Web Scraping

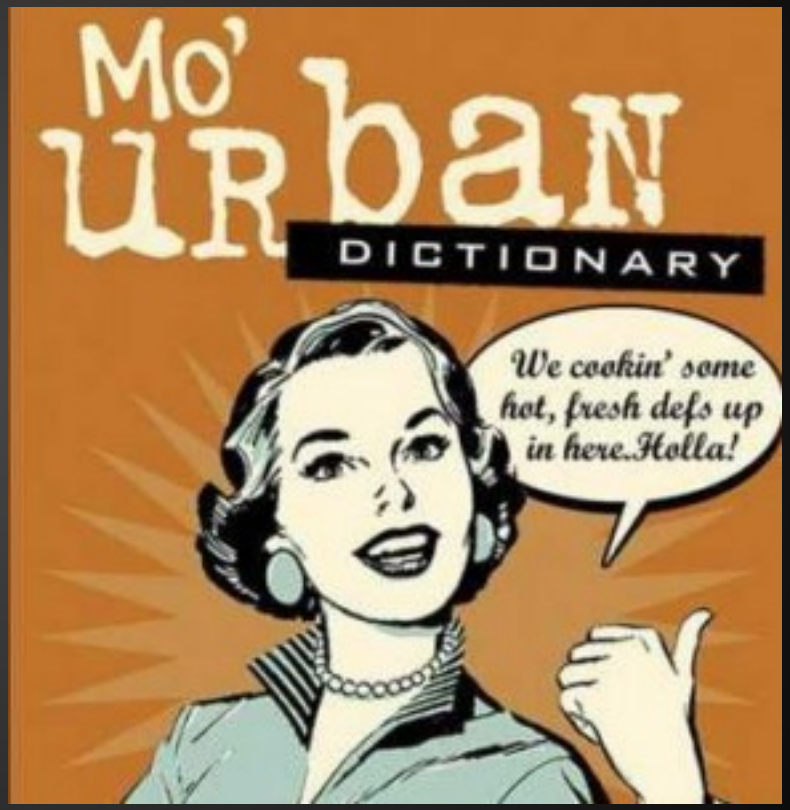
Scraping steps

Steps:

1. Figure out the URL(s) to scrape
2. Write code to download the HTML given the URL
3. Find unique HTML tags surrounding your target information
   a. Here's the title!
4. Write code using string manipulation to extract relevant data
5. Display and/or store scraped information

Case Study: Urban Dictionary

- We will write a scraper for Urban Dictionary
- Given a term inputted by our user, print the first definition and example found on Urban Dictionary for that term



Step 1: Figure out URLs

- First thing we do is visit the website!
- Click a few definitions on urbandictionary.com
- Try to find a pattern in the URLs
  - Find the “minimum URL” that will give you a certain page

