



Midterm
Review

Sorting Dictionary

Nested Dictionaries

Files

CSV

Regular Expression

HTML

Beautiful Soup

Practice

Problem

Discussion 9: Midterm Review, HTML and Beautiful Soup

SI 206: Data-Oriented Programming

Instructor: Dr. Barbara (Barb) Ericson

GSI: Kexuan (Michael) Huang

IA: Cristina & Jade

School of Information
University of Michigan

Fall 2023

Midterm
Review

Sorting Dictionary

Nested Dictionaries

Files

CSV

Regular Expression

HTML

Beautiful Soup

Practice
Problem

Reminders

- Midterm 2 during lecture time (Wed/Thur) in the lecture room
- Can bring one double-side **cheat sheet** with notes
- **No** outside resource (Google, ChatGPT, Regex101, Runestone...)

Deadlines

- Nothing due this week
- Project 2 due next Friday (11/10)

Table of Contents

Midterm
Review

Sorting Dictionary

Nested Dictionaries

Files

CSV

Regular Expression

HTML

Beautiful Soup

Practice
Problem

① Midterm Review

② Practice Problem

Table of Contents

Midterm Review

Sorting Dictionary

Nested Dictionaries

Files

CSV

Regular Expression

HTML

Beautiful Soup

Practice Problem

① Midterm Review

② Practice Problem

Sorting Dictionary

Midterm
Review

Sorting Dictionary

Nested Dictionaries

Files

CSV

Regular Expression

HTML

Beautiful Soup

Practice
Problem

Sorting Dictionary

```
1 my_dict = {'a': 2, 'c': 1, 'b': 3}
2
3 # Sort by keys: returns [('a', 2), ('b', 3), ('c', 1)]
4 sorted(my_dict.items())
5
6 # Sort by values: returns [('c', 1), ('a', 2), ('b', 3)]
7 sorted(my_dict.items(), key=lambda x: x[1])
8
9 # Sort by keys in reverse order: returns [('c', 1), ('b', 3), ('a', 2)]
10 sorted(my_dict.items(), reverse=True)
11
12 # Sort by value in reverse order: returns [('b', 3), ('a', 2), ('c', 1)]
13 sorted(my_dict.items(), key=lambda x: x[1], reverse=True)
```

Nested Dictionaries

- Access an inner value of a nested dictionary with sequential square brackets
- Loop through a nested dictionary with nested `for` loops

Nested Dictionaries

```
1 nested_dict = {'layer_1': {'layer_2': 'hello'}}  
2  
3 # Access an inner value of a nested dictionary  
4 nested_dict['layer_1']['layer_2'] # returns 'hello'  
5  
6 # Loop through a nested dictionary  
7 for layer_1_key, layer_1_value in nested_dict.items():  
8     for layer_2_key, layer_2_value in layer_1_value.items():  
9         # do something here  
10        print(layer_2_value) # prints 'hello'
```

Files

Midterm
Review

Sorting Dictionary

Nested Dictionaries

Files

CSV

Regular Expression

HTML

Beautiful Soup

Practice
Problem

- Use `open()` and `close()`, or use `with` keyword to automatically close the file
- Use `readlines()` to get a list that contains each line of file as items

Work with Files

```
1 # Use open() and close()
2 f = open('file.txt', 'r')
3 content = f.read()           # returns the content of the file as a string
4 f.close()

5

6 # Use 'with' keyword
7 with open('file.txt', 'r') as f:
8     lines = f.readlines() # gives us a list of lines in the file
9     for line in lines:
10         print(line)       # print one line
```



CSV Reader

Midterm
Review

Sorting Dictionary

Nested Dictionaries

Files

csv

Regular Expression

HTML

Beautiful Soup

Practice

Problem

Read CSV with CSV Reader.

Example

```
1 import csv
2
3 with open('demo.csv') as csv_file:
4     csv_reader = csv.reader(csv_file) # read the file with csv reader
5     header = next(csv_reader)        # get the first line as a list
6     for row in csv_reader:          # iterate through the rows
7         # here row is a list of data in one row
```

CSV Writer

Write CSV with CSV Writer.

