

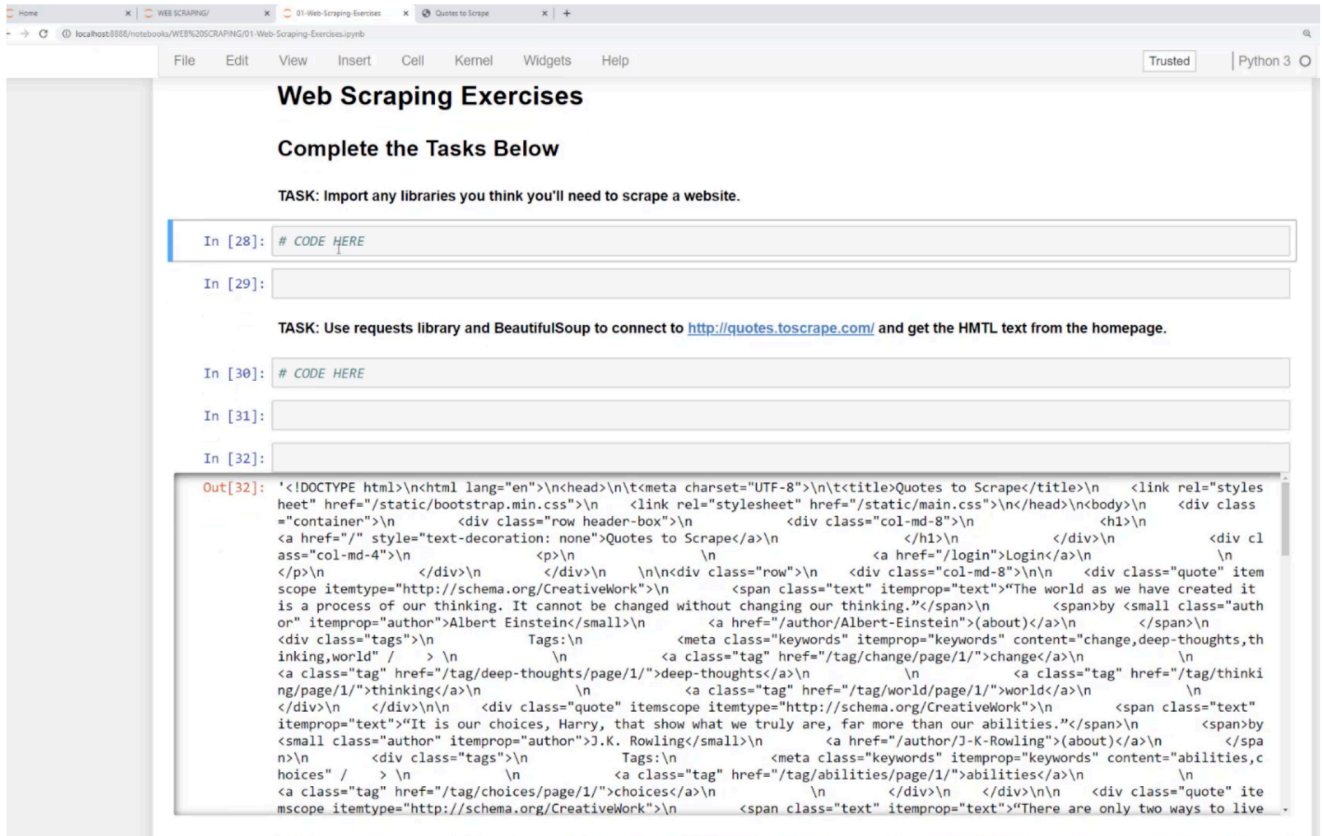
SECTION 15: WEB SCRAPING WITH PYTHON - 1 hour 40 minutes, 9 parts

