.num_banners > 0 and settings.show_rs_banner: }}

Please Support Runestone
(/runestone/default/donate?banner={{=banner_num}})
{{ pass }}

```



Plan 3: Get info from all tags of a certain type

To get information from the bb.q Chicken locations page, we need to figure out which tags we should get from the soup, and what information we should get from the tags.

A great way to figure this out is to use the “inspect” function on your browser.

The screenshot shows the bb.q Chicken Locations page. At the top, there's a navigation bar with links for MENU, ABOUT, FRANCHISING, LOCATIONS (which is underlined), REWARDS, GIFT CARDS, and ORDER ONLINE. Below the navigation, a banner says "Crispy chicken is closer than you think." On the left, there's a search bar with placeholder text "Enter City AND State, or Zipcode" and a button "Click to Find Location". Below the search bar, it shows a result for "bb.q Chicken NY K-Town (Chicken & Beer)" in "New York, NY". The address is "25 W 32nd St., New York, NY 10001", the phone number is "(201) 665-3585", and it's "Now Open • Closes at 10:00pm". There's also a link for "PRIVATE EVENT INQUIRY" and buttons for "Catering" and "Favorite Location". To the right, there's a map with numerous red location pins, one of which is highlighted with a large red pin containing a chicken icon. The bottom of the page has a footer with the bb.q logo and the text "bb.q CHICKEN".

We see that we need to get info from all the `h4` tags from the webpage. The *text* in those tags has the information we need!

Looking closer at a tag

Behind every webpage is HTML code. HTML code is made up of *tags*.

([plan1.html](#))

Here is the tag that creates the name of one of the bb.q Chicken's IL locations. The tag is surrounded by the blue rectangle. It is an '`h4`' tag.

[bb.q CHICKEN IL Locations](#)

<div class="loc-content">

([plan5.html](#))

bb.q Chicken Urbana Champaign
Urbana, IL

700 S Gregory St,
Urbana, IL 61801

(_images/bbq_h4_text.png)

The name of this tag is 'h4'. In-between the start and end tag (between the `<h4>` and `</h4>`) is the tag's **text**. For this tag, the text is **bb.q Chicken Urbana Champaign Urbana, IL**. The `
` tag stands for **break** and adds a line break to the text.

Plan 3: Example

Here is how to get **text** from all the '**h4**' tags from webpage:

Goal: Get info from all tags of a certain type

```
# Get all tags of a certain type from the soup
tags = soup.find_all('h4')
# Collect info from the tags
collect_info = []
for tag in tags:
    # Get info from tag
    info = tag.text
    collect_info.append(info)
```

Plan 3: How to use it

Once you've found the tags you want to get information from, do two things:

1. Find the **tag description** and put it into the first slot.

How do you do that? Here are some examples:

What you see when you inspect		Tag description in the code
<p>	->	'p'
<h3>	->	'h3'
<div class="comment">	->	'div', class_='comment'
	->	'span', style='X5e72;'
	->	'a', class_='css4z' >

- (plan1.html) 2. Determine if you want to get **text** from a tag, or a **link** from a tag. Put that information into the second slot.
- (plan5.html)

The info you want	What you put in the code	
The tag's text	->	text
The tag's link	->	get('href')

Plan 3: Exercises

p3-1: What is the text of the tag below?

```
-----  
      </div>  
    </div>  
    <h2 class="menuTitle">Today's Menu</h2> == $0  
  </div>  
</div>
```

- Today's Menu
- Correct! This text is between the `<h2 class="menuTitle">` and `</h2>`
- h2
- No, h2 is the tag name
- menuTitle
- No
- class
- No

p3-2: What is the tag description of the tag below?

```
-----  
      </div>  
    </div>  
    <h2 class="menuTitle">Today's Menu</h2> == $0  
  </div>  
</div>
```

- 'h2', class_= 'menuTitle'
- Correct! This is how you would describe the tag type in our web scraping code.
- 'h2'
- That is a part of the tag description, but we can be more specific.
- 'h2', class_= 'menuTitle'
- Very close, but in web scraping code you should use "class_"
- `<h2 class="menuTitle">`
- This is what is actually in the tag, but it's not how we would describe the tag in web scraping code.

(plan1.html)

(plan5.html)

p3-3: Right now, this code gets the *text* from all 'h3' tags in the webpage. If you wanted to get the *links* from all the 'a', class_= 'headline' tags in the webpage, which part(s) of the code below would

you change? Check out "how to use this plan".

```
# Get all tags of a certain type from the soup
tags = soup.find_all('h3')

# Collect info from the tags
collect_info = []
for tag in tags:
    # Get info from tag
    info = tag.text
    collect_info.append(info)
```



p3-4: Fill in the plan in order to get the text from all `<div class="headline">` tags on a webpage.

```
# Get all tags of a certain type from the soup
tags = soup.find_all( [ ] )

# Collect info from the tags
collect_info = []

for tag in tags:
    # Get info from tag
    info = tag. [ ]
    collect_info.append(info)
```



p3-5: Which tag in the picture below has text?

What is Information Sciences?

Information Science focuses on how people use technology to better organize, analyze, and manage information, as well as how society and individuals relate to information. In our program, you'll learn programming, database concepts, and research methods. You'll also explore broader social issues such as information equity and ethics. Our program will prepare you to become a specialist in the information sciences who understands the human perspective, social context, and policy implications of information in society.

- 'h2'
- No, there is no h2 tag in this image.
- 'p'
- Correct! The text starts with "Information Science focuses on how people use..."
- `span, style='font-weight: 400;'`
- No, this tag contains the p tag.
- 'style'
- No, style is an attribute

```
<div class="row"> [flex]
  <div class="col-12 col-lg-6">
    <div class="programDescription section">
      <h2>What is Information Sciences?</h2>
      <p></p>
      <p>Information Science focuses on how people use technology to better organize, analyze, and manage information, as well as how society and individuals relate to information. In our program, you'll learn programming, database concepts, and research methods. You'll also explore broader social issues such as information equity and ethics. Our program will prepare you to become a specialist in the information sciences who understands the human perspective, social context, and policy implications of information in society.</p>
      <p></p>
      <p></p>
    </div> == $0
    ...
  </div>
  <div class="programCareerExamples section"> ... </div>
  <div class="section"> ... </div>
</div>
<div class="col-12 col-lg-5 offset-lg-1"> ... </div>
```

(plan1.html)

(plan5.html)

Note

[Click here to go back to the bb.q Chicken example \(/ns/books/published/cs102web/example1.html\)](#)



You have attempted 0 activities on this page

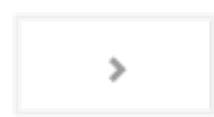
© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (<http://runestoneinteractive.org/>) 5.3.0.

| Back to