Figure Out the URL

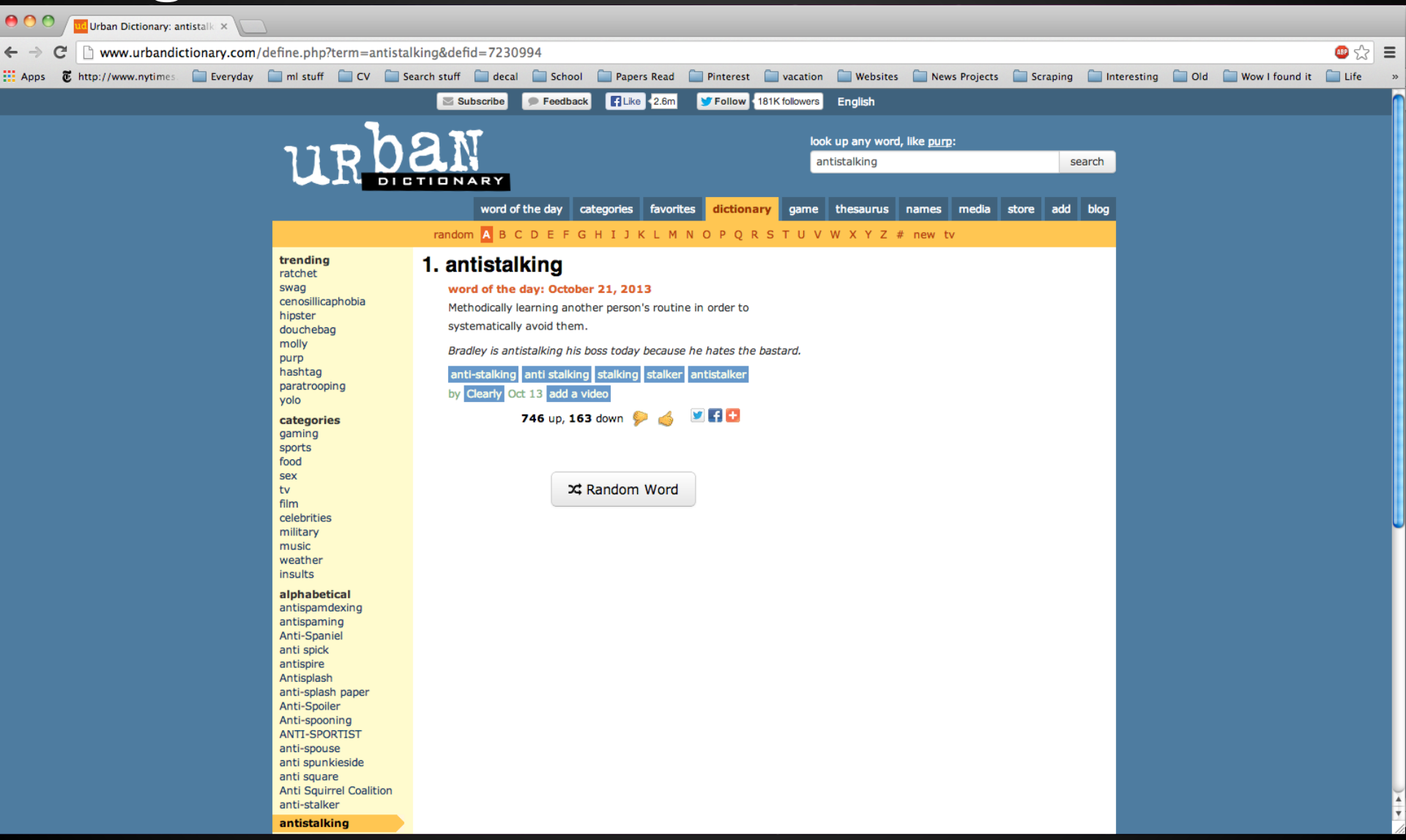
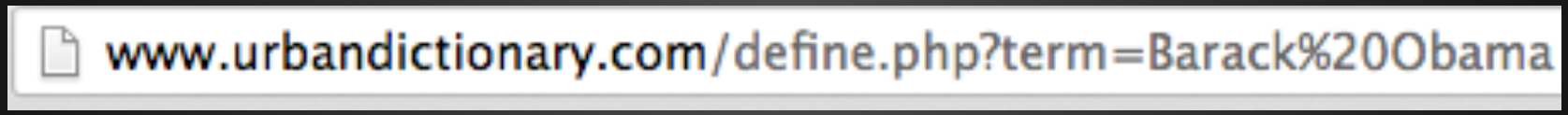


Figure Out the URL

Figure Out the URL

Figure Out the URL



Step 2: Download the HTML

```
import urllib2
url = 'http://www.urbandictionary.com/defi
    ne.php?term=antistalking’
html = urllib2.urlopen(url).read()
print html
```

Note that this will only work in the terminal!