7/9 Python Web Scraping - Book Examples Part Two

- iterating through the different books on the webpage to extract information

- context

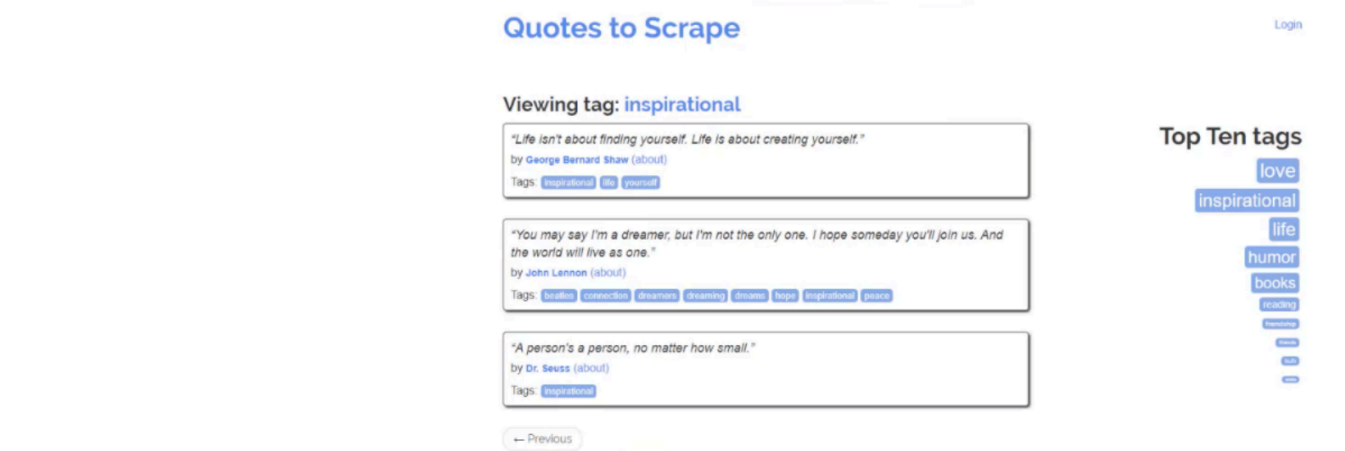
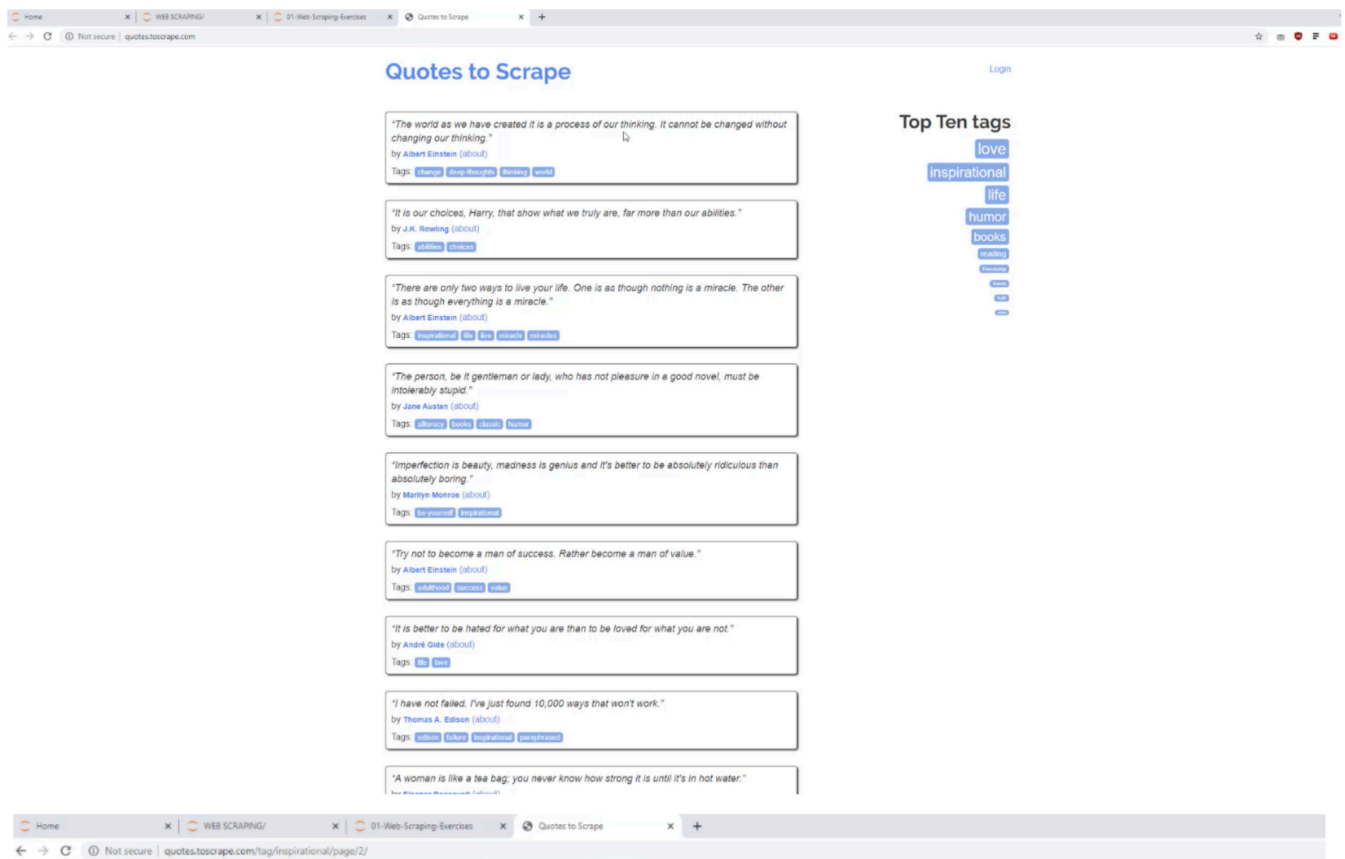
- -> he's scraped the html for one of the webpages and put it into a cell in an ipynb file
- -> this is information for the price of books
- -> if the class is associated with a two star rating
- -> grabbing the titles of those books
- -> **there are multiple ways we can solve the same problem**
- -> we have an example object <- this is a beautiful soup object
- -> we can do this using a quick approach, or by using beautiful soup



The screenshot shows a Jupyter Notebook interface with the title "Web Scraping Exercises". The notebook contains several code cells. The first cell is labeled "In [28]: # CODE HERE". The second cell is labeled "In [29]:". Below these is a task instruction: "TASK: Use requests library and BeautifulSoup to connect to <http://quotes.toscrape.com/> and get the HTML text from the homepage." The third cell is labeled "In [30]: # CODE HERE". The fourth cell is labeled "In [31]:". The fifth cell is labeled "In [32]:". The output of the fifth cell is displayed as "Out[32]:" and shows a large block of HTML code, which is a snippet of the HTML from the quotes.toscrape.com website. The HTML code includes tags for the page header, navigation links, and a list of quotes.

- -> quick approach

- -> converting the scraped html into a string
- -> and then checking if text is in the html (which tells us if there are two stars for the book or not)
- -> this returns a boolean
- -> this won't always work



○ -> a more general approach

- -> if we are looking for a certain class
- -> e.g for a book object on the html page, the class which gives the book two stars
- -> example.select <- then the argument of this in this case is searching for a piece of css / html (a class)
- -> this will return an empty list if it's not correct
- -> he's defined a boolean which checks if this is an empty list or not
- -> then he's grabbing the title
- -> he's printed out an example
- -> there is an alt on one of the <h3> tags
- -> he is extracting / targeting the information on the webpage
- -> it's still a beautiful soup tag
- -> then extracting a book title from it
- -> once we have extracted information for one of the books, we can repeat this for all of them with a for loop
 - -> he is using .format for this <- this is a common one
 - -> storing the results in variables

- -> we have lists of different books, and are iterating through them (to filter them)
- -> when iterating through the list of books, he is applying a condition to each
- -> we are extracting information in a string check
- -> the string example
- -> **if we are scraping information from a site, sometimes a firewall can block us if we make too many repeated requests**
- -> he's iterated through all of the 51 pages of the books
- -> then extracted all of the information, and filtered for certain information
- -> **he's scraped all of the books (from the html page), then iterated through them and filtered for the ones with less than two stars**
- -> **ask for permission before scraping sites**
- -> some sites try and stop people from scraping
- -> they also block you from making repeated requests

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

```
In [33]: # CODE HERE
```

```
In [37]: authors
```

```
Out[37]: {'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.

```
In [13]: #CODE HERE
```

```
In [15]: quotes
```

```
Out[15]: ["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.",
```

```
J.K. Rowling',  
'Jane Austen',  
'Marilyn Monroe',  
'Steve Martin',  
'Thomas A. Edison']
```

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

In [13]: #CODE HERE

In [15]: quotes

```
Out[15]: ["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.

In [16]: # CODE HERE

In [19]:

love

inspirational

life

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!

In [22]: # CODE HERE

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.