```
{{ if request.application == 'runestone':}} {{ pass }}
```



(plan1.html)



(plan5.html)

```
 {{ if response.serve_ad and settings.adsenseid: }} {{ pass }}  
 {{ if settings.num_banners > 0 and settings.show_rs_banner: }}  
  
Please Support Runestone  
(/runestone/default/donate?banner={{=banner_num}})  
{{ pass }}
```



Plan 5: Print info

It's very common to print the information you have just scraped.

Plan 5: Example

If you want to print info from only one tag, like if you've just used *Plan 4: Get info from a single tag*, then do this:

Goal: Print the info

```
# Print the info  
print( info )
```

If you want to print info from multiple tags, like if you've just used *Plan 5: Get info from all tags of a certain type*, then do this:

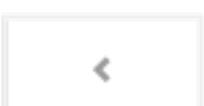
Goal: Print the info

```
# Print the info  
print( collect_info )
```

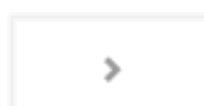
Plan 5: Exercises

p5-1: Here is the code to collect all the locations from the bb.q Chicken locations page.

What should fill the slot in Plan 9 below?



(plan3.html)



(example2.html)

```
#Get one webpage
# Load libraries for web scraping
from bs4 import BeautifulSoup
import requests
# Get a soup from a URL
url = 'https://bbqchicken.com/locations/'
r = requests.get(url)
soup = BeautifulSoup(r.content, 'html.parser')

#Get info from all tags of a certain type
# Get all tags of a certain type from the soup
tags = soup.find_all('h4')
# Collect info from the tags
collect_info = []
for tag in tags:
    # Get info from tag
    info = tag.text
    collect_info.append(info)

#print the info
# Print the info
print(_____)
```

_____ should be

Note

[Click here](#) to go back to the bb.q Chicken example (/ns/books/published/cs102web/example1.html)

Note

[Click here](#) to go back to the Faculty Pages example (/ns/books/published/cs102web/example2.html)

You have attempted of activities on this page

© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (<http://runestoneinteractive.org/>) 5.3.0.

| Back to

`{{ if request.application == 'runestone':}} {{ pass }}`



(plan3.html)



(example2.html)

```
{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}
{{ if settings.num_banners > 0 and settings.show_rs_banner: }}
```

Please Support Runestone

(/runestone/default/donate?banner={{=banner_num}})

{{ pass }}



Get news links from faculty webpages

Let's say that you want to get the link to the first news article on your favorite SCDS faculty's webpages.

(_images/faculty_pages_scoll.gif)

But clicking through to gather all those links would be a pain. Fortunately, we can do that task with BeautifulSoup!

Run the code below to see what it collects.

```
#Get the webpages
# Load libraries for web
```

This code is made up of three plans. Click on each of the plans below to learn more about it.

- Plan 2: Get a soup from multiple URLs (plan2.html)
- Plan 4: Get info from a single tag (plan4.html)
- Plan 5: Print info (plan5.html)

(plan5.html)

(plan2.html)

Plan 2: Get a soup from multiple URLs

```
# Load libraries for web scraping
from bs4 import BeautifulSoup
import requests

# Get a soup from multiple URLs
base_url = 'https://siebelsschool.illinois.edu/about/people/faculty/'
endings = ['katcun', 'challen', 'mnowak1']
for ending in endings:
    url = base_url + ending
    r = requests.get(url)
    soup = BeautifulSoup(r.content, 'html.parser')
```

(/ns/books/published/cs102web/plan2.html)

**Plan 4: Get info from a single tag**

```
# Get first tag of a certain type from the soup
tag = soup.find('a', class_='text-decoration-none')
# Get info from tag
info = tag.get('href')
```

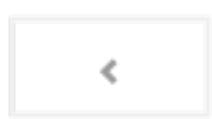
(/ns/books/published/cs102web/plan4.html)

Plan 5: Print info

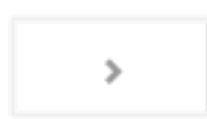
```
# Print the info
print(info)
```

(/ns/books/published/cs102web/plan5.html)

You have attempted 0 of activities on this page



(plan5.html)



(plan2.html)

```
{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}
{{ if settings.num_banners > 0 and settings.show_rs_banner: }} {{ pass }}
```

Please Support Runestone

(/runestone/default/donate?banner={{=banner_num}})

{{ pass }}



Plan 2: Get a soup from multiple URLs

Plan 2: Example

Sometimes we want to get information from multiple web pages that have the same layout. For example, all of the SCDS faculty pages have the same general design.

Katie Cunningham

Assistant Professor
katcun@illinois.edu
 3215 Siebel Center for Comp Sci

For More Information

- [Lab website](#)
- [Personal website](#)

Education

- Ph.D., Information, School of Information, University of Michigan, 2020
- M.S., Human-Centered Computing, School of Interactive Computing, Georgia Institute of Technology, 2018
- B.S., Computer Science and Molecular and Cellular Biology, University of Arizona, 2013

Academic Positions

- Assistant Professor, Computer Science, University of Illinois Urbana-Champaign, July 2022– Present
- Postdoctoral Scholar, Northwestern University, January 2021 - July 2022

Related News

- Illinois CS Places 28 Faculty on CITL List of Teachers Ranked as Excellent By Their Students
- 12 New Faculty Add to Expertise in Research and Education at Illinois Computer Science

(_images/katie_cunningham.png)

Geoffrey Werner Challen

Teaching Professor
 (217) 300-6150
challen@illinois.edu
 2227 Siebel Center for Comp Sci

For More Information

- [CS 124 Website](#)
- [Personal Website](#)
- geoffreychallen.com
- [Learn CS Online: Public CS1 Materials](#)

Education

- Ph.D. Computer Science, Harvard University, 6/2010
- AB Physics, Harvard University, 6/2003

Academic Positions

- Associate Teaching Professor, University of Illinois, 8/2017–

Research Areas

- [Computers and Education](#)

Related News

- CS Professor Shares Course Materials for Free Online
- Illinois Computer Science Summer Teaching Workshop Presents Innovative Approaches to Teaching the Subject
- UI Professor Builds New Computer Science Project, Seeks Students to Aid in Development
- CS Professors Prepare to Enhance Engineering Teaching Through SiIP Grants
- Illinois CS Teaching Inspires Fisher to Extend Support for Award, and Add Honors for TAs, CAs
- CS Department Hires 8 Faculty Members To Meet An

(_images/geoffrey_challen.png)

We are interested in getting information about multiple SCDS professors: Prof. Katie Cunningham, Prof. Geoffrey Challen, and Prof. Michael Nowak.



Their webpages are:

<https://siebelschool.illinois.edu/about/people/faculty/katcun>

<https://siebelschool.illinois.edu/about/people/faculty/challen>

<https://siebelschool.illinois.edu/about/people/faculty/mnowak1>

In this code, we get a **soup** from multiple **SCDS faculty pages**.

Goal: Get a soup from multiple webpages