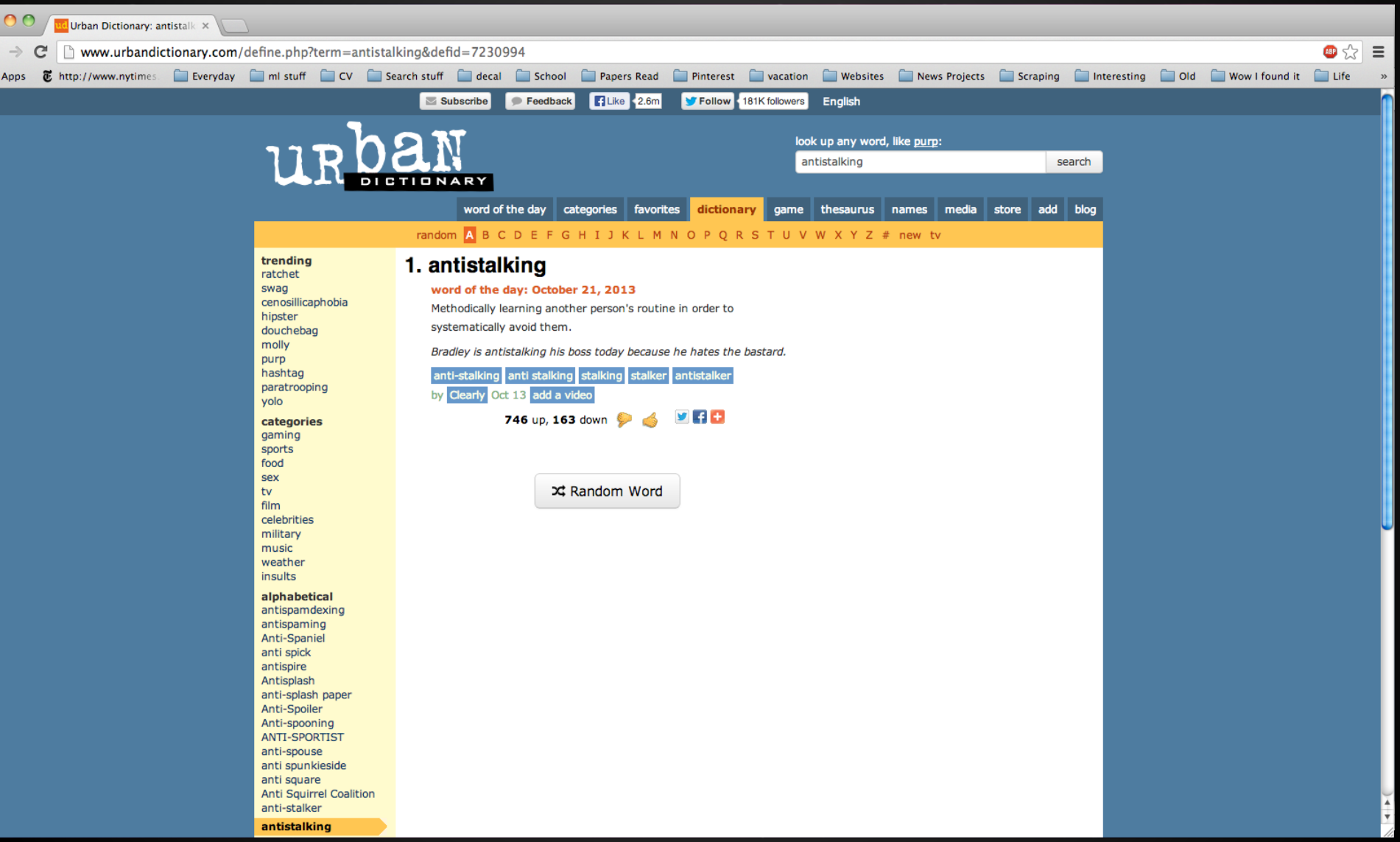
How can we allow user input?

```
import urllib2
user_word = raw_input(‘Define a word: ’)
url = 'http://www.urbandictionary.com/defi
    ne.php?term=’ + user_word
html = urllib2.urlopen(url).read()
print html
```