Example

```
1 import csv
2
3 data = [['firstName', 'lastName'], ['Matty', 'Tran']]
4
5 with open('output.csv', 'w') as csv_file:
6     # write the file with csv writer
7     csv_writer = csv.writer(csv_file, delimiter=',')
8
9     # iterate through the rows
10    for row in data:
11        # here row is the inner list
12        csv_writer.writerow(row)
```

Regular Expression

Midterm
Review

Sorting Dictionary

Nested Dictionaries

Files

CSV

Regular Expression

HTML

Beautiful Soup

Practice
Problem

- \w: alphabet, number and underscore
- \s: any whitespace character
- \b: boundary between alphanumeric characters and whitespace
- ^: start of a string
- \$: end of a string
- [...]: match a set of characters
- [^...]: any character not in the brackets
- .: any character
- *: repeat 0 or more times
- +: repeat 1 or more times
- {m, n}: repeat m to n times (included)
- (...): returns only the expression inside the parenthesis
- (?....): treats the characters in the parenthesis as a whole expression



Regular Expression

Midterm
Review

Sorting Dictionary

Nested Dictionaries

Files

CSV

Regular Expression

HTML

Beautiful Soup

Practice
Problem

Example

```
1 import re
2
3 my_string = '123 4567 8 910'
4 pattern = r'\d{3}'
5
6 # findall returns a list of matches
7 re.findall(pattern, my_string)      # returns ['123', '456', '910']
8
9 # search returns only the first match
10 re.search(pattern, my_string).group() # returns '123'
11
12 # match returns only the match from the beginning of string
13 re.match(pattern, myString).group()  # returns '123'
```

HTML

Midterm Review

Sorting Dictionary

Nested Dictionaries

Files

CSV

Regular Expression

HTML

Beautiful Soup

Practice Problem



What we see

A horizontal black arrow pointing to the right.

```
<!DOCTYPE html> == 50
<html dir="ltr" lang="en" class="lazy-loaded">
  <head> ...
    <body style="background-color: #53, 54, 58;">
      <!--body style="background-color: #53, 54, 58;"-->
      <!--frame id="backgroundImage" src="#"-->
      <!--ntp-app most-visited-reflow-on-overflow-enabled_remove-scrim-->
      <!--shadow-root (open)>
        <!--_html_template_start__-->
        <style> :host([hidden]),[hidden]{display:none!important} </style>
        <!--style-->
        <!--style include="cr-hidden-style cr-icons"-->
        <!--style include="cr-shared-style"-->
        <!--div id="content" style="--color-new-tab-page-attribution-foreground: rgba(232, 234, 237, 1.00); --color-new-tab-page-most-visited-foreground: rgba(232, 234, 237, 1.00); --ntp-logo-color: rgba(255, 255, 255, 1.00);"> ...
          <!--div flex-->
          <!--dom-if reststamp style="display: none;"-->
          <!--dom-if id="customizeDialogIf" reststamp style="display: none;"-->
          <!--svg-->
          <!--_html_template_end__-->
    </ntp-app>
    <script type="module" src="new_tab_page.js"></script>
    <link rel="stylesheet" href="chrome://resources/css/text_defaults_md.css">
    <link rel="stylesheet" href="chrome://theme/colors.css?sets=ui_chrome">
    <link rel="stylesheet" href="shared_vars.css">
    <script type="module" src="/lazy_load.js"></script>
  </body>
</html>
<!doctype>
```

Behind the scenes

HTML

Midterm
Review

Sorting Dictionary

Nested Dictionaries

Files

CSV

Regular Expression

HTML

Beautiful Soup

Practice
Problem

- HTML (HyperText Markup Language)
- HTML is the standard markup language for web pages
- HTML elements are delineated by **tags**

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```



This is a Heading

This is a paragraph.

HTML Tag

Midterm
Review

Sorting Dictionary

Nested Dictionaries

Files

CSV

Regular Expression

HTML

Beautiful Soup

Practice

Problem

```
<elementName attribute="value">Content</elementName>
```

Examples:

- ``
- `Visit W3Schools.com!`
- `<div class="myDiv">`
 `<h2>This is a heading in a div element</h2>`
 `<p>This is some text in a div element.</p>`
`</div>`