```
# Load libraries for web scraping
from bs4 import BeautifulSoup
import requests

# Get a soup from multiple URLs
base_url = 'https://siebelschool.illinois.edu/about/people/faculty/'
endings = ['katcun', 'challen', 'mnowak1']
for ending in endings:
    url = base_url + ending
    r = requests.get(url)
    soup = BeautifulSoup(r.content, 'html.parser')
```



Plan 2: When to use this plan

Use this plan when you want to scrape the same thing from multiple webpages.

Plan 2: How to use this plan

Look at the webpages you want to scrape and determine which parts they have in common, and which parts are different. The parts that they have in common are the `base_url`. The parts that are different are the `endings`.

Plan 2: Exercises

If you want to also get the link to the most recent news item from Director Nancy Amato's page, how would you change the code below? Director Amato's web page is <https://siebelschool.illinois.edu/about/people/faculty/namato>.

Change the code and run it to see if you're right!

```
#Get the webpage
# Load libraries for web
(example2.html)
```

(plan4.html)

Note

Click here to go back to the Faculty Pages example (/ns/books/published/cs102web/example2.html)

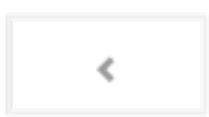


You have attempted 0 activities on this page

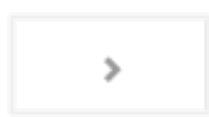
© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (<http://runestoneinteractive.org/>) 5.3.0.

| Back to

```
{{ if request.application == 'runestone':}} {{ pass }}
```



(example2.html)



(plan4.html)

```

{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}
{{ if settings.num_banners > 0 and settings.show_rs_banner: }}

Please Support Runestone
(/runestone/default/donate?banner={{=banner_num}})
{{ pass }}

```



Plan 4: Get info from a single tag

Plan 4: Example

Maybe we want to get just one piece of information from a webpage. In this example, we want to get the first link to a news story from a professor's page.

Here's what we see when we use the "inspect" function in the browser.

(_images/news_cunningham.png)

Since that tag is the first of its type on the page, we can use the plan **Get info from a single tag**.

Goal: Get info from a single tag

```

# Get first tag of a certain type from the soup
tag = soup.find('a', class_='text-decoration-none')
# Get info from tag
info = tag.get('href')

```

Plan 4: When to use it

Use this when you want to get information that is in the first tag of a certain type on the page.

Plan 4: How to use it

Once you've found the tag you want to get information from, do two things:

- Find the **tag description** and put it into the first slot.

How do you do that? Here are some examples:



(activity-writing-1.html)

What you see when you inspect

Tag description in the code

What you see when you inspect		Tag description in the code	
<p>	->	'p'	
<h3>	->	'h3'	
<div class="comment">	->	'div', class_='comment'	
	->	'span', style='X5e72;'	
	->	'a', class_='css4z'	

2. Determine if you want to get **text** from a tag, or a **link** from a tag

The info you want		What you put in the code
The tag's text	->	text
The tag's link	->	get('href')

One type of tag, the `a` tag, holds a link.

Here is the tag that creates the link to the ISR dining hall page. It is an 'a' tag.

Illinois Street Dining Center (ISR Dining Center)

Located on the second-level, or the Illinois Street entrance level of the [Illinois Street Residence Halls](#), this newly-renovated, state-of-the-art dining center offers eight micro-style restaurants with a variety of meal options for breakfast, lunch, late lunch and dinner.

```
"Located on the second-level, or the Illinois Street
entrance level of the "
<a href="/Living-communities/halls/isr">Illinois Street
Residence Halls</a> == $0
", this newly-renovated, state-of-the-art dining center
offers eight micro-style restaurants with a variety of m
options for breakfast, lunch, late lunch and dinner."
</p>
><p>@@</p>
><p>@@</p>
<h3>Lincoln Avenue Dining Hall (LAR Dining Hall)</h3>
><p>@@</p>
><p>@@</p>
><p>@@</p>
<h3>Pennsylvania Avenue Dining Hall (PAR Dining Hall)</h3>
><p>@@</p>
```

If you want to get the link from a tag, use `get('href')` in the second slot in this plan.

Plan 4: Exercises

p4-1: What is the link of the tag below?



Katie Cunningham

Assistant Professor



```
=126128&s=400&type=portrait)"></div>
><div class="details">
><div class="name">
><a href="/about/people/all-faculty/katcun"
aria-label="Katie Cunningham">Katie Cunningham
</a> == $0
</div>
><div class="pronouns hide-empty" data-value>
</div>
><div class="title">Assistant Professor</div>
```

- <https://siebelschool.illinois.edu/about/people/faculty/katcun> (<https://siebelschool.illinois.edu/about/people/faculty/katcun>)
- No, this is the full link, but there is a relative link in the tag.

- /about/people/faculty/katcun
- Correct!
- a
- No, this is the name of the tag
- Katie Cunningham
- No, this is the text of the tag



p4-2: If you wanted to get a link from the first 'a' tag on a webpage, which part(s) of the code below would you change? Click on those parts of the code. Check out the plan outline above to identify the slot.

```
# Get first tag of a certain type from the soup
tag = soup.find('div', class_='Comments_StyledComments-dzyvwm-0 dvnRbr')

# Get info from tag
info = tag.text
```

Check out the image below, that inspects the name of the class INFO 102.

```
main is content class site main post site course type course
blish henry tag-minor department-info"
  ><div class="page-header">
    ><h1 class="entry-title">INFO 102 · Little Bits to Big Ideas</h1>
  </div>
  ><div class="page-content">...</div>
</main>
<div data-elementor-type="footer" data-elementor-id="10327" class="elementor-10327 elementor elementor-10327 elementor-location-footer" data-elementor-post-type="elementor_library">...</div>
<script>...</script>
<script src="https://informatics.ischool.illinois.edu/wp-includes/js/jquery/ui/core.min.js?ver=1.13.3" id="jquery-ui-core-js"></script>
<script src="https://informatics.ischool.illinois.edu/wp-includes/js/j
```

_images/info102_onetag.png)

p4-3: Choose the subgoals that get the text from the tag that has the name of the class 'entry-title'. You do not need to use all the blocks.