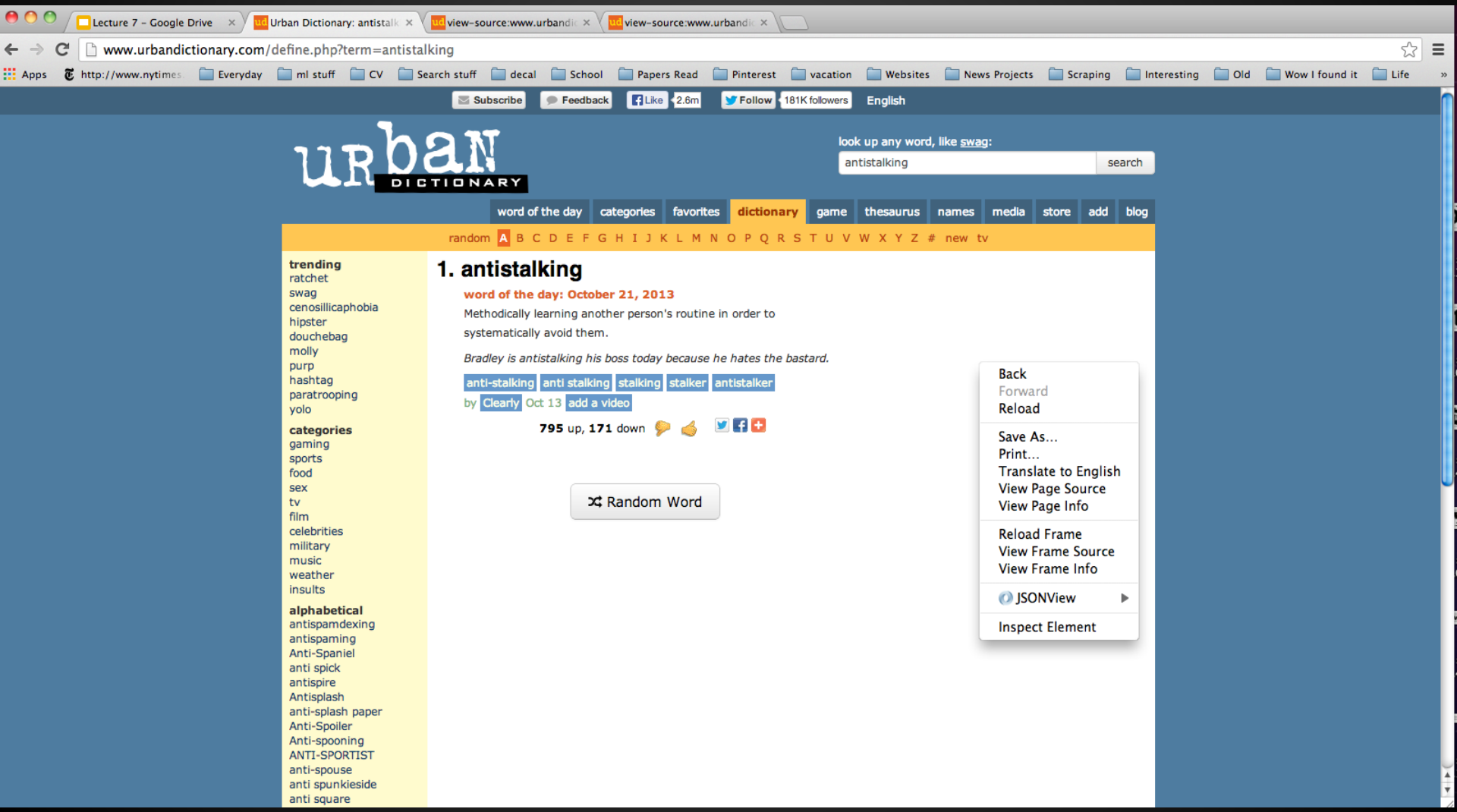
Step 3: Find unique HTML tags

- View page source (or look at the html you printed to the terminal)
- Find the first definition / example on the page
  - Use Ctrl-F to find specific words!
- Look for surrounding tags/patterns
  - If we're lucky they'll be unique

