Python course materials

# Web Scraping Exercises - Solutions

## Complete the Tasks Below

**TASK: Import any libraries you think you’ll need to scrape a website.**

# CODE HERE

import requests  
import bs4

**TASK: Use requests library and BeautifulSoup to connect to http://quotes.toscrape.com/ and get the HMTL text from the homepage.**

# CODE HERE

res = requests.get("http://quotes.toscrape.com/")

res.text

'<!DOCTYPE html>\n<html lang="en">\n<head>\n\t<meta charset="UTF-8">\n\t<title>Quotes to Scrape</title>\n <link rel="stylesheet" href="/static/bootstrap.min.css">\n <link rel="stylesheet" href="/static/main.css">\n</head>\n<body>\n <div class="container">\n <div class="row header-box">\n <div class="col-md-8">\n <h1>\n <a href="/" style="text-decoration: none">Quotes to Scrape</a>\n </h1>\n </div>\n <div class="col-md-4">\n <p>\n \n <a href="/login">Login</a>\n \n </p>\n </div>\n </div>\n \n\n<div class="row">\n <div class="col-md-8">\n\n <div class="quote" itemscope itemtype="http://schema.org/CreativeWork">\n <span class="text" itemprop="text">“The world as we have created it is a process of our thinking. It cannot be changed without changing our thinking.”</span>\n <span>by <small class="author" itemprop="author">Albert Einstein</small>\n <a href="/author/Albert-Einstein">(about)</a>\n </span>\n <div class="tags">\n Tags:\n <meta class="keywords" itemprop="keywords" content="change,deep-thoughts,thinking,world" / > \n \n <a class="tag" href="/tag/change/page/1/">change</a>\n \n <a class="tag" href="/tag/deep-thoughts/page/1/">deep-thoughts</a>\n \n <a class="tag" href="/tag/thinking/page/1/">thinking</a>\n \n <a class="tag" href="/tag/world/page/1/">world</a>\n \n </div>\n </div>\n\n <div class="quote" itemscope itemtype="http://schema.org/CreativeWork">\n <span class="text" itemprop="text">“It is our choices, Harry, that show what we truly are, far more than our abilities.”</span>\n <span>by <small class="author" itemprop="author">J.K. Rowling</small>\n <a href="/author/J-K-Rowling">(about)</a>\n </span>\n <div class="tags">\n Tags:\n <meta class="keywords" itemprop="keywords" content="abilities,choices" / > \n \n <a class="tag" href="/tag/abilities/page/1/">abilities</a>\n \n <a class="tag" href="/tag/choices/page/1/">choices</a>\n \n </div>\n </div>\n\n <div class="quote" itemscope itemtype="http://schema.org/CreativeWork">\n <span class="text" itemprop="text">“There are only two ways to live your life. One is as though nothing is a miracle. The other is as though everything is a miracle.”</span>\n <span>by <small class="author" itemprop="author">Albert Einstein</small>\n <a href="/author/Albert-Einstein">(about)</a>\n </span>\n <div class="tags">\n Tags:\n <meta class="keywords" itemprop="keywords" content="inspirational,life,live,miracle,miracles" / > \n \n <a class="tag" href="/tag/inspirational/page/1/">inspirational</a>\n \n <a class="tag" href="/tag/life/page/1/">life</a>\n \n <a class="tag" href="/tag/live/page/1/">live</a>\n \n <a class="tag" href="/tag/miracle/page/1/">miracle</a>\n \n <a class="tag" href="/tag/miracles/page/1/">miracles</a>\n \n </div>\n </div>\n\n <div class="quote" itemscope itemtype="http://schema.org/CreativeWork">\n <span class="text" itemprop="text">“The person, be it gentleman or lady, who has not pleasure in a good novel, must be intolerably stupid.”</span>\n <span>by <small class="author" itemprop="author">Jane Austen</small>\n <a href="/author/Jane-Austen">(about)</a>\n </span>\n <div class="tags">\n Tags:\n <meta class="keywords" itemprop="keywords" content="aliteracy,books,classic,humor" / > \n \n <a class="tag" href="/tag/aliteracy/page/1/">aliteracy</a>\n \n <a class="tag" href="/tag/books/page/1/">books</a>\n \n <a class="tag" href="/tag/classic/page/1/">classic</a>\n \n <a class="tag" href="/tag/humor/page/1/">humor</a>\n \n </div>\n </div>\n\n <div class="quote" itemscope itemtype="http://schema.org/CreativeWork">\n <span class="text" itemprop="text">“Imperfection is beauty, madness is genius and it&#39;s better to be absolutely ridiculous than absolutely boring.”</span>\n <span>by <small class="author" itemprop="author">Marilyn Monroe</small>\n <a href="/author/Marilyn-Monroe">(about)</a>\n </span>\n <div class="tags">\n Tags:\n <meta class="keywords" itemprop="keywords" content="be-yourself,inspirational" / > \n \n <a class="tag" href="/tag/be-yourself/page/1/">be-yourself</a>\n \n <a class="tag" href="/tag/inspirational/page/1/">inspirational</a>\n \n </div>\n </div>\n\n <div class="quote" itemscope itemtype="http://schema.org/CreativeWork">\n <span class="text" itemprop="text">“Try not to become a man of success. Rather become a man of value.”