Requests

Midterm
Review

Sorting Dictionary

Nested Dictionaries

Files

CSV

Regular Expression

HTML

Beautiful Soup

Practice

Problem

Library for making HTTP (Hypertext Transfer Protocol) requests to interact with web services, websites, and APIs.

```
import requests

url = "https://www.google.com/"

r = requests.get(url)

print(r)

print(r.text)
```



```
<Response [200]>
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lan
, including webpages, images, videos and more. Google has many special fea
ame="description"><meta content="nooop" name="robots"><meta content="text/
nt=/Images/branding/googleleg/1x/googleleg_standard_color_128dp.png" itemprop
-w07ce20A>(function(){var _g=(kEl:'QNU2ZdmNpj52roP2M-y4As',kEXP1:0,1816
7528,16112,28687,22430,1362,12314,17585,4998,17875,38444,2872,2891,3926,21
894,29703,1457,22610,6627,7596,1,42154,2,16395,342,3533,19491,5679,1021,31
14491,873,19633,7,1922,9779,42459,20199,20136,14,82,7651,8863,3692,109,241
,,12889,1632,2173,6669,868,3785,949,3692,8565,7769,146,21746,5203198,12,692
5,2,40,7,16,6,9,8,9,23940932,4044106,14298,2374,39458,1446,1763,1216,3,210
3,4636,2945,5463,2893,787,1597,26,3525,5494,943,19,216,241,1725,3780,7351,
2496,419,1104,1484,149,1243,3,3363,2174,365,1718,2,3263,2811,9,217,3590,56
,,2116,4,498,574,1594,85,614,4,872,438,534,894,1,14,762,9,217,1328,750,7,1,
,,706,233,46,1095,35,707,828,42,28,3,2,537,406,114,127,795,206,128,2,1021,6
40,298,426,106,109,1150,78,11,1,3,4,2,2,2,545,41,539,2,497,131,773,206,429
7,477,227,29,337,86,714,1,437,5,243,1,3,27,136,300,7,108,192,4,193,238,47,
449);(function(){var a;(null==(a=window.google)?@:a.stvsc)@google.kEl=_g.k
sm="webhp";google.KHL="en";});}(function(){
var h=this||self;if(function l(){return void 0!=window.google&&void 0!=wind
null};var m={};function p(a){for(var b@@(a.getAttribute)||!(b=a.getAttribute("leid")):@=a.pare
==window.location.protocol&&!google.ml@&google.ml(Error("a"),1,{src:a,gm
for(var b=nUL;a@@(a.getAttribute||!(b=a.getAttribute("leid")):@=a.pare
==window.location.protocol&&!google.ml@&google.ml(Error("a"),1,{src:a,gm
t(a,b,c,d,k){var e="";-1==b.search("&ie=")&&e="&ie="&pid","-1==
b.search("&cshid=")&&"shd"!=a,f=[];f.push(["zx",Date.now().toString()])
.push(["opn",c.toString()]);for(c=0;<f.length;c++){if(0==c||0<c)d="&g=
"+String(a)+"&cad="+b+e+d};m@google.kEl@google.getEl@&google.getEL@q=
c,d,k,e){e==void 0==?e:c||(c=@(a,b,e,d,k);if(c==r(c)){a=new Image;var g
elete n[g];a.src=c};google.logUrl=function(a,b){b==void 0==?b:@;return
=];google.x=function(a){if(a){var c=a.id;else{do c=Math.random();while(g
(a){google.ly.push(a);google.ly=@[];google.plm=function(a){google.ln.push(
c{google.lq.push([a,b,c]);google.loadAll=function(a,b){google.lq.push(
e,fc=function(a,b,c){d.push([a,b,c,e]);google.qce=d});call(this);goog
document.documentElement.addEventListener("submit",function(b){var a;if(a=
=c)||"q"==c@a.elements.q.value?0:1){else a=1;a@@(b.preventDefault(),b,
istener("click",function(b){var a:@;for(a=b.target;a@@a==document.docume
getAttribute("data-nohref");break a})@=1;a@@b.preventDefault(),!0});}.cal
```