```
# Get first tag of a certain type from the soup
tag = soup.find('h1', class_='entry-title')
---

# Get first tag of a certain type from the soup
tag = soup.find('h1') #paired
---

# Get info from tag
info = tag.text
---

# Get info from tag
info = tag.get('main') #distractor
---

# Get all tags of a certain type from the soup
tags = soup.find_all('div')#distractor
```



p4-4: What is the code to get a link from a tag?

```
# Get first tag of a certain type from the soup  
tags = soup.find('a', class_='announcement')  
  
# Get info from tag  
info = tag. 
```

**Note**

[Click here](#) to go back to the Faculty Pages example (/ns/books/published/cs102web/example2.html)

You have attempted 0 activities on this page

© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (<http://runestoneinteractive.org/>) 5.3.0.

| Back to

```
{{ if request.application == 'runestone':}} {{ pass }}
```



(plan2.html)



(activity-writing-1.html)

```
{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}
{{ if settings.num_banners > 0 and settings.show_rs_banner: }}
```

Please Support Runestone

(/runestone/default/donate?banner={{=banner_num}})

{{ pass }}



Code writing activity part 1

On this page, you will complete an activity to write code that:

Scrapes all the comments on the Rate My Professor page for Prof. Geoffrey Herman and Prof. Abdussalam Alawini and prints them

Here is the link to Prof. Herman's Rate My Professor page (<https://www.ratemyprofessors.com/professor/2442487>).

Here is the link to Prof. Alawini's Rate My Professor page (<https://www.ratemyprofessors.com/professor/1799030>).

You can see that both the pages have the same layout.

Advertisement

Canva

Try today

4.5 / 5

Overall Quality Based on 36 ratings

Geoffrey Herman

Professor in the Electrical & Comp. Engineering department at University Of Illinois at Urbana - Champaign

93% | 3.4

Would take again | Level of Difficulty

Rate → Compare

I'm Professor Herman

Rating Distribution

Rating	Count
Awesome 5	25
Great 4	9
Good 3	0
OK 2	0
Awful 1	2

Similar Professors

Professor Name	Rating
Raluca Ilie	5.00
Simeon Bogdanov	5.00
Chandrasekhar Radhakrishnan	4.70

Advertisement

Ulta Beauty 71K Followers

babe get in the car it's 21 DAYS OF BEAUTY! Visit Ulta Beauty and grab your favorites at 50% off during Beauty's Biggest Event!!! Shop now through March 27!...

6.2K Likes 68 Comments 76 Shares

Shop Now

Advertisement

Save 30% off Canva Pro when you pay yearly

Get Offer

Search

Social Post Poster

Reese

Moving is a snap with the Xfinity app

xfinity

R M P Professors Professor name Your school Log In Sign Up Help

Advertisement

Moving is a snap with the Xfinity app

xfinity

4.2 / 5

Overall Quality Based on 88 ratings

Abdussalam Alawini

Professor in the Computer Science department at University Of Illinois at Urbana - Champaign

75% | 3.4

Would take again | Level of Difficulty

Rate → Compare

Rating Distribution

Rating	Count
Awesome 5	64
Great 4	4
Good 3	5
OK 2	7
Awful 1	8

Similar Professors

Advertisement

Children's TYLENOL

fever Pain & Fever Fever 2-12 Years

Free On Demand Delivery

watch cheddar

The comments all have the same tag name, which is 'div' tag with class='Comments__StyledComments-dzzvym-0 dvnRbr'. Here's what it looks like when you inspect Prof. Herman's page:

Geoffrey Herman

Electrical & Comp. Engineering
University Of Illinois at Urbana - Champaign

Rate Compare

QUALITY	5.0	CS233	May 19th, 2023
DIFFICULTY	3.0	For Credit: Yes Attendance: Not Mandatory Would Take Again: Yes Grade: A+ Textbook: N/A	
		The course has many moving parts, but Prof Herman keeps everything clear and on track. He has structured the course exceptionally well, building simple concept upon simple concept until you look back and realise how far you've come. There are few professors as caring and accessible as he is, and I would gladly take any course he decides to run.	
		EXTRA CREDIT AMAZING LECTURES RESPECTED	
		Helpful 0 0	

The 'div' tag with class='Comments__StyledComments-dzzvym-0 dvnRbr' is only use to hold student comments

5.0

CS233 May 2nd, 2023

For Credit: Yes Attendance: Not Mandatory Would Take Again: Yes

Grade: A Textbook: N/A

He cares a lot for students! Every exam has 2nd chance. There are plenty of extra credits, and final weighs only 10%. The course is interesting (cuz I am a great fan of computer architecture, but anyway, you will learn A LOT), and the concepts are generally not too difficult. You should spend more time learning stash, cache, and conscious coding.

The 'div' tag with class='Comments__StyledComments-dzzvym-0 dvnRbr' is only use to hold student comments

writing1.html

Plan #2: Get a soup from a URL (/ns/books/published/cs102web/plan1.html)

Plan #1: Get a soup from multiple URLs (/ns/books/published/cs102web/plan2.html)

Plan #3: Get info from all tags of a certain type (/ns/books/published/cs102web/plan3.html)

Plan #4: Get info from a single tag (/ns/books/published/cs102web/plan4.html)

Plan #5: Print info (/ns/books/published/cs102web/plan5.html)

Links to plans

- Plan 1: Get a soup from a URL (/ns/books/published/cs102web/plan1.html)
- Plan 2: Get a soup from multiple URLs (/ns/books/published/cs102web/plan2.html)
- Plan 3: Get info from all tags of a certain type (/ns/books/published/cs102web/plan3.html)
- Plan 4: Get info from a single tag (/ns/books/published/cs102web/plan4.html)
- Plan 5: Print info (/ns/books/published/cs102web/plan5.html)

You have attempted 0 activities on this page

© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (<http://runestoneinteractive.org/>) 5.3.0.

| Back to



```
{{ if request.application == 'runestone':}} {{ pass }}
```



(plan4.html)



(activity-writing-2.html)

