Python les-materialen

Guide to Web Scraping

Let's get you started with web scraping and Python. Before we begin, here are some important rules to follow and understand:

1. Always be respectful and try to get premission to scrape, do not bombard a website with scraping requests, otherwise your IP address may be blocked!
2. Be aware that websites change often, meaning your code could go from working to totally broken from one day to the next.
3. Pretty much every web scraping project of interest is a unique and custom job, so try your best to generalize the skills learned here.

OK, let's get started with the basics!

Basic components of a WebSite

HTML

HTML stands for Hypertext Markup Language and every website on the internet uses it to display information. Even the jupyter notebook system uses it to display this information in your browser. If you right click on a website and select "View Page Source" you can see the raw HTML of a web page. This is the information that Python will be looking at to grab information from. Let's take a look at a simple webpage's HTML:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Title on Browser Tab</title>
    </head>
    <body>
        <h1> Website Header </h1>
        <p> Some Paragraph </p>
    <body>
</html>
```

Let's breakdown these components.

Every indicates a specific block type on the webpage:

```
1.<DOCTYPE html> HTML documents will always start with this type declaration, letting the br
2. The component blocks of the HTML document are placed between <html> and </html>.
3. Meta data and script connections (like a link to a CSS file or a JS file) are often place
4. The <title> tag block defines the title of the webpage (its what shows up in the tab of a
5. Is between <body> and </body> tags are the blocks that will be visible to the site visito
6. Headings are defined by the <h1> through <h6> tags, where the number represents the size
7. Paragraphs are defined by the <p> tag, this is essentially just normal text on the websit
```

```
There are many more tags than just these, such as <a> for hyperlinks, <table> for tables, <t
```

CSS

CSS stands for Cascading Style Sheets, this is what gives “style” to a website, including colors and fonts, and even some animations! CSS uses tags such as **id** or **class** to connect an HTML element to a CSS feature, such as a particular color. **id** is a unique id for an HTML tag and must be unique within the HTML document, basically a single use connection. **class** defines a general style that can then be linked to multiple HTML tags. Basically if you only want a single html tag to be red, you would use an id tag, if you wanted several HTML tags/blocks to be red, you would create a class in your CSS doc and then link it to the rest of these blocks.

Scraping Guidelines

Keep in mind you should always have permission for the website you are scraping! Check a websites terms and conditions for more info. Also keep in mind that a computer can send requests to a website very fast, so a website may block your computer’s ip address if you send too many requests too quickly. Lastly, websites change all the time! You will most likely need to update your code often for long term web-scraping jobs.

Web Scraping with Python

There are a few libraries you will need, you can go to your command line and install them with conda install (if you are using anaconda distribution), or pip install for other python distributions.

```
conda install requests
conda install lxml
conda install bs4
```

if you are not using the Anaconda Installation, you can use **pip install** instead of **conda install**, for example:

```
pip install requests
pip install lxml
pip install bs4
```

Now let’s see what we can do with these libraries.

Example Task 0 - Grabbing the title of a page

Let’s start very simple, we will grab the title of a page. Remember that this is the HTML block with the **title** tag. For this task we will use **www.example.com** which is a website specifically made to serve as an example domain. Let's go through the main steps:

```python
import requests
```

```python
# Step 1: Use the requests library to grab the page
# Note, this may fail if you have a firewall blocking Python/Jupyter
# Note sometimes you need to run this twice if it fails the first time
res = requests.get("http://www.example.com")
```

This object is a requests.models.Response object and it actually contains the information from the website, for example:

```python
type(res)
```

```
requests.models.Response
```

```python
res.text
```

```
'<!doctype html>\n<html>\n<head>\n    <title>Example Domain</title>\n\n    <meta charset="ut
```

Now we use BeautifulSoup to analyze the extracted page. Technically we could use our own custom script to loook for items in the string of **res.text** but the BeautifulSoup library already has lots of built-in tools and methods to grab information from a string of this nature (basically an HTML file). Using BeautifulSoup we can create a "soup" object that contains all the "ingredients" of the webpage. Don't ask me about the weird library names, I didn't choose them! :)

```python
import bs4
```

```python
soup = bs4.BeautifulSoup(res.text,"lxml")
```

```python
soup
```

```
<!DOCTYPE html>
<html>
<head>
<title>Example Domain</title>
<meta charset="utf-8"/>
<meta content="text/html; charset=utf-8" http-equiv="Content-type"/>
<meta content="width=device-width, initial-scale=1" name="viewport"/>
<style type="text/css">
    body {
        background-color: #f0f0f2;
        margin: 0;
        padding: 0;
        font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans",

    }
    div {
        width: 600px;
        margin: 5em auto;
        padding: 2em;
        background-color: #fdfdff;
        border-radius: 0.5em;
        box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
    }
    a:link, a:visited {
        color: #38488f;
        text-decoration: none;
    }
    @media (max-width: 700px) {
        div {
            margin: 0 auto;
            width: auto;
        }
    }
    </style>
</head>
<body>
<div>
<h1>Example Domain</h1>
<p>This domain is for use in illustrative examples in documents. You may use this
    domain in literature without prior coordination or asking for permission.</p>
<p><a href="https://www.iana.org/domains/example">More information...</a></p>
</div>
</body>
</html>
```

Now let's use the **.select()** method to grab elements. We are looking for the 'title' tag, so we will pass in 'title'

```
soup.select('title')
```

```
[<title>Example Domain</title>]
```

Notice what is returned here, its actually a list containing all the title elements (along with their tags). You can use indexing or even looping to grab the elements from the list. Since this object it still a specialized tag, we cna use method calls to grab just the text.

```
title_tag = soup.select('title')
```

```
title_tag[0]
```

```
<title>Example Domain</title>
```

```
type(title_tag[0])
```

```
bs4.element.Tag
```

```
title_tag[0].getText()
```

```
'Example Domain'
```

Example Task 1 - Grabbing all elements of a class

Let’s try to grab all the section headings of the Wikipedia Article on Grace Hopper from this URL: https://en.wikipedia.org/wiki/Grace_Hopper

```
# First get the request
res = requests.get('https://en.wikipedia.org/wiki/Grace_Hopper')
```

```
# Create a soup from request
soup = bs4.BeautifulSoup(res.text,"lxml")
```

Now its time to figure out what we are actually looking for. Inspect the element on the page to see that the section headers have the class “mw-headline”. Because this is a class and not a straight tag, we need to adhere to some syntax for CSS. In this case

Syntax to pass to the .select() method

Match Results

soup.select(‘div’)

All elements with the <div> tag

soup.select(‘#some_id’)

The HTML element containing the id attribute of some_id

soup.select(‘.notice’)

All the HTML elements with the CSS class named notice

soup.select(‘div span’)

Any elements named <span> that are within an element named <div>

soup.select(‘div > span’)

Any elements named <span> that are directly within an element named <div>, with no other element in between

```
# note depending on your IP Address,
# this class may be called something different
soup.select(".toctext")
```

```
[<span class="mw-headline" id="Early_life_and_education">Early life and education</span>,
 <span class="mw-headline" id="Career">Career</span>,
 <span class="mw-headline" id="World_War_II">World War II</span>,
 <span class="mw-headline" id="UNIVAC">UNIVAC</span>,
 <span class="mw-headline" id="COBOL">COBOL</span>,
 <span class="mw-headline" id="Standards">Standards</span>,
 <span class="mw-headline" id="Retirement">Retirement</span>,
 <span class="mw-headline" id="Post-retirement">Post-retirement</span>,
 <span class="mw-headline" id="Anecdotes">Anecdotes</span>,
 <span class="mw-headline" id="Death">Death</span>,
 <span class="mw-headline" id="Dates_of_rank">Dates of rank</span>,
 <span class="mw-headline" id="Awards_and_honors">Awards and honors</span>,
 <span class="mw-headline" id="Military_awards">Military awards</span>,
 <span class="mw-headline" id="Other_awards">Other awards</span>,
 <span class="mw-headline" id="Legacy">Legacy</span>,
 <span class="mw-headline" id="Places">Places</span>