Beautiful Soup

Midterm Review

Sorting Dictionary

Nested Dictionaries

Files

CSV

Regular Expression

HTML

Beautiful Soup

Practice Problem

```
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en"><head><meta content="Search the world's information , including webpages, images, videos and more. Google has many special features to help you find exactly what you're looking for." name="description"><meta content="noodp" name="robots"><meta content="text/html; charset=UTF-8" http-equiv="Content-Type"><meta content="https://images.branding.google/1x/google_standard_color_128dp.png" itemprop="image"><title>Google</title><script nonce="6_A56q9g4ZJHm -w07ce20A">(function(){var g={kEI:'0NU2ZdmNpjS2roP2M-y4As',kEXPI:'0',v:'0',18167,1347301,206,4884,2316,383,246,5,1129120,1197767,303195,7 7528,16112,28687,22430,1362,12314,17585,4998,17075,38444,2872,2891,3926,213,7615,606,29842,35,30813,15324,781,1244,1,16918,2650,4,32 894,29703,1457,22610,6627,7596,1,42154,2,16395,342,3533,19491,5679,1021,31122,4568,6255,23421,1249,33067,2,2,1,24626,2006,8155,8860, 14491,873,19633,7,1922,9779,42459,20199,20136,14,82,7651,8863,3692,109,2412,5856,18960,5403,782,2248,15816,1804,7734,15103,2995,9161 ,12089,1632,2173,6669,868,3785,949,3692,8565,7769,146,21746,5203198,12,6929,2,196,5994891,678,97,2806428,936,4,29229,5,31,47,2,56,16 5,2,40,7,16,6,9,8,9,2394932,4844106,14298,2374,39458,1446,1763,1216,3,2109,3,1165,4,297,415,561,296,331,1398873,23759270,5178,2,298 3,4636,2945,5463,2093,787,1597,26,3525,5494,943,19,216,241,1725,3780,7351,6319,872,1,1,2,3,6580,3117,1535,2464,1879,3235,303,7054,1, 2490,419,1184,1484,149,1243,3,3363,2174,365,1718,2,3263,2811,9,217,3590,565,1074,430,2742,1,1,2,3,249,933,506,969,1083,27,6,5,15,311 ,2116,4,498,574,1594,85,614,4,872,438,534,894,1,14,762,9,217,1328,750,7,1,8,354,374,35,1200,1363,1,520,746,789,412,453,356,11,1,1288 ,706,233,46,1095,35,707,828,42,28,3,2,537,406,114,127,795,206,128,2,1021,668,4,447,726,6,2,1,198,148,5,145,123,41,1,673,130,56,120,2 40,298,426,106,109,1150,78,11,1,3,4,2,2,2,2,545,41,539,2,497,131,773,206,429,153,50,28,763,1071,3,463,18,28,101,240,183,450,756,39,217 7,477,227,29,337,86,714,1,437,5,243,1,3,27,136,300,7,108,192,4,193,238,47,2,876,1,216,1251,693,2,1,6,1532,408',kBL:'0Jw0',kOPI:'89978 449};(function(){var a;{null===(a>window.google)?0:a.stvsc?g:{kEI:'window.google.google'}.call(this);});}(function(){google. sn='webhp';google.kHL='en';})());}(function(){ var h=this||self;function l(){return void 0===(window.google&&void 0==!(window.google.kOPI&&!window.google.kOPI?window.google.kOPI: null);var m,n=[];function p(a){for(var b;a&&(a.getAttribute||!!(b=a.getAttribute("eid"))));a=a.parentNode;return b||[]}{function q(a){for(var b=n;l=a&&(a.getAttribute||!!(b=a.getAttribute("leid"))));a=a.parentNode;return b||[]}{function r(a){"/^http://i.test(a)&&https://" ==window.location.protocol&&(google.ml&&google.ml||(Error("a")),1,{src:a,gLmn:1});a="";return a} function t(a,b,c,d,k){var e="";-1==b.search("&ei")&&(e+="p+d"),-1==b.search("&ei=")&&(d=q(d))&&(e+="&lei="+d);d="";var g=e-1 ==b.search("&cshid=")&&"&lsh!="=a,f=[];f.push(["z"+Date.now().toString()]);h._cshid&g&f.push(["cshid",h._cshid]);c=c();null!=c&&f .push(["opi",c,toString()]);for(c=0;c<f.length;c++){if(0==c||0<c)&#39;=&#39;&d=f[c][0]&#39;+=&#39;+f[c][1];return"/"+(k)||"gen_204"+"?atyp=" +c+"+String(a)+"&cad="+(b+e+d);m=google.kEI;google.getEI=p;google.getLEI=q;google.ml=function(){return null};google.log=function(a,b, c,d,k,e){void 0==e?l:a||c||(a,b,e,d,k);if(c=r(c)){a=new Image;var g=n.length;[g]=a;onerror=a.onerror;a.onload=a.onabort=function(){d elete n[g];a.src=c};google.logUrl=function(a,b){void 0==b?l:b{return return t("",a,b)}},call(this);(function(){google.v={};google.sy ={};google.x=function(a,b){if(a){var c=a.id;else do c=Math.random();while(google.y[c])google.y[c]=a;b;return i};google.sx=function (a){google.sy.push(a);google.lm=[];google.plm=function(a){google.lm.push.apply(google.lm,a)};google.lq=[],google.load=function(a, b){google.lq.push([a,b,c]);google.loadAll=function(a,b){google.lq.push([a,b])};google.bx=1;google.lx=function(){var d=[];googl e.fce=function(a,b,c,e){d.push([a,b,c,e])};google.qce=d};(function(){document.documentElement.addEventListener("submit",function(b){var a;if(a=b.target){var c=a.getAttribute("data-submitfalse");a="1"==c||"q"==c&&a.elements.q.value?!"!":1}else a=!1;a&&b.preventDefault(),b.stopPropagation(),!0});document.documentElement.addEventListener("click",function(b){var a;if(a=b.target){a&&(!document.documentElement==a.parentNode)if("A"==a.tagName){a="1"==a.getAttribute("data-nohref");break}a!=1)a&&b.preventDefault(),!0});}).call(this);google.f={};(function(){document.documentElement.addEventListener("submit",function(b){var a;if(a=b.target){var c=a.getAttribute("data-submitfalse");a="1"==c||"q"==c&&a.elements.q.value?!"!":1}else a=!1;a&&b.preventDefault(),b.stopPropagation(),!0});document.documentElement.addEventListener("click",function(b){var a;if(a=b.target){a&&(!document.documentElement==a.parentNode)if("A"==a.tagName){a="1"==a.getAttribute("data-nohref");break}a!=1)a&&b.preventDefault(),!0});}).call(this);</script><style>#gbar,#guser{font-size:13px;paddin
```