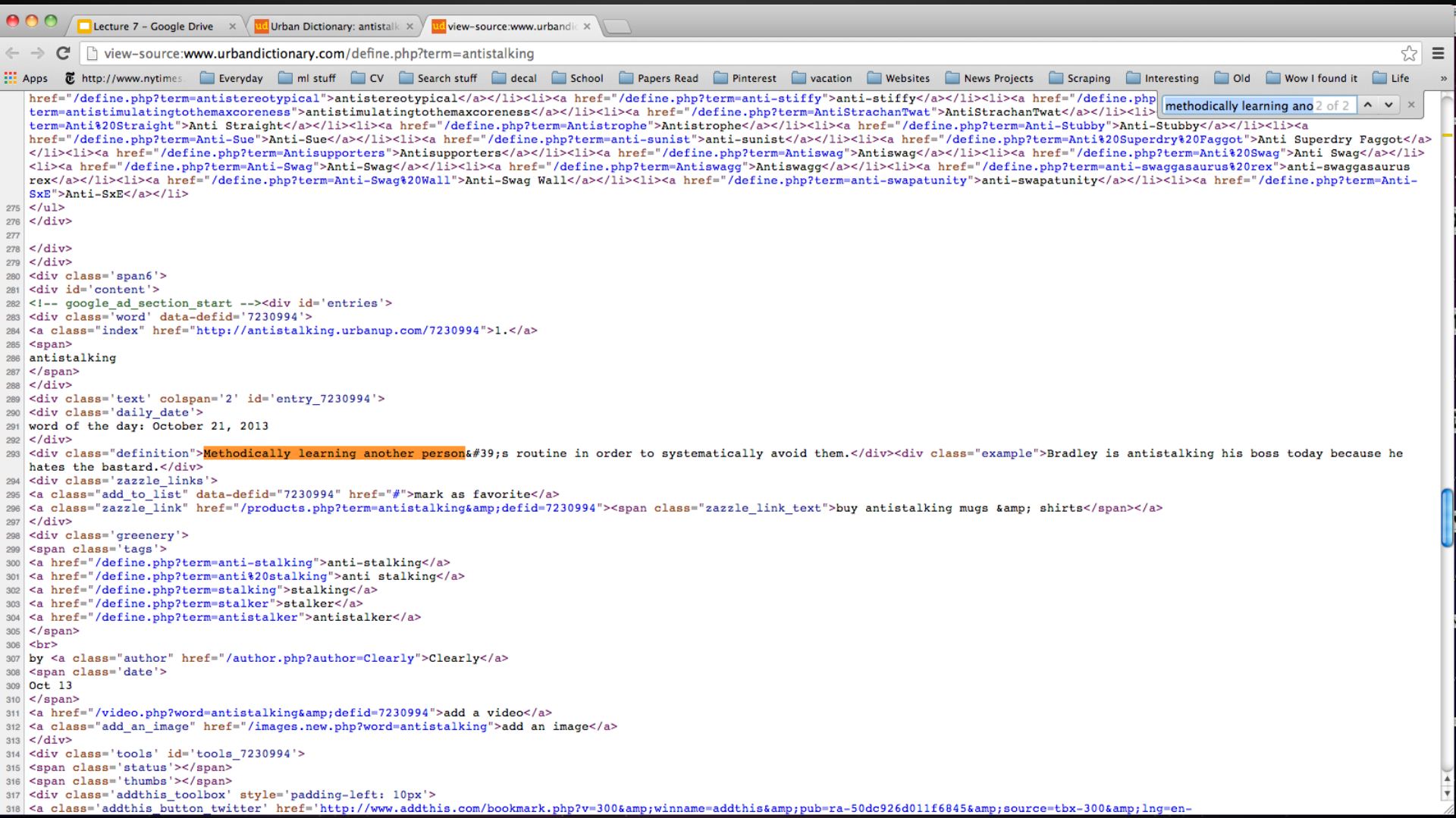
Find Unique HTML Tags



Find Unique HTML Tags

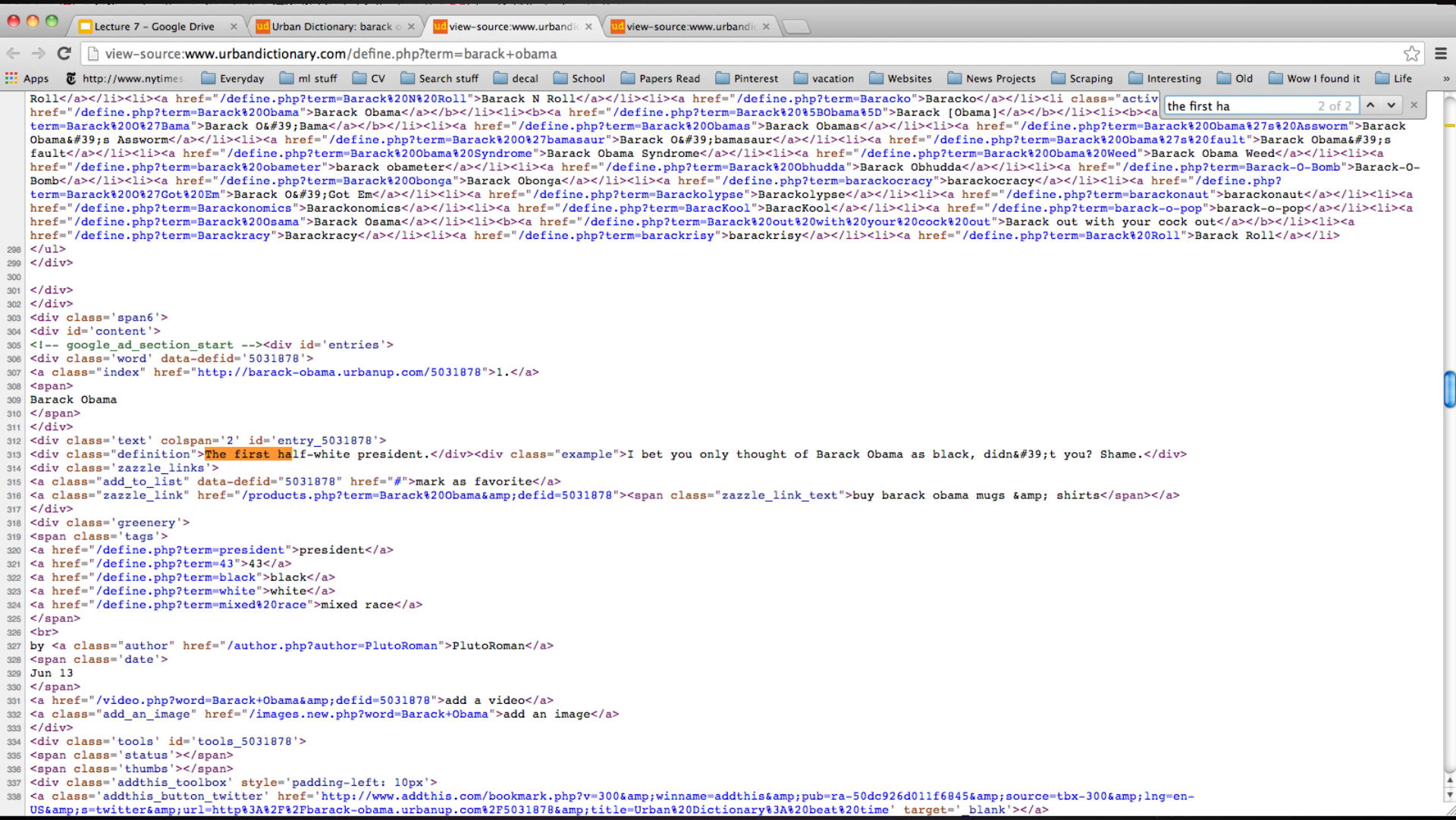


Find Unique HTML Tags



Find Unique HTML Tags

Find Unique HTML Tags



Step 4: Write code to extract data

- Given the HTML string, how do we pull out just the information between our target tags?
- Ex.
  - html = ‘This is a long block of text blah blah this part is important! more stuff we could care less about’
- We need more string manipulation!

Advanced String Manipulation

- string.find(s) returns the index of s in the given string
  - index means the number of characters from the beginning of the string
  - ‘hello’.find(‘h’) # returns 0
  - ‘hello’.find(‘e’) # returns 1
  - ‘hello’.find(‘l’) # returns 2
  - ‘hello’.find(‘o’) # returns 4
  - ‘hello’.find(‘hell’) # returns 0
  - ‘hello’.find(‘llo’) # returns 3
  - ‘hello’.find(‘b’) # returns -1

Advanced String Manipulation

- string[a:b] returns the substring of string from a (inclusive) to b (non-inclusive)
  - can also use string[a:] to go through end of word or string [:b] to go from beginning of word
  - ‘happiness’[0:2] # returns ‘ha’
  - ‘happiness’[1:4] # returns ‘app’
  - ‘happiness’[-4:-1] # returns ‘nes’
  - ‘happiness’[-4:] # returns ‘ness’
  - ‘happiness’[:4] # returns ‘happ’

Advanced String Manipulation

- string.replace(old, new) replaces all occurrences of old with new in string
  - “brown fox”.replace(“brown”, “red”) # returns ‘red fox’
  - “brown fox”.replace(“fox”, “cow”) # returns ‘brown cow’
  - “brown fox”.replace(“ ”, “,”) # returns ‘brown,fox’
  - “brown fox”.replace(“o”, “a”) # returns ‘brawn fax’

Advanced String Manipulation

```
words = ‘hello there’
print words.split(‘ ’)                  # [‘hello’, ‘there’]
print words.find(‘the’)                 # 6
print words[6]                          # ‘t’
print words[6:9]                        # ‘the’
print words.replace(‘the’, ‘sart’)      # ‘hello sartre’
```

Where to look stuff up

- Documentation: http://docs.python.org/2/library/string.html
  - Ctrl-F and search for “string.find” in the docs
- Google!
  - ex. “how to split a string in python”

Step 4: Write code to extract data

html = ‘This is a long block of text blah blah
this part is important!
more stuff we could care less about’

start_idx = html.find(‘ ’)

definition_substring = html[start_idx:]

print definition_substring

What will this print?

this part is important! more stuff we could care less about’

Step 4: Write code to extract data

html = ‘This is a long block of text blah blah
this part is important!
more stuff we could care less about’

start_idx = html.find(‘ ’)
start_idx = start_idx + len(‘ ’)

definition_substring = html[start_idx:]

print definition_substring

What will this print?
this part is important! more stuff we could care less about’

Step 4: Write code to extract data

```
html = 'This is a long block of text blah blah
<div class="definition">this part is important!</div>
more stuff we could care less about'
start_idx = html.find('<div class="definition">')
start_idx = start_idx + len('<div class="definition">')
definition_substring = html[start_idx:]
end_idx = definition_substring.find('</div>')
definition_substring = definition_substring[:end_idx]
print definition_substring
```

Step 5: Display or store information

- For now, just print it out!
- In upcoming weeks we will learn how to use Python to save and load text files for long-term use

Another Pop Quiz!