DATA SCIENCE WITH PYTHON

CS 498 (SPECIAL TOPICS IN COMPUTER SCIENCE)

WEB SCRAPING

Dr. Malak Abdullah

Computer Science Department

Jordan University of Science and Technology

WHAT IS WEB SCRAPING

- Web scraping is about downloading **structured data** from the web, selecting some of that data, and passing along what you selected to another process.

- It's the process of extracting information from a web page **by taking advantage of patterns** in the web page's underlying code.

OUR EXAMPLE TODAY

- On July 21, 2017, the New York Times updated an opinion article called Trump's Lies, detailing every public lie the President has told since taking office.
- This is a newspaper, the information was published as a block of text
- Do you know how to build a website?

Examining the New York Times article

Here's the way the article presented the information:

> **JAN. 21** "I wasn't a fan of Iraq. I didn't want to go into Iraq." *(He was for an invasion before he was against it.)* **JAN. 21** "A reporter for Time magazine — and I have been on their cover 14 or 15 times. I think we have the all-time record in the history of Time magazine." *(Trump was on the cover 11 times and Nixon appeared 55 times.)* **JAN. 23** "Between 3 million and 5 million illegal votes caused me to lose the popular vote." *(There's no evidence of illegal voting.)* **JAN. 25** "Now, the audience was the biggest ever. But this crowd was massive. Look how far back it goes. This crowd was massive." *(Official aerial photos show Obama's 2009 inauguration was much more heavily*

When converting this into a dataset, **you can think of each lie as a "record" with four fields:**

1. The date of the lie.
2. The lie itself (as a quotation).
3. The writer's brief explanation of why it was a lie.
4. The URL of an article that substantiates the claim that it was a lie.

BEFORE WE START

- let's understand the basics of HTML and some rules of scraping
- To view the HTML code that generates a web page, you right click on it and select "View Page Source" or "view source"
- Every <tag> serves a block inside the webpage
- *Tags* can have attributes

  <img src="img_girl.jpg">

  <p title="title">

- Remember that *Tags* can be nested

```
<!DOCTYPE html>
<html>
    <head>
    </head>
    <body>
        <h1> First Scraping </h1>
        <p> Hello World </p>
    <body>
</html>
```

IMPORT THE FOLLOWING

- BeautifulSoup4, bs4 : for handling all of your HTML processing
- urllib, or requests : for performing your HTTP requests

Python

```
from requests import get
from requests.exceptions import RequestException
from contextlib import closing
from bs4 import BeautifulSoup
```

```
 6
 7
 8  import bs4
 9  from bs4 import BeautifulSoup
10  import urllib
11  import re
12  import urllib.request
13
14
```

BEAUTIFULSOUP ☺

Making sense of HTML structure

```
33
34  html = """<html>
35  <head>
36  <title>Employee Profile</title>
37  <meta charset="utf-8"/>
38  </head>
39  <body>
40  <div class="name"><b>Name:</b>Dr Peter Parker</div>
41  <div class="job"><b>Job:</b>Machine Learning Engineer</div>
42  <div class="telephone"><b>Telephone:</b>+12345678910</div>
43  <div class="email"><b>Email:</b><a href="mailto:peteparker@svalley.com">
44  peteparker@svalley.com</a></div>
45  <div class="website"><b>Website:</b><a href="http://pparkerworks.com">
46  pparkerworks.com</a></div>
47  </body>
48  </html>
49  """
50
51
52  soup = BeautifulSoup(html, "lxml")
53  print (soup)
54
```

```
<html>
<head>
<title>Employee Profile</title>
<meta charset="utf-8"/>
</head>
<body>
<div class="name"><b>Name:</b>Dr Peter Parker</div>
<div class="job"><b>Job:</b>Machine Learning Engineer</div>
<div class="telephone"><b>Telephone:</b>+12345678910</div>
<div class="email"><b>Email:</b><a href="mailto:peteparker@svalley.com">
peteparker@svalley.com</a></div>
<div class="website"><b>Website:</b><a href="http://pparkerworks.com">
pparkerworks.com</a></div>
</body>
</html>
```

PRETTIFY

```
51
52  soup = BeautifulSoup(html, "lxml")
53  print (soup)
54
55  print (soup.prettify())
56
```

```
<html>
<head>
<title>Employee Profile</title>
<meta charset="utf-8"/>
</head>
<body>
<div class="name"><b>Name:</b>Dr Peter Parker</div>
<div class="job"><b>Job:</b>Machine Learning Engineer</div>
<div class="telephone"><b>Telephone:</b>+12345678910</div>
<div class="email"><b>Email:</b><a href="mailto:peteparker@svalley.com">
peteparker@svalley.com</a></div>
<div class="website"><b>Website:</b><a href="http://pparkerworks.com">
pparkerworks.com</a></div>
</body>
</html>

<html>
 <head>
  <title>
   Employee Profile
  </title>
  <meta charset="utf-8"/>
 </head>
 <body>
  <div class="name">
   <b>
    Name:
   </b>
   Dr Peter Parker
```