Beautiful Soup

Midterm
Review

Sorting Dictionary

Nested Dictionaries

Files

CSV

Regular Expression

HTML

Beautiful Soup

Practice

Problem

Example

```
1 import requests
2 from bs4 import BeautifulSoup
3
4 url = "https://www.google.com/"
5 response = requests.get(url)      # get raw data with requests
6
7 if response.status_code == 200: # check if response is normal
8     html = response.text          # extract HTML text from raw data
9 else:
10    exit("Failed to retrieve the web page.")
11
12 soup = BeautifulSoup(html, 'html.parser') # parse data with beautifulsoup
```

Parsing Tags

Midterm
Review

Sorting Dictionary

Nested Dictionaries

Files

CSV

Regular Expression

HTML

Beautiful Soup

Practice
Problem

```
soup = BeautifulSoup(html,  
'html.parser')  
  
print(soup.find('a'))
```



```
<a class="gb1"  
href="https://www.google.com/imghp?hl=en&tab=wi">Images</a>
```

Return first match

```
soup = BeautifulSoup(html,  
'html.parser')  
  
print(soup.find_all('a'))
```



```
[<a class="gb1"  
href="https://www.google.com/imghp?hl=en&tab=wi">Images</a>, <a class="gb1"  
href="https://maps.google.com/maps?hl=en&tab=w1">Maps</a>, <a class="gb1"  
href="https://play.google.com/?hl=en&tab=w8">Play</a>, <a class="gb1"  
href="https://www.youtube.com/?tab=w1">YouTube</a>, ...]
```

Return list of all matches



Working with Tag Objects

Midterm
Review

Sorting Dictionary

Nested Dictionaries

Files

CSV

Regular Expression

HTML

Beautiful Soup

Practice

Problem

```
soup = BeautifulSoup(html,
'html.parser')
first_tag = soup.find('a')

print(first_tag.attrs)
print(first_tag.get('class'))
```



```
{'class': ['gb1'], 'href':
'https://www.google.com/imghp?hl=en&tab=wi'}
```

```
['gb1']
```

```
<a class="gb1"
href="https://www.google.
com/imghp?hl=en&tab=w
i">Images</a>
```



Working with Tag Objects

Midterm
Review

Sorting Dictionary

Nested Dictionaries

Files

CSV

Regular Expression

HTML

Beautiful Soup

Practice

Problem

You can pass attributes and their values as arguments to your function to run a more specific search.

