

Quiz 1: Web Mining with BeautifulSoup

Casey Owen

September 5, 2024

CS119 Big Data

Spring 2024

Objective: To use BeautifulSoup, the Python Library, to extract some information from the internet about BeautifulSoup, the Lewis Carroll poem.

```
[1]: from bs4 import BeautifulSoup
import requests

def make_soup(url):
    '''
    Creates a soup object from a given url on the web
    '''
    html_content = requests.get(url).text
    soup = BeautifulSoup(html_content, "html.parser")
    return soup
```

1 books()

```
[2]: def books():
    '''
    Returns the names of the canon books as a list.
    '''
    home_url = "https://aliceinwonderland.fandom.com/wiki/"
    ↪Alice_in_Wonderland_Wiki"
    soup = make_soup(home_url)
    nav_bar = soup.find(class_='fandom-community-header__local-navigation')
    tabs = nav_bar.find(class_='wds-tabs')
    # Gets the second child in the tabs - the lit tab
    lit_tab = tabs.findChildren("li", recursive=False)[1]
    lit_content = lit_tab.find("ul", class_="wds-list wds-is-linked")
    # Canon books is the first option in the literature tab
    canon_books_tab = lit_content.findChildren("li", recursive=False)[0]
    canon_books = canon_books_tab.find("ul", class_="wds-list wds-is-linked")
    # All names are listed in spans
    book_names = canon_books.find_all("span")
    book_names_text = [name.get_text() for name in book_names]
```

```
return book_names_text
```

1.1 Test the output against known answer

```
[3]: result = books()
print(result)

p1_answer = [
    "Alice's Adventures in Wonderland",
    'Through the Looking-Glass, and What Alice Found There',
    "Alice's Adventures Underground"
]
if result == p1_answer:
    print('CORRECT')
else:
    print("INCORRECT")
```

```
["Alice's Adventures in Wonderland", 'Through the Looking-Glass, and What Alice
Found There', "Alice's Adventures Underground"]
CORRECT
```

2 poems()

```
[4]: def poems():
    """
    Returns the names of the canon poems and their urls, as a list of tuples.
    """
    home_url = "https://aliceinwonderland.fandom.com/wiki/
↪Alice_in_Wonderland_Wiki"
    soup = make_soup(home_url)
    nav_bar = soup.find(class_='fandom-community-header__local-navigation')
    tabs = nav_bar.find(class_='wds-tabs')
    # Gets the second child in the tabs - the lit tab
    lit_tab = tabs.findChildren("li", recursive=False)[1]
    lit_content = lit_tab.find("ul", class_="wds-list wds-is-linked")
    # Canon poems is the second option in the literature tab
    canon_poems_tab = lit_content.findChildren("li",
↪class_="wds-dropdown-level-nested", recursive=False)[1]
    canon_poems = canon_poems_tab.find("ul", class_="wds-list wds-is-linked")
    poem_names = canon_poems.find_all("span")
    poem_names_text = [name.get_text() for name in poem_names]
    # Links are listed as the href part of an a tag
    poem_links = canon_poems.find_all("a")
    poem_links_text = [link['href'] for link in poem_links]
    return list(zip(poem_names_text, poem_links_text))
```

2.1 Test the output against known answer

```
[5]: result = poems()
print(result)

p2_answer = [
    ('Jabberwocky', 'https://aliceinwonderland.fandom.com/wiki/Jabberwocky'),
    ('How Doth the Little Crocodile', 'https://aliceinwonderland.fandom.com/
↪wiki/How_Doth_the_Little_Crocodile'),
    ('The Walrus and the Carpenter', 'https://aliceinwonderland.fandom.com/wiki/
↪The_Walrus_and_the_Carpenter_(poem)'),
    ('You Are Old, Father William', 'https://aliceinwonderland.fandom.com/wiki/
↪You_Are_Old,_Father_William'),
    ('Humpty Dumpty's Recitation', 'https://aliceinwonderland.fandom.com/wiki/
↪Humpty_Dumpty%27s_Recitation'),
    ('Turtle Soup', 'https://aliceinwonderland.fandom.com/wiki/Turtle_Soup'),
    ('Tis the Voice of the Lobster', 'https://aliceinwonderland.fandom.com/wiki/
↪Tis_the_Voice_of_the_Lobster')
]
if result == p2_answer:
    print('CORRECT')
else:
    print("INCORRECT")
```

```
[('Jabberwocky', 'https://aliceinwonderland.fandom.com/wiki/Jabberwocky'), ('How
Doth the Little Crocodile',
'https://aliceinwonderland.fandom.com/wiki/How_Doth_the_Little_Crocodile'),
('The Walrus and the Carpenter', 'https://aliceinwonderland.fandom.com/wiki/The_
Walrus_and_the_Carpenter_(poem)'), ('You Are Old, Father William',
'https://aliceinwonderland.fandom.com/wiki/You_Are_Old,_Father_William'),
('Humpty Dumpty's Recitation',
'https://aliceinwonderland.fandom.com/wiki/Humpty_Dumpty%27s_Recitation'),
('Turtle Soup', 'https://aliceinwonderland.fandom.com/wiki/Turtle_Soup'), ('Tis
the Voice of the Lobster',
'https://aliceinwonderland.fandom.com/wiki/Tis_the_Voice_of_the_Lobster')]
CORRECT
```