FIND TAGS

propython.py

```
print (soup.prettify())
print ("*"*100)

print (soup.find("div"))
print ("*"*100)

print (soup.find_all("div"))
```

```
   </b>
   <a href="http://pparkerworks.com">
    pparkerworks.com
   </a>
  </div>
 </body>
</html>

****************************************************************************************************
<div class="name"><b>Name:</b>Dr Peter Parker</div>
****************************************************************************************************
[<div class="name"><b>Name:</b>Dr Peter Parker</div>, <div class="job"><b>Job:</b>Machine Learning Engineer</div>, <
div class="telephone"><b>Telephone:</b>+12345678910</div>, <div class="email"><b>Email:</b><a
href="mailto:peteparker@svalley.com">
peteparker@svalley.com</a></div>, <div class="website"><b>Website:</b><a href="http://pparkerworks.com">
pparkerworks.com</a></div>]
[Finished in 2.9s]
```

FIND AND FINDALL

propython.py

```
66  print ("*"*100)
67  for link in soup.findAll('a', attrs={'href': re.compile("^http://")}):
68      print (link.get('href'))
69
70  print ("*"*100)
71
72  print (soup.find("body").get_text())
```

```
****************************************************************************************************
http://pparkerworks.com
****************************************************************************************************

Name:Dr Peter Parker
Job:Machine Learning Engineer
Telephone:+12345678910
Email:
peteparker@svalley.com
Website:
pparkerworks.com
```

- find( ): searches for the first matching tag, and returns a Tag object

- find_all( ): searches for all matching tags, and returns a ResultSet object (which you can treat like a list of Tags)

READING WEB!

USE THIS!

```
 8  import bs4
 9  from bs4 import BeautifulSoup
10  import urllib
11  import re
12  import urllib.request
13
14
15  req = urllib.request.Request('http://www.just.edu.jo/Pages/Def
16  html = urllib.request.urlopen(req).read()
17  print(html)
18
```

```
b'\r\n\r\n\r\n\r\n<!DOCTYPE HTML>\r\n<html dir="ltr" class="ms-isBo
lang="en-US">\r\n    <head><meta charset="utf-8" /><meta
http-equiv="X-UA-Compatible" content="IE=edge" /><meta name="GENERA
content="Microsoft SharePoint" /><meta http-equiv="Content-type"
content="text/html; charset=utf-8" /><meta http-equiv="Expires"
content="0" /><meta name="viewport" content="width=device-width,
initial-scale=1" />\r\n\t\t\r\n\t\t<!-- this meta from majestic for
webometrics purposes -->\r\n\t\t<meta name="majestic-site-verificat
content="MJ12_5a0c36ec-d0ea-42e6-89a2-1249de3f180c"
/>\r\n\t\t\r\n\t\t\r\n        <!-- ===== STARTER: Robots  ==========
```

OR REQUESTS → WE WILL USE THIS NOW

```
75  import requests
76  r = requests.get('https://www.nytimes.com/interactive/2017/06/
77  print (r.text[1:300])
78
79
80
```

```
!DOCTYPE html>
<!--[if (gt IE 9)|!(IE)]> <!--><html lang="en" class="no-js
page-interactive section-opinion page-theme-standard tone-opinion
page-interactive-default limit-small layout-xlarge app-interactive"
itemid="https://www.nytimes.com/interactive/2017/06/23/opinion/
trumps-lies.html" itemtype=
[Finished in 1.6s]
```

from bs4 import BeautifulSoup

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

KNOWING EACH RECORD CONSISTS OF:

 DATE LIE EXPLANATION

record in the history of Time magazine." *(Trump was on the cover 11 times and Nixon appeared 55 times.)* **JAN. 23** "Between 3 million and 5 million illegal votes caused me to lose the popular vote." *(There's no evidence of illegal voting.)* **JAN. 25** "Now, the audience was the biggest ever. But this crowd was massive. Look how far back it goes. This crowd was massive." *(Official aerial photos show Obama's 2009 inauguration was much more heavily*

```
094 <span class="short-desc"><strong>Jan. 21 </strong>"I wasn't a
fan of Iraq. I didn't want to go into Iraq." <span class="short-
truth"><a href="https://www.buzzfeed.com/andrewkaczynski/in-2002-
donald-trump-said-he-supported-invading-iraq-on-the"
target="_blank">(He was for an invasion before he was against it.)
</a></span></span>  <span class="short-desc"><strong>Jan.
21 </strong>"A reporter for Time magazine — and I have been on
their cover 14 or 15 times. I think we have the all-time record in
the history of Time magazine." <span class="short-truth"><a
href="http://nation.time.com/2013/11/06/10-things-you-didnt-know-
about-time/" target="_blank">(Trump was on the cover 11 times and
Nixon appeared 55 times.)</a></span></span>  <span
```