</span>\n <span>by <small class="author" itemprop="author">Albert Einstein</small>\n <a href="/author/Albert-Einstein">(about)</a>\n </span>\n <div class="tags">\n Tags:\n <meta class="keywords" itemprop="keywords" content="adulthood,success,value" / > \n \n <a class="tag" href="/tag/adulthood/page/1/">adulthood</a>\n \n <a class="tag" href="/tag/success/page/1/">success</a>\n \n <a class="tag" href="/tag/value/page/1/">value</a>\n \n </div>\n </div>\n\n <div class="quote" itemscope itemtype="http://schema.org/CreativeWork">\n <span class="text" itemprop="text">“It is better to be hated for what you are than to be loved for what you are not.”</span>\n <span>by <small class="author" itemprop="author">André Gide</small>\n <a href="/author/Andre-Gide">(about)</a>\n </span>\n <div class="tags">\n Tags:\n <meta class="keywords" itemprop="keywords" content="life,love" / > \n \n <a class="tag" href="/tag/life/page/1/">life</a>\n \n <a class="tag" href="/tag/love/page/1/">love</a>\n \n </div>\n </div>\n\n <div class="quote" itemscope itemtype="http://schema.org/CreativeWork">\n <span class="text" itemprop="text">“I have not failed. I&#39;ve just found 10,000 ways that won&#39;t work.”</span>\n <span>by <small class="author" itemprop="author">Thomas A. Edison</small>\n <a href="/author/Thomas-A-Edison">(about)</a>\n </span>\n <div class="tags">\n Tags:\n <meta class="keywords" itemprop="keywords" content="edison,failure,inspirational,paraphrased" / > \n \n <a class="tag" href="/tag/edison/page/1/">edison</a>\n \n <a class="tag" href="/tag/failure/page/1/">failure</a>\n \n <a class="tag" href="/tag/inspirational/page/1/">inspirational</a>\n \n <a class="tag" href="/tag/paraphrased/page/1/">paraphrased</a>\n \n </div>\n </div>\n\n <div class="quote" itemscope itemtype="http://schema.org/CreativeWork">\n <span class="text" itemprop="text">“A woman is like a tea bag; you never know how strong it is until it&#39;s in hot water.”</span>\n <span>by <small class="author" itemprop="author">Eleanor Roosevelt</small>\n <a href="/author/Eleanor-Roosevelt">(about)</a>\n </span>\n <div class="tags">\n Tags:\n <meta class="keywords" itemprop="keywords" content="misattributed-eleanor-roosevelt" / > \n \n <a class="tag" href="/tag/misattributed-eleanor-roosevelt/page/1/">misattributed-eleanor-roosevelt</a>\n \n </div>\n </div>\n\n <div class="quote" itemscope itemtype="http://schema.org/CreativeWork">\n <span class="text" itemprop="text">“A day without sunshine is like, you know, night.”</span>\n <span>by <small class="author" itemprop="author">Steve Martin</small>\n <a href="/author/Steve-Martin">(about)</a>\n </span>\n <div class="tags">\n Tags:\n <meta class="keywords" itemprop="keywords" content="humor,obvious,simile" / > \n \n <a class="tag" href="/tag/humor/page/1/">humor</a>\n \n <a class="tag" href="/tag/obvious/page/1/">obvious</a>\n \n <a class="tag" href="/tag/simile/page/1/">simile</a>\n \n </div>\n </div>\n\n <nav>\n <ul class="pager">\n \n \n <li class="next">\n <a href="/page/2/">Next <span aria-hidden="true">&rarr;</span></a>\n </li>\n \n </ul>\n </nav>\n </div>\n <div class="col-md-4 tags-box">\n \n <h2>Top Ten tags</h2>\n \n <span class="tag-item">\n <a class="tag" style="font-size: 28px" href="/tag/love/">love</a>\n </span>\n \n <span class="tag-item">\n <a class="tag" style="font-size: 26px" href="/tag/inspirational/">inspirational</a>\n </span>\n \n <span class="tag-item">\n <a class="tag" style="font-size: 26px" href="/tag/life/">life</a>\n </span>\n \n <span class="tag-item">\n <a class="tag" style="font-size: 24px" href="/tag/humor/">humor</a>\n </span>\n \n <span class="tag-item">\n <a class="tag" style="font-size: 22px" href="/tag/books/">books</a>\n </span>\n \n <span class="tag-item">\n <a class="tag" style="font-size: 14px" href="/tag/reading/">reading</a>\n </span>\n \n <span class="tag-item">\n <a class="tag" style="font-size: 10px" href="/tag/friendship/">friendship</a>\n </span>\n \n <span class="tag-item">\n <a class="tag" style="font-size: 8px" href="/tag/friends/">friends</a>\n </span>\n \n <span class="tag-item">\n <a class="tag" style="font-size: 8px" href="/tag/truth/">truth</a>\n </span>\n \n <span class="tag-item">\n <a class="tag" style="font-size: 6px" href="/tag/simile/">simile</a>\n </span>\n \n \n </div>\n</div>\n\n </div>\n <footer class="footer">\n <div class="container">\n <p class="text-muted">\n Quotes by: <a href="https://www.goodreads.com/quotes">GoodReads.com</a>\n </p>\n <p class="copyright">\n Made with <span class=\'sh-red\'>❤</span> by <a href="https://scrapinghub.com">Scrapinghub</a>\n </p>\n </div>\n </footer>\n</body>\n</html>'