Example

```
1 # Find all tags that have a 'href' attribute with Google URL as value
2 soup.find_all('a', href='https://www.google.com')
3
4 # Find all divs with the class 'myClass'
5 # 'class' is a reserved keyword in Python, so use 'class_' instead
6 soup.find_all('div', class_='myClass')
7
8 # Find all divs with multiple classes
9 soup.find_all('div', class_='myClass myClass2 myClass3')
```

Working with Tag Objects

Midterm
Review

Sorting Dictionary

Nested Dictionaries

Files

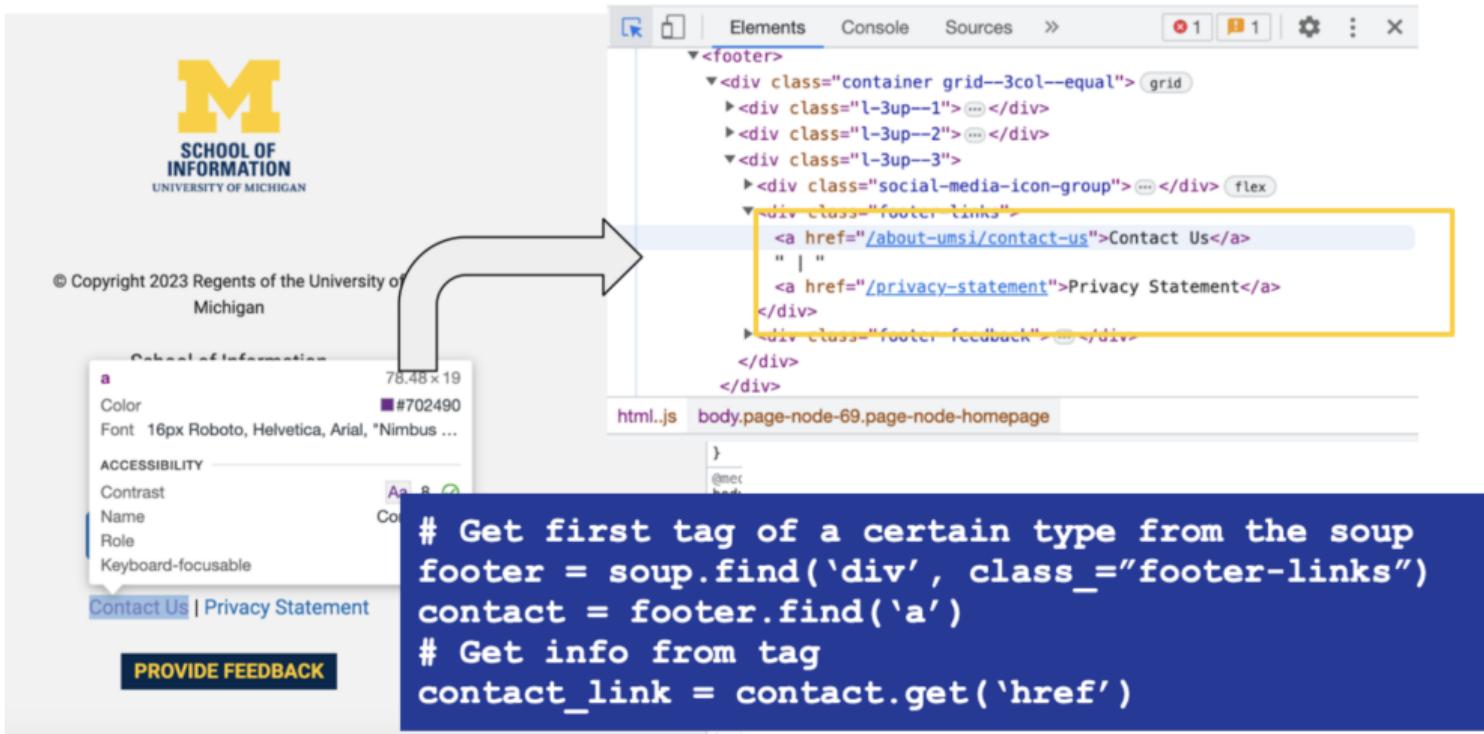
CSV

Regular Expression

HTML

Beautiful Soup

Practice
Problem



The screenshot illustrates how to extract specific links from a website's footer using BeautifulSoup. The DOM shows a `<div>` element with class `"l-3up--3"` containing two `<a>` tags. The first `<a>` tag has a href of `/about-umsi/contact-us` and the second has a href of `/privacy-statement`. A developer tools panel shows the CSS properties for the `a` tag, including color (#702490) and font (16px Roboto, Helvetica, Arial, "Nimbus ..."). The bottom right corner contains Python code demonstrating the extraction process:

```
# Get first tag of a certain type from the soup
footer = soup.find('div', class_="footer-links")
contact = footer.find('a')
# Get info from tag
contact_link = contact.get('href')
```