WE CAN FIND ALL THE LIES!

```
 7
 8  import bs4
 9  from bs4 import BeautifulSoup
10  import requests
11  r = requests.get('https://www.nytimes.com/interactive/2017/06/23/opinion/tru
12  soup = BeautifulSoup(r.text, 'html.parser')
13  results = soup.find_all('span', attrs={'class':'short-desc'})
14  print (len(results))
15  print (results[5])
16
17
```

```
180
<span class="short-desc"><strong>Jan. 25 </strong>“You had millions of people that now
aren't insured anymore.” <span class="short-truth"><a
href="https://www.nytimes.com/2017/03/13/us/politics/
fact-check-trump-obamacare-health-care.html" target="_blank">(The real number is less
than 1 million, according to the Urban Institute.)</a></span></span>
[Finished in 1.6s]
```

EXTRACTING THE DATE

```
13  results = soup.find_all('span', attrs={'c
14  print (len(results))
15  print (results[5])
16  print ("*"*100)
17  x=results[5].find('strong').text
18  print (x , len(x))
19  y=results[5].find('strong').text[0:-1]
20  print (y , len(y))
21
22
```

```
****************************************************************************************************
Jan. 25  8
Jan. 25 7
```

EXTRACTING THE LIE

❖There is no Tag !!!!!!!!!!!!

Jan. 21 “I wasn't a fan of Iraq. I didn't want to go into Iraq.”(He was for an invasion before he was against it.)

SO → use contents

```
22
23  print (results[1].contents)
24  print ("*"*30)
25  print (results[1].contents[1])
26
```

```
[<strong>Jan. 21 </strong>, '“A reporter for Time magazine – and I have been on their cover 14 or 15
times. I think we have the all-time record in the history of Time magazine.” ', <span
class="short-truth"><a href="http://nation.time.com/2013/11/06/10-things-you-didnt-know-about-time/"
target="_blank">(Trump was on the cover 11 times and Nixon appeared 55 times.)</a></span>]
******************************
“A reporter for Time magazine – and I have been on their cover 14 or 15 times. I think we have the
all-time record in the history of Time magazine.”
[Finished in 2.6s]
```

•text: extracts the text of a Tag, and returns a string

•contents: extracts the children of a Tag, and returns a list of Tags and strings

EXTRACTING THE EXPLANATION & URL

- For explanation:
  - Use this → results[x].contents[2]
  - Or use this → results[x].find('a')

- For URL:
  - Results[x].find('a')['href']

(He was for an invasion before he was against it.)

BUILDING THE DATASET

```
29
30  records = []
31  for result in results:
32      date = result.find('strong').text[0:-1] + ', 2017'
33      lie = result.contents[1][1:-2]
34      explanation = result.find('a').text[1:-1]
35      url = result.find('a')['href']
36      records.append((date, lie, explanation, url))
37
38
39  print (records[1])
40  print ("*"*30)
41
42  print (records[1][0])
43
```

```
('Jan. 21, 2017', 'A reporter for Time magazine – and I have been on their cover 14 or 15 times. I
think we have the all-time record in the history of Time magazine.', 'Trump was on the cover 11 times
and Nixon appeared 55 times.',
'http://nation.time.com/2013/11/06/10-things-you-didnt-know-about-time/')
******************************
Jan. 21, 2017
```

USING PANDA TO GET DATAFRAME

```
import pandas as pd
df = pd.DataFrame(records, columns=['date', 'lie', 'explanation', 'url'])
```

The DataFrame includes a `head()` method, which allows you to examine the top of the DataFrame:

```
df.head()
```

	date	lie	explanation	url
0	Jan. 21, 2017	I wasn't a fan of Iraq. I didn't want to go in...	He was for an invasion before he was against it.	https://www.buzzfeed.com/andrewkaczynski/in-20...
1	Jan. 21, 2017	A reporter for Time magazine — and I have been...	Trump was on the cover 11 times and Nixon appe...	http://nation.time.com/2013/11/06/10-things-yo...
2	Jan. 23, 2017	Between 3 million and 5 million illegal votes ...	There's no evidence of illegal voting.	https://www.nytimes.com/2017/01/23/us/politics...
3	Jan. 25, 2017	Now, the audience was the biggest ever. But th...	Official aerial photos show Obama's 2009 inaug...	https://www.nytimes.com/2017/01/21/us/politics...
4	Jan. 25, 2017	Take a look at the Pew reports (which show vot...	The report never mentioned voter fraud.	https://www.nytimes.com/2017/01/24/us/politics...

EXPORTING THE DATASET TO A CSV FILE

- df.to_csv('trump_lies.csv', index=**False**, encoding='utf-8')

- If you need to read it later:

df = pd.read_csv('trump_lies.csv', parse_dates=['date'], encoding='utf-8')

Out References for this class are

https://github.com/justmarkham/trump-lies/blob/master/trump_lies.ipynb

https://www.youtube.com/playlist?list=PL5-da3qGB5IDbOi0g5WFh1YPDNzXw4LNL