```
{% if response.serve_ad and settings.adsenseid: %} {{ pass }}  
{% if settings.num_banners > 0 and settings.show_rs_banner: %}
```

Please Support Runestone
(/runestone/default/donate?banner={{=banner_num}})
{ pass }



Code writing activity part 2

On this page, you will complete a second activity to write code that:

Scrapes all the comments on the Rate My Professor page for Prof. Geoffrey Herman and Prof. Abdu Alawini and prints them

Here is the link to Prof. Herman's Rate My Professor page (<https://www.ratemyprofessors.com/professor/2442487>).

Here is the link to Prof. Alawini's Rate My Professor page (<https://www.ratemyprofessors.com/professor/1799030>).

You can see that both the pages have the same layout.

The screenshot shows a professor profile for Geoffrey Herman. At the top, there's a large '4.5 / 5' rating with a 'Try today' button for Canva. Below it, the overall quality rating is based on 36 ratings. The professor's name is Geoffrey Herman, accompanied by a graduation cap icon. He is listed as a Professor in Electrical & Comp. Engineering at the University Of Illinois at Urbana - Champaign. His teaching rating is 93%, and his level of difficulty is 3.4. He has a 'Would take again' button and 'Rate →' and 'Compare' buttons. A testimonial from a student is visible below. To the right, there's a 'Rating Distribution' chart showing the percentage of students giving different ratings: Awesome (5), Great (4), Good (3), OK (2), and Awful (1). Below that is a section for 'Similar Professors' with links to their profiles. On the far right, there are three advertisement banners for Canva, Ulta Beauty, and a general offer for Canva Pro.

Advertisement  

Moving is a snap with the Xfinity app

xfinity

4.2 / 5

Overall Quality Based on 88 ratings

Abdussalam Alawini 

(activity-writing-1.html)

Professor in the Computer Science department at University Of Illinois at Urbana - Champaign

75%  3.4

Would take again Level of Difficulty

Rating Distribution

Rating	Count
Awesome 5	64
Great 4	4
Good 3	5
OK 2	7
Awful 1	8

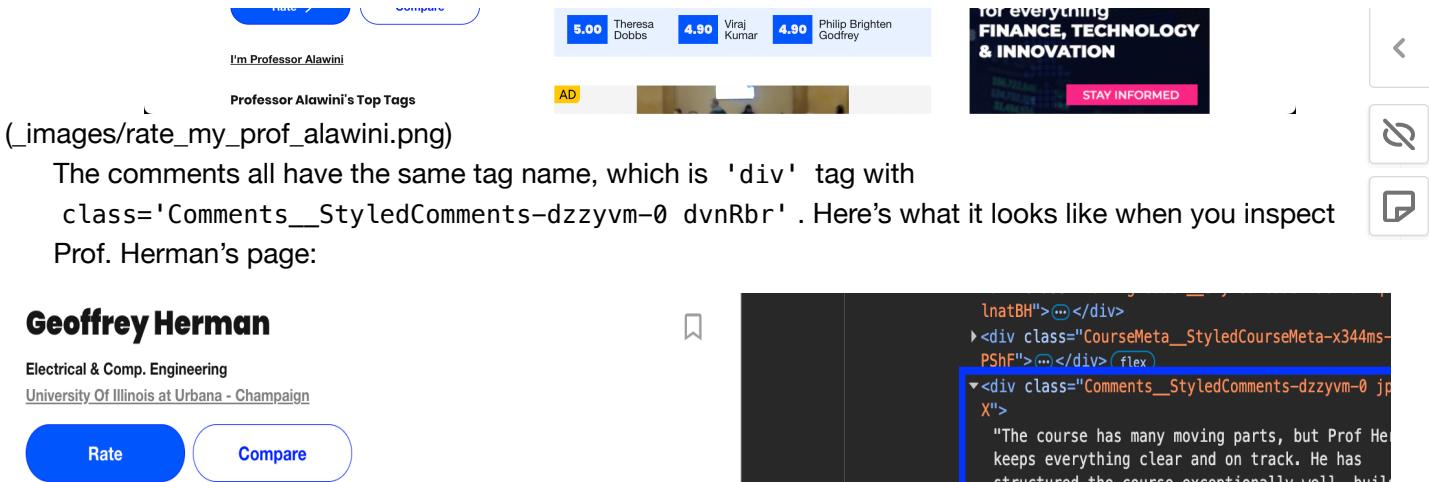
Similar Professors

Advertisement  

(activity-writing-3.html)

Advertisement  

watch cheddar



The comments all have the same tag name, which is 'div' tag with class='Comments__StyledComments-dzzyvm-0 dvnRbr'. Here's what it looks like when you inspect Prof. Herman's page:

Geoffrey Herman

Electrical & Comp. Engineering
University Of Illinois at Urbana - Champaign

Rate Compare

QUALITY
5.0

CS233 May 19th, 2023

For Credit: Yes Attendance: Not Mandatory Would Take Again: Yes

Grade: A+ Textbook: N/A

DIFFICULTY
3.0

The course has many moving parts, but Prof Herman keeps everything clear and on track. He has structured the course exceptionally well, building simple concept upon simple concept until you look back and realise how far you've come. There are few professors as caring and accessible as he is, and I would gladly take any course he decides to run.

EXTRA CREDIT AMAZING LECTURES RESPECTED

Helpful 0 0

The 'div' tag with class='Comments__StyledComments-dzzym-0 dvnRbr' is only use to hold student comments

QUALITY
5.0

CS233 May 2nd, 2023

For Credit: Yes Attendance: Not Mandatory Would Take Again: Yes

Grade: A Textbook: N/A

DIFFICULTY
3.0

He cares a lot for students! Every exam has 2nd chance. There are plenty of extra credits, and final weighs only 10%. The course is interesting (cuz I am a great fan of computer architecture, but anyway, you will learn A LOT), and the concepts are generally not too difficult. You should spend more time learning stash, cache, and conscious coding.

`<lnatBH>...</div><div class="CourseMeta__StyledCourseMeta-x344ms-PShF">...</div><div class="Comments__StyledComments-dzzym-0 jX">"The course has many moving parts, but Prof Herman keeps everything clear and on track. He has structured the course exceptionally well, building simple concept upon simple concept until you look back and realise how far you've come. There are few professors as caring and accessible as he is, and I would gladly take any course he decides to run.</div><div class="RatingTags__StyledTags-sc-1boeqx2-0 l">...</div><div class="RatingFooter__StyledRatingFooter-civ-0 hPfCXn">...</div></div></div><div class="Rating__StyledRating-sc-1rhvpxz-1 j0ZHgV"><div class="Rating__RatingBody-sc-1rhvpxz-0 hRxBtU">...</div><div class="RatingHeader__StyledHeader-sc-1dlkqw1-0 PzwL">...</div><div class="RatingValues__StyledRatingValues-sc-6-0 PzwL">...</div><div class="Rating__RatingInfo-sc-1rhvpxz-3 kfdAA-0 PzwL">...</div></div><div class="RatingHeader__StyledHeader-sc-1dlkqw1-0 lnatBH">...</div><div class="CourseMeta__StyledCourseMeta-x344ms-PShF">...</div><div class="Comments__StyledComments-dzzym-0 jX">"He cares a lot for students! Every exam has 2 chance. There are plenty of extra credits, and final weighs only 10%. The course is interesting (cuz I am a great fan of computer architecture, but anyway, you will learn A LOT), and the concepts are generally not too difficult. You should spend more time learning stash, cache, and conscious codi</div><div class="RatingTags__StyledTags-sc-1boeqx2-0 l">...</div>`

(images/rate my prof herman tags.png)

Links to plans

Plan 1: Get a soup from a URL (/ns/books/published/cs102web/plan1.html)

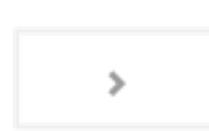
Plan 2: Get a soup from multiple URLs (/ns/books/published/cs102web/plan2.html)

Plan 3: Get info from all tags of a certain type (/ns/books/published/cs102web/plan3.htm)

Plan 4: Get info from a single tag (/ns/books/published/cs102web/plan4.html)

Plan 5: Print info (/ns/books/published/cs102web/plan5.html)