Working with Tag Objects

Midterm Review

Sorting Dictionary

Nested Dictionaries

Files

CSV

Regular Expression

HTML

Beautiful Soup

Practice Problem

HOME / RESEARCH / RESEARCH AREAS

Research areas

h2.research-area-teaser__title	210.87 × 45
Color	#00274C
Font	20px "Roboto Condensed", sans-serif
Margin	10px 0px 11.25px
ACCESSIBILITY	
Contrast	Aa 15.05
Name	Accessibility and Computing
Role	heading
Keyboard-focusable	



Accessibility and Computing

Accessible and inclusive technology



Archives and Digital Curation

Creation, curation, preservation and use of information resources

Collective Intelligence and Organizational Technology

Socio-technical systems to support institutions and collaboration

```
# Get all tags of a certain type from the soup
research_areas = soup.find_all("h2", class_="research-area-teaser__title")
research_areas_list = []
for tag in research_areas:
    # Get info from tag
    research_areas_list.append(tag.text)
```

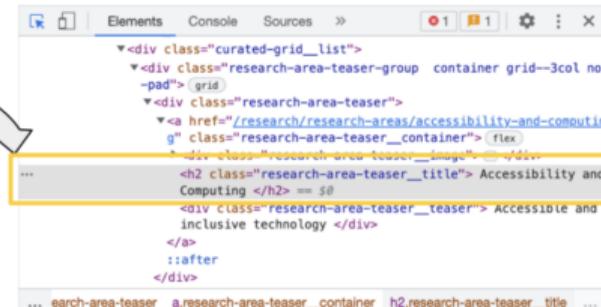


Table of Contents

Midterm
Review

Sorting Dictionary

Nested Dictionaries

Files

CSV

Regular Expression

HTML

Beautiful Soup

Practice
Problem

① Midterm Review

② Practice Problem



Practice Problems

Midterm
Review

Sorting Dictionary

Nested Dictionaries

Files

CSV

Regular Expression

HTML

Beautiful Soup

Practice

Problem

Go to Canvas → Assignment → Discussion 9 and clone the GitHub Repo.

Your tasks

- Implement `get_emblem(soup)`: returns the link (source) for the University of Michigan emblem
- Implement `get_school_rankings(soup)`: get the details of the table named "USNWR Global Program Rankings", and returns a Dictionary with program name as key and ranking as value.
- In `main()` function: create a BeautifulSoup object, call your functions with the soup object and sort the dictionary by rank to see what the 3 highest ranked programs are at the University of Michigan