**TASK: Get the names of all the authors on the first page.**

# CODE HERE

soup = bs4.BeautifulSoup(res.text,'lxml')

soup.select(".author")

[<small class="author" itemprop="author">Albert Einstein</small>,  
 <small class="author" itemprop="author">J.K. Rowling</small>,  
 <small class="author" itemprop="author">Albert Einstein</small>,  
 <small class="author" itemprop="author">Jane Austen</small>,  
 <small class="author" itemprop="author">Marilyn Monroe</small>,  
 <small class="author" itemprop="author">Albert Einstein</small>,  
 <small class="author" itemprop="author">André Gide</small>,  
 <small class="author" itemprop="author">Thomas A. Edison</small>,  
 <small class="author" itemprop="author">Eleanor Roosevelt</small>,  
 <small class="author" itemprop="author">Steve Martin</small>]

# I used a set to not worry about repeat authors.  
authors = set()   
for name in soup.select(".author"):  
 authors.add(name.text)

authors

{'Albert Einstein',  
 'André Gide',  
 'Eleanor Roosevelt',  
 'J.K. Rowling',  
 'Jane Austen',  
 'Marilyn Monroe',  
 'Steve Martin',  
 'Thomas A. Edison'}

**TASK: Create a list of all the quotes on the first page.**

#CODE HERE

quotes = []  
for quote in soup.select(".text"):  
 quotes.append(quote.text)

quotes

['“The world as we have created it is a process of our thinking. It cannot be changed without changing our thinking.”',  
 '“It is our choices, Harry, that show what we truly are, far more than our abilities.”',  
 '“There are only two ways to live your life. One is as though nothing is a miracle. The other is as though everything is a miracle.”',  
 '“The person, be it gentleman or lady, who has not pleasure in a good novel, must be intolerably stupid.”',  
 "“Imperfection is beauty, madness is genius and it's better to be absolutely ridiculous than absolutely boring.”",  
 '“Try not to become a man of success. Rather become a man of value.”',  
 '“It is better to be hated for what you are than to be loved for what you are not.”',  
 "“I have not failed. I've just found 10,000 ways that won't work.”",  
 "“A woman is like a tea bag; you never know how strong it is until it's in hot water.”",  
 '“A day without sunshine is like, you know, night.”']