```
writing2-1: Choose which of the following plans you will use, and put them in <  
# Load libraries for web scraping  
from bs4 import BeautifulSoup  
import requests  
# Get a soup from multiple URLs  
base_url = _____  
endings = _____  
for ending in endings:  
    url = base_url + ending  
    r = requests.get(url)  
    soup = BeautifulSoup(r.content, 'html.parser')  
---  
# Load libraries for web scraping  
from bs4 import BeautifulSoup  
import requests  
# Get a soup from a URL  
url = _____  
r = requests.get(url)  
soup = BeautifulSoup(r.content, 'html.parser')#paired  
---  
# Get all tags of a certain type from the soup  
tags = soup.find_all(_____  
# Collect info from the tags  
collect_info = []  
for tag in tags:  
    info = tag._____  
    collect_info.append(info)  
---  
# Get first tag of a certain type from the soup  
tag = soup.find(_____  
# Get info from the tag  
info = tag._____#paired  
---  
# Get first tag of a certain type from the soup  
first_tag = soup.find(_____  
# Get all tags of a certain type from the first tag  
tags = first_tag.find_all(_____  
# Collect info from the tags  
collect_info = []  
for tag in tags:  
    info = tag._____  
    collect_info.append(info)#paired  
---  
# Print the info  
print(_____  
---  
# Load library for json files  
import json  
# Put info into file  
f = open(_____, 'w')  
json.dump(_____, f)
```



(activity-writing-3.html)

```
f.close()#paired
```



You have attempted of activities on this page



© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (<http://runestoneinteractive.org/>) 5.3.0.

| Back to

```
{{ if request.application == 'runestone':}} {{ pass }}
```



(activity-writing-1.html)



(activity-writing-3.html)

```
{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}
{{ if settings.num_banners > 0 and settings.show_rs_banner: }}
```

Please Support Runestone

(/runestone/default/donate?banner={{=banner_num}})

{{ pass }}



Code writing activity part 3

On this page, you will complete the final activity to write code that:

Scrapes all the comments on the Rate My Professor page for Prof. Geoffrey Herman and Prof. Abdur Alawini and prints them

Here is the link to Prof. Alawini's Rate My Professor page (<https://web.archive.org/web/20250331181529/> <https://www.ratemyprofessors.com/professor/1799030>).

Here is the link to Prof. Herman's Rate My Professor page (<https://web.archive.org/web/20250331181529/> <https://www.ratemyprofessors.com/professor/2442487>).

You can see that both the pages have the same layout.

Advertisement

Canva

Try today

4.5 / 5

Overall Quality Based on 36 ratings

Geoffrey Herman

Professor in the Electrical & Comp. Engineering department at University Of Illinois at Urbana - Champaign

93% Would take again | 3.4 Level of Difficulty

Rate → Compare

I'm Professor Herman

Rating Distribution

Rating	Count
Awesome 5	25
Great 4	9
Good 3	0
OK 2	0
Awful 1	2

Similar Professors

Professor Name	Rating
Raluca Ilie	5.00
Simeon Bogdanov	5.00
Chandrasekhar Radhakrishnan	4.70

Advertisement

Ulta Beauty 71K Followers

babe get in the car it's 21 DAYS OF BEAUTY! Visit Ulta Beauty and grab your favorites at 50% off during Beauty's Biggest Event!!! Shop now through March 27!...

6.2K Likes 68 Comments 76 Shares

Shop Now

Advertisement

Save 30% off Canva Pro when you pay yearly

Get Offer

Search

Social Post Poster

Reese

Moving is a snap with the Xfinity app

xfinity

R M P Professors Professor name Your school Log In Sign Up Help

Advertisement

Moving is a snap with the Xfinity app

xfinity

4.2 / 5

Overall Quality Based on 88 ratings

Abdussalam Alawini

Professor in the Computer Science department at University Of Illinois at Urbana - Champaign

75% Would take again | 3.4 Level of Difficulty

Rate → Compare

Rating Distribution

Rating	Count
Awesome 5	64
Great 4	4
Good 3	5
OK 2	7
Awful 1	8

Similar Professors

Advertisement

Children's TYLENOL

TYLENOL Children's Pain Fever Reliever 2-12 Years

Free On Demand Delivery

watch cheddar

The comments all have the same tag name, which is 'div' tag with class='Comments__StyledComments-dzzvym-0'. Here's what it looks like when you inspect Prof. Herman's page:

Geoffrey Herman

Electrical & Comp. Engineering
University Of Illinois at Urbana - Champaign

Rate Compare

QUALITY **5.0** **CS233** May 19th, 2023

For Credit: Yes Attendance: Not Mandatory Would Take Again: Yes

Grade: A+ Textbook: N/A

DIFFICULTY **3.0**

The course has many moving parts, but Prof Herman keeps everything clear and on track. He has structured the course exceptionally well, building simple concept upon simple concept until you look back and realise how far you've come. There are few professors as caring and accessible as he is, and I would gladly take any course he decides to run.

EXTRA CREDIT AMAZING LECTURES RESPECTED

Helpful 0 0

The 'div' tag with class='Comments__StyledComments-dzzvym-0 dvnRbr' is only use to hold student comments

QUALITY **5.0** **CS233** May 2nd, 2023

For Credit: Yes Attendance: Not Mandatory Would Take Again: Yes

Grade: A Textbook: N/A

DIFFICULTY **3.0**

He cares a lot for students! Every exam has 2nd chance. There are plenty of extra credits, and final weighs only 10%. The course is interesting (cuz I am a great fan of computer architecture, but anyway, you will learn A LOT), and the concepts are generally not too difficult. You should spend more time learning stash, cache, and conscious coding.



Here is the code that you assembled from the plans. Now that you've assembled the correct plans, fill in the blanks to complete the code.

Links to plans

Plan 1: Get a soup from a URL (/ns/books/published/cs102web/plan1.html)

Plan 2: Get a soup from multiple URLs (/ns/books/published/cs102web/plan2.html)

Plan 3: Get info from all tags of a certain type (/ns/books/published/cs102web/plan3.htm

Plan 4: Get info from a single tag (/ns/books/published/cs102web/plan4.html)

Plan 5: Print info (/ns/books/published/cs102web/plan5.html)

(activity-debugging.html)