Task 1: Get Emblem

Midterm Review

Sorting Dictionary

Nested Dictionaries

Files

CSV

Boundary Expansion

14783

Practice

Midterm
Review

Sorting Dictionary

Nested Dictionaries

Files

CSV

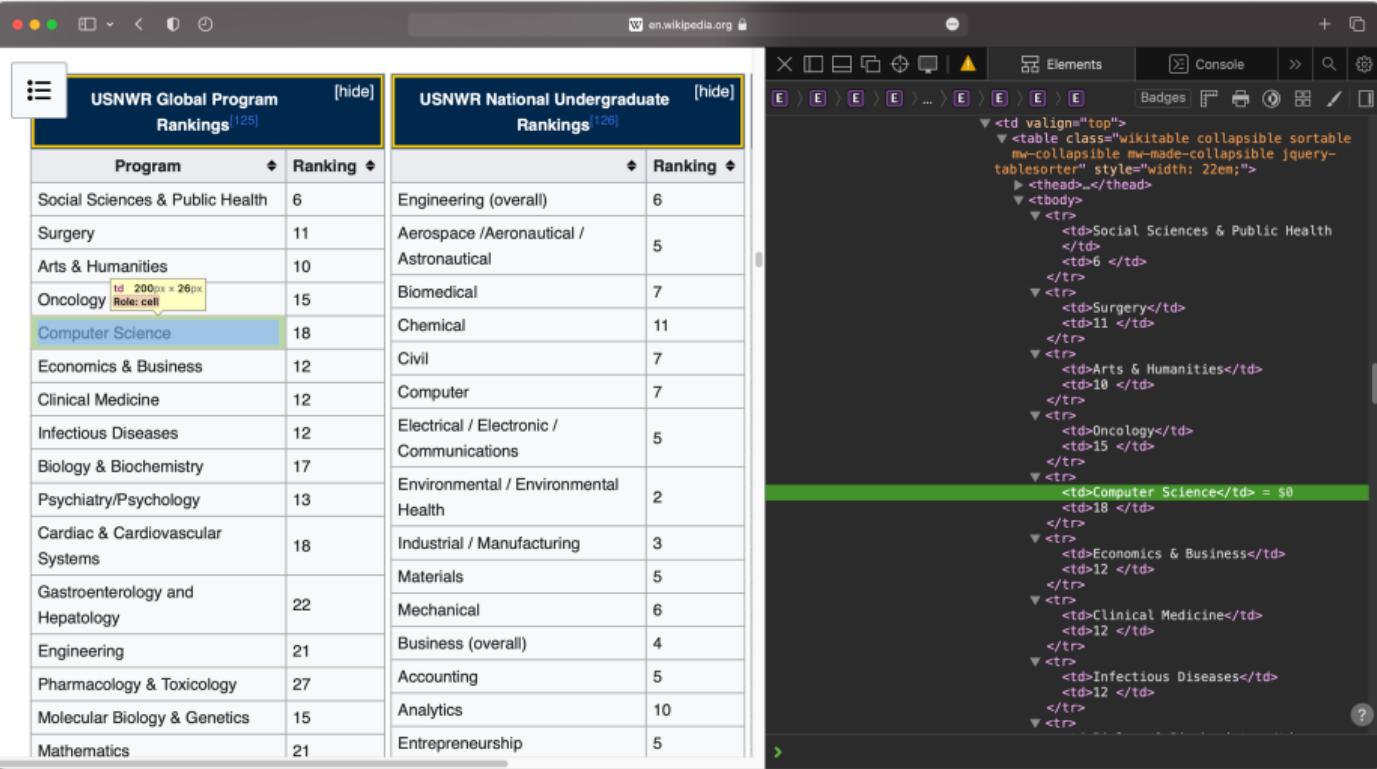
Regular Expression

HTML

Beautiful Soup

Practice
Problem

Task 2: Get School Rankings



Program	Ranking
Social Sciences & Public Health	6
Surgery	11
Arts & Humanities	10
Oncology	15
Computer Science	18
Economics & Business	12
Clinical Medicine	12
Infectious Diseases	12
Biology & Biochemistry	17
Psychiatry/Psychology	13
Cardiac & Cardiovascular Systems	18
Gastroenterology and Hepatology	22
Engineering	21
Pharmacology & Toxicology	27
Molecular Biology & Genetics	15
Mathematics	21

	Ranking
Engineering (overall)	6
Aerospace /Aeronautical / Astronautical	5
Biomedical	7
Chemical	11
Civil	7
Computer	7
Electrical / Electronic / Communications	5
Environmental / Environmental Health	2
Industrial / Manufacturing	3
Materials	5
Mechanical	6
Business (overall)	4
Accounting	5
Analytics	10
Entrepreneurship	5

```
<td valign="top">
  <table class="wikitable collapsible sortable mw-collapsible mw-movable-collapsible jquery-tablesorter" style="width: 22em;">
    <thead></thead>
    <tbody>
      <tr>
        <td>Social Sciences & Public Health</td>
        <td>6 </td>
      </tr>
      <tr>
        <td>Surgery</td>
        <td>11 </td>
      </tr>
      <tr>
        <td>Arts & Humanities</td>
        <td>10 </td>
      </tr>
      <tr>
        <td>Oncology</td>
        <td>15 </td>
      </tr>
      <tr>
        <td>Computer Science</td> = $0
        <td>18 </td>
      </tr>
      <tr>
        <td>Economics & Business</td>
        <td>12 </td>
      </tr>
      <tr>
        <td>Clinical Medicine</td>
        <td>12 </td>
      </tr>
      <tr>
        <td>Infectious Diseases</td>
        <td>12 </td>
      </tr>
    </tbody>
  </table>
</td>
```