**TASK: Inspect the site and use Beautiful Soup to extract the top ten tags from the requests text shown on the top right from the home page (e.g Love,Inspirational,Life, etc…). HINT: Keep in mind there are also tags underneath each quote, try to find a class only present in the top right tags, perhaps check the span.**

# CODE HERE

soup = bs4.BeautifulSoup(res.text,'lxml')

soup.select('.tag-item')

[<span class="tag-item">  
 <a class="tag" href="/tag/love/" style="font-size: 28px">love</a>  
 </span>, <span class="tag-item">  
 <a class="tag" href="/tag/inspirational/" style="font-size: 26px">inspirational</a>  
 </span>, <span class="tag-item">  
 <a class="tag" href="/tag/life/" style="font-size: 26px">life</a>  
 </span>, <span class="tag-item">  
 <a class="tag" href="/tag/humor/" style="font-size: 24px">humor</a>  
 </span>, <span class="tag-item">  
 <a class="tag" href="/tag/books/" style="font-size: 22px">books</a>  
 </span>, <span class="tag-item">  
 <a class="tag" href="/tag/reading/" style="font-size: 14px">reading</a>  
 </span>, <span class="tag-item">  
 <a class="tag" href="/tag/friendship/" style="font-size: 10px">friendship</a>  
 </span>, <span class="tag-item">  
 <a class="tag" href="/tag/friends/" style="font-size: 8px">friends</a>  
 </span>, <span class="tag-item">  
 <a class="tag" href="/tag/truth/" style="font-size: 8px">truth</a>  
 </span>, <span class="tag-item">  
 <a class="tag" href="/tag/simile/" style="font-size: 6px">simile</a>  
 </span>]

for item in soup.select(".tag-item"):  
 print(item.text)

love  
  
  
inspirational  
  
  
life  
  
  
humor  
  
  
books  
  
  
reading  
  
  
friendship  
  
  
friends  
  
  
truth  
  
  
simile

**TASK: Notice how there is more than one page, and subsequent pages look like this http://quotes.toscrape.com/page/2/. Use what you know about for loops and string concatenation to loop through all the pages and get all the unique authors on the website. Keep in mind there are many ways to achieve this, also note that you will need to somehow figure out how to check that your loop is on the last page with quotes. For debugging purposes, I will let you know that there are only 10 pages, so the last page is http://quotes.toscrape.com/page/10/, but try to create a loop that is robust enough that it wouldn’t matter to know the amount of pages beforehand, perhaps use try/except for this, its up to you!**

# CODE HERE

### Possible Solution #1 ( Assuming You Know Number of Pages)

url = 'http://quotes.toscrape.com/page/'

authors = set()  
  
for page in range(1,10):  
  
 # Concatenate to get new page URL  
 page\_url = url+str(page)  
 # Obtain Request  
 res = requests.get(page\_url)  
 # Turn into Soup  
 soup = bs4.BeautifulSoup(res.text,'lxml')  
 # Add Authors to our set  
 for name in soup.select(".author"):  
 authors.add(name.text)

### Possible Solution #2 ( Unknown Number of Pages, but knowledge of last page)

Let’s check what the last invalid page looks like:

# Choose some huge page number we know doesn't exist  
page\_url = url+str(9999999)  
  
# Obtain Request  
res = requests.get(page\_url)  
  
# Turn into Soup  
soup = bs4.BeautifulSoup(res.text,'lxml')

soup

<!DOCTYPE html>  
<html lang="en">  
<head>  
<meta charset="utf-8"/>  
<title>Quotes to Scrape</title>  
<link href="/static/bootstrap.min.css" rel="stylesheet"/>  
<link href="/static/main.css" rel="stylesheet"/>  
</head>  
<body>  
<div class="container">  
<div class="row header-box">  
<div class="col-md-8">  
<h1>  
<a href="/" style="text-decoration: none">Quotes to Scrape</a>  
</h1>  
</div>  
<div class="col-md-4">  
<p>  
<a href="/login">Login</a>  
</p>  
</div>  
</div>  
<div class="row">  
<div class="col-md-8">  
  
No quotes found!  
  