```
#Get the webpage  
# Load libraries for web
```

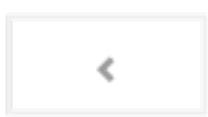


You have attempted of activities on this page

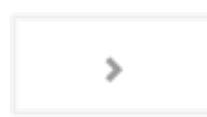
© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (<http://runestoneinteractive.org/>) 5.3.0.

| Back to

```
{{ if request.application == 'runestone':}} {{ pass }}
```



(activity-writing-2.html)



(activity-debugging.html)

```
{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}
{{ if settings.num_banners > 0 and settings.show_rs_banner: }}
```

Please Support Runestone

(/runestone/default/donate?banner={{=banner_num}})

{{ pass }}



Code debugging activity

Moore's Rescue Ranch in Champaign has pets for adoption (<https://mooresrescueranch.org/product/frodo/>). The code below is supposed to **get the webpage of adoptable cat Frodo, scrape the text of the description excerpt shown, and print it.**

Moore's Rescue Ranch

[Home](#) / [Adoption](#) / [Cat](#) / Frodo



(_images/frodo_adoption.png)

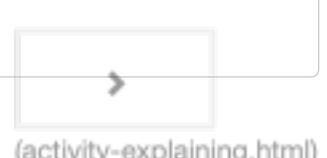
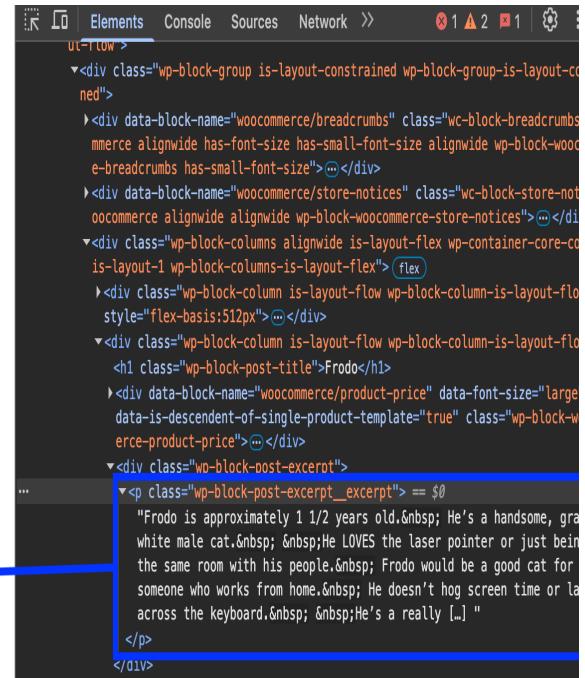
However, it doesn't work! Instead of printing the title text, it prints nothing.

Can you fix it? Here is the buggy code:

Plan 1: Get a soup from a URL

```
# Load libraries for web scraping
from bs4 import BeautifulSoup
import requests
# Get a soup from a URL
url = 'https://mooresrescueranch.org/product/frodo/'
r = requests.get(url)
soup = BeautifulSoup(r.content, 'html.parser')
```

(/ns/books/published/cs102web/plan1.html)
(activity-writing-3.html)



Plan 4: Get info from a single tag

```
# Get first tag of a certain type from the soup  
tag = soup.find('p', class_='wp-block-post-excerpt__excerpt')  
# Get info from tag  
info = tag.get('href')
```

(/ns/books/published/cs102web/plan4.html)

**Plan 5: Print the info**

```
# Print the info  
print(info)
```

(/ns/books/published/cs102web/plan5.html)

Try to fix the buggy code below. Run the code to save your progress.

```
#Get the webpage  
# Load libraries for web
```

Links to plans

Plan 1: Get a soup from a URL (/ns/books/published/cs102web/plan1.html)

Plan 2: Get a soup from multiple URLs (/ns/books/published/cs102web/plan2.html)

Plan 3: Get info from all tags of a certain type (/ns/books/published/cs102web/plan3.htm

Plan 4: Get info from a single tag (/ns/books/published/cs102web/plan4.html)

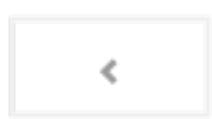
Plan 5: Print info (/ns/books/published/cs102web/plan5.html)

You have attempted 0 activities on this page

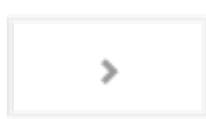
© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (<http://runestoneinteractive.org/>) 5.3.0.

| Back to

{{ if request.application == 'runestone':}} {{ pass }}



(activity-writing-3.html)



(activity-explaining.html)

```
{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}
{{ if settings.num_banners > 0 and settings.show_rs_banner: }}
```

Please Support Runestone

(/runestone/default/donate?banner={{=banner_num}})

{{ pass }}



Code explaining activity

Look at the code below, and try to determine what it does. Describe the code in your own words.

Relevant tags

Here's the relevant tag from <https://ischool.illinois.edu/people/judith-pintar>:

The screenshot shows a web browser with the 'Elements' developer tool open. The page content includes a sidebar with 'IN THE NEWS' and several news items. Two specific news items are highlighted with blue boxes and connected by arrows to the corresponding HTML code in the developer tool's element tree:

- News item 1:** "iSchool instructors ranked as excellent" (September 9, 2024). The highlighted code in the developer tool is:


```
<a class="news-row_link" href="/news-events/news/2024/09/ischool-instructors-ranked-as-excellent">iSchool instructors ranked as excellent</a>
      <time class="news-row_publish-date" datetime="2024-09-09">...</time>
```
- News item 2:** "New project to prepare global and area studies units for emerging technologies" (August 20, 2024). The highlighted code in the developer tool is:


```
<a class="news-row_link" href="/news-events/news/2024/08/new-project-prepare-global-and-area-studies-units-emerging-technologies">New project to prepare global and area
      studies units for emerging technologies</a>
      <time class="news-row_publish-date" datetime="2024-08-20">...</time>
```

_images/news_pintar_multiple.png

```
# Load libraries for web scraping
from bs4 import BeautifulSoup
import requests

# Get a soup from a URL
url = 'https://www.info.univ.edu/people/jane-doe'
r = requests.get(url)
soup = BeautifulSoup(r.content, 'html.parser')
```

(/ns/books/published/cs102web/plan1.html)



(activity-debugging.html)



(end_here_comparison.html)

```
# Get all tags of a certain type from the soup
tags = soup.find_all('a', class_='item-teaser--more')
# Collect info from the tags
collect_info = []
for tag in tags:
    # Get info from tag
    info = tag.get('href')
    collect_info.append(info)
```

(/ns/books/published/cs102web/plan3.html)

```
# Get a soup from multiple URLs
base_url = 'https://www.info.univ.edu/'
endings = collect_info
for ending in endings:
    url = base_url + ending
    r = requests.get(url)
    soup = BeautifulSoup(r.content, 'html.parser')
```

(/ns/books/published/cs102web/plan2.html)

```
# Get all tags of a certain type from the soup
tags = soup.find_all('p')
# Collect info from the tags
collect_info = []
for tag in tags:
    # Get info from tag
    info = tag.text
    collect_info.append(info)
```

(/ns/books/published/cs102web/plan3.html)

```
# Print the info
print(collect_info)
```

(/ns/books/published/cs102web/plan5.html)

explaining-1: Write down your best guess of what the code does.

You can
run the
code
below and
see what
(activity_explaining.html)
happens.

Links to plans

Plan 1: Get a soup from a URL (/ns/books/published/cs102web/plan1.html)

Plan 2: Get a soup from multiple URLs (/ns/books/published/cs102web/plan2.html)

Plan 3: Get info from all tags of a certain type (/ns/books/published/cs102web/plan3.htm

Plan 4: Get info from a single tag (/ns/books/published/cs102web/plan4.html)

Plan 5: Print info (/ns/books/published/cs102web/plan5.html)

```
#Get the webpage  
# Load libraries for web
```

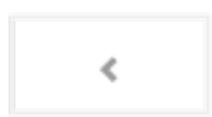


You have attempted of activities on this page

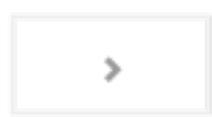
© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (<http://runestoneinteractive.org/>) 5.3.0.

| Back to

```
{{ if request.application == 'runestone':}} {{ pass }}
```



(activity-debugging.html)



(end_here_comparison.html)

```
{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}  
{{ if settings.num_banners > 0 and settings.show_rs_banner: }} {{ pass }}
```

Please Support Runestone

(/runestone/default/donate?banner={{=banner_num}})
{{ pass }}



Before You Complete The Activity - 1

Please rate how much you agree with the each statement. Your responses will be anonymized.

I am interested in learning how to extract data from websites using the web scraping tools such as Beautiful Soup.

- 1. Strongly Disagree
- 2. Disagree
- 3. Somewhat Disagree
- 4. Neither Agree nor Disagree
- 5. Somewhat Agree
- 6. Agree
- 7. Strongly Agree

I am confident in my ability to use web scraping tools such as Beautiful Soup to scrape or parse web pages for data.

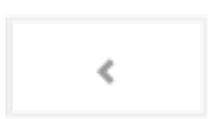
- 1. Strongly Disagree
- 2. Disagree
- 3. Somewhat Disagree
- 4. Neither Agree nor Disagree
- 5. Somewhat Agree
- 6. Agree
- 7. Strongly Agree

You have attempted of activities on this page

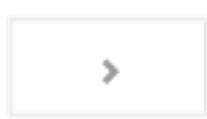
© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (<http://runestoneinteractive.org/>) 5.3.0.

| Back to

```
{{ if request.application == 'runestone':}} {{ pass }}
```



(activity-explaining.html)



(end_here_usability.html)

```
{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}  
{{ if settings.num_banners > 0 and settings.show_rs_banner: }} {{ pass }}
```

Please Support Runestone

(/runestone/default/donate?banner={{=banner_num}})

```
{{ pass }}
```



Before You Complete The Activity - 2

Rate how much you agree with the given statement based on your experience with this interface.

Overall, I am satisfied with how easy it is to use this system.

- 1. Strongly Disagree
- 2. Disagree
- 3. Somewhat Disagree
- 4. Neither Agree nor Disagree
- 5. Somewhat Agree
- 6. Agree
- 7. Strongly Agree

I felt comfortable using this system.

- 1. Strongly Disagree
- 2. Disagree
- 3. Somewhat Disagree
- 4. Neither Agree nor Disagree
- 5. Somewhat Agree
- 6. Agree
- 7. Strongly Agree

I believe I learned this material effectively using this system.

- 1. Strongly Disagree
- 2. Disagree
- 3. Somewhat Disagree
- 4. Neither Agree nor Disagree
- 5. Somewhat Agree
- 6. Agree
- 7. Strongly Agree

The information provided with the system was effective in helping me complete the tasks & scenarios.

- 1. Strongly Disagree
- 2. Disagree
- 3. Somewhat Disagree
- 4. Neither Agree nor Disagree
- 5. Somewhat Agree

- 6. Agree
- 7. Strongly Agree



I liked using the interface of this system.

- 1. Strongly Disagree
- 2. Disagree
- 3. Somewhat Disagree
- 4. Neither Agree nor Disagree
- 5. Somewhat Agree
- 6. Agree
- 7. Strongly Agree

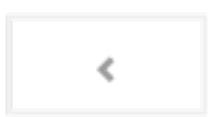
writing3-6: If you have any additional feedback, please share it below (optional).

You have attempted of activities on this page

© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (<http://runestoneinteractive.org/>) 5.3.0.

| Back to

`{{ if request.application == 'runestone':}} {{ pass }}`



(end_here_comparison.html)