3 poem_title_text()

```
[6]: def poem_title_text(n):
    """
    Returns the corresponding poem title and text, according to the order of_
    ↪outputs of poems(). Only works for n=2 or n=5.
    """
    if n not in [2,5]:
        raise ValueError('n can only be 2 or 5')
```

```

title, link = poems()[n]
soup = make_soup(link)
text_h2 = soup.select('span#Text')[0]
poem_paragraphs = text_h2.find_all_next('p')
poem_text = ""
for paragraph in poem_paragraphs:
    poem_text += paragraph.get_text()
return title, poem_text

```

3.1 Display the output

```

[7]: for i in [2,5]:
    title, text = poem_title_text(i)
    print(title + '\n')
    print(text)

```

The Walrus and the Carpenter

The sun was shining on the sea,
 Shining with all his might:
 He did his very best to make
 The billows smooth and bright--
 And this was odd, because it was
 The middle of the night.
 The moon was shining sulkily,
 Because she thought the sun
 Had got no business to be there
 After the day was done--
 "It's very rude of him," she said,
 "To come and spoil the fun!"
 The sea was wet as wet could be,
 The sands were dry as dry.
 You could not see a cloud, because
 No cloud was in the sky:
 No birds were flying overhead--
 There were no birds to fly.
 The Walrus and the Carpenter
 Were walking close at hand;
 They wept like anything to see
 Such quantities of sand:
 "If this were only cleared away,"
 They said, "it would be grand!"
 "If seven maids with seven mops
 Swept it for half a year.
 Do you suppose," the Walrus said,
 "That they could get it clear?"
 "I doubt it," said the Carpenter,

And shed a bitter tear.
"O Oysters, come and walk with us!"
The Walrus did beseech.
"A pleasant walk, a pleasant talk,
Along the briny beach:
We cannot do with more than four,
To give a hand to each."
The eldest Oyster looked at him,
But never a word he said:
The eldest Oyster winked his eye,
And shook his heavy head--
Meaning to say he did not choose
To leave the oyster-bed.
But four young Oysters hurried up,
All eager for the treat:
Their coats were brushed, their faces washed,
Their shoes were clean and neat--
And this was odd, because, you know,
They hadn't any feet.
Four other Oysters followed them,
And yet another four;
And thick and fast they came at last,
And more, and more, and more--
All hopping through the frothy waves,
And scrambling to the shore.
The Walrus and the Carpenter
Walked on a mile or so,
And then they rested on a rock
Conveniently low:
And all the little Oysters stood
And waited in a row.
"The time has come," the Walrus said,
"To talk of many things:
Of shoes--and ships--and sealing-wax--
Of cabbages--and kings--
And why the sea is boiling hot--
And whether pigs have wings."
"But wait a bit," the Oysters cried,
"Before we have our chat;
For some of us are out of breath,
And all of us are fat!"
"No hurry!" said the Carpenter.
They thanked him much for that.
"A loaf of bread," the Walrus said,
"Is what we chiefly need:
Pepper and vinegar besides
Are very good indeed--
Now if you're ready, Oysters dear,

We can begin to feed."
"But not on us!" the Oysters cried,
Turning a little blue.
"After such kindness, that would be
A dismal thing to do!"
"The night is fine," the Walrus said.
"Do you admire the view?"
"It was so kind of you to come!
And you are very nice!"
The Carpenter said nothing but
"Cut us another slice:
I wish you were not quite so deaf--
I've had to ask you twice!"
"It seems a shame," the Walrus said,
"To play them such a trick,
After we've brought them out so far,
And made them trot so quick!"
The Carpenter said nothing but
"The butter's spread too thick!"
"I weep for you," the Walrus said:
"I deeply sympathize."
With sobs and tears he sorted out
Those of the largest size,
Holding his pocket-handkerchief
Before his streaming eyes.
"O Oysters," said the Carpenter,
"You've had a pleasant run!
Shall we be trotting home again?"
But answer came there none--
And this was scarcely odd, because
They'd eaten every one.

Turtle Soup

Beautiful Soup, so rich and green,
Waiting in a hot tureen!
Who for such dainties would not stoop?
Soup of the evening, beautiful Soup!
Soup of the evening, beautiful Soup!
Beau--ootiful Soo--oop!
Beau--ootiful Soo--oop!
Soo--oop of the e--e--evening,
Beautiful, beautiful Soup!
Beautiful Soup! Who cares for fish,
Game or any other dish?
Who would not give all else for two
Pennyworth only of Beautiful Soup?
Pennyworth only of beautiful Soup?

Beau--ootiful Soo--oop!
Beau--ootiful Soo--oop!
Soo--oop of the e--e--evening,
Beautiful, beauti--FUL SOUP!