 <nav>  
<ul class="pager">  
<li class="previous">  
<a href="/page/9999998/"><span aria-hidden="true">←</span> Previous</a>  
</li>  
</ul>  
</nav>  
</div>  
<div class="col-md-4 tags-box">  
<h2>Top Ten tags</h2>  
<span class="tag-item">  
<a class="tag" href="/tag/love/" style="font-size: 28px">love</a>  
</span>  
<span class="tag-item">  
<a class="tag" href="/tag/inspirational/" style="font-size: 26px">inspirational</a>  
</span>  
<span class="tag-item">  
<a class="tag" href="/tag/life/" style="font-size: 26px">life</a>  
</span>  
<span class="tag-item">  
<a class="tag" href="/tag/humor/" style="font-size: 24px">humor</a>  
</span>  
<span class="tag-item">  
<a class="tag" href="/tag/books/" style="font-size: 22px">books</a>  
</span>  
<span class="tag-item">  
<a class="tag" href="/tag/reading/" style="font-size: 14px">reading</a>  
</span>  
<span class="tag-item">  
<a class="tag" href="/tag/friendship/" style="font-size: 10px">friendship</a>  
</span>  
<span class="tag-item">  
<a class="tag" href="/tag/friends/" style="font-size: 8px">friends</a>  
</span>  
<span class="tag-item">  
<a class="tag" href="/tag/truth/" style="font-size: 8px">truth</a>  
</span>  
<span class="tag-item">  
<a class="tag" href="/tag/simile/" style="font-size: 6px">simile</a>  
</span>  
</div>  
</div>  
</div>  
<footer class="footer">  
<div class="container">  
<p class="text-muted">  
 Quotes by: <a href="https://www.goodreads.com/quotes">GoodReads.com</a>  
</p>  
<p class="copyright">  
 Made with <span class="sh-red">❤</span> by <a href="https://scrapinghub.com">Scrapinghub</a>  
</p>  
</div>  
</footer>  
</body>  
</html>

# This solution requires that the string "No quotes found!" only occurs on the last page.  
# If for some reason this string was on the other pages, we would need to be more detailed.  
"No quotes found!" in res.text

True

page\_still\_valid = True  
authors = set()  
page = 1  
  
while page\_still\_valid:  
  
 # Concatenate to get new page URL  
 page\_url = url+str(page)  
   
 # Obtain Request  
 res = requests.get(page\_url)  
   
 # Check to see if we're on the last page  
 if "No quotes found!" in res.text:  
 break  
   
 # Turn into Soup  
 soup = bs4.BeautifulSoup(res.text,'lxml')  
   
 # Add Authors to our set  
 for name in soup.select(".author"):  
 authors.add(name.text)  
   
 # Go to Next Page  
 page += 1

authors

{'Albert Einstein',  
 'Alexandre Dumas fils',  
 'Alfred Tennyson',  
 'Allen Saunders',  
 'André Gide',  
 'Ayn Rand',  
 'Bob Marley',  
 'C.S. Lewis',  
 'Charles Bukowski',  
 'Charles M. Schulz',  
 'Douglas Adams',  
 'Dr. Seuss',  
 'E.E. Cummings',  
 'Eleanor Roosevelt',  
 'Elie Wiesel',  
 'Ernest Hemingway',  
 'Friedrich Nietzsche',  
 'Garrison Keillor',  
 'George Bernard Shaw',  
 'George Carlin',  
 'George Eliot',  
 'George R.R. Martin',  
 'Harper Lee',  
 'Haruki Murakami',  
 'Helen Keller',  
 'J.D. Salinger',  
 'J.K. Rowling',  
 'J.M. Barrie',  
 'J.R.R. Tolkien',  
 'James Baldwin',  
 'Jane Austen',  
 'Jim Henson',  
 'Jimi Hendrix',  
 'John Lennon',  
 'Jorge Luis Borges',  
 'Khaled Hosseini',  
 "Madeleine L'Engle",  
 'Marilyn Monroe',  
 'Mark Twain',  
 'Martin Luther King Jr.',  
 'Mother Teresa',  
 'Pablo Neruda',  
 'Ralph Waldo Emerson',  
 'Stephenie Meyer',  
 'Steve Martin',  
 'Suzanne Collins',  
 'Terry Pratchett',  
 'Thomas A. Edison',  
 'W.C. Fields',  
 'William Nicholson'}

There are lots of other potential solutions that are even more robust and flexible, the main idea is the same though, use a while loop to cycle through potential pages and have a break condition based on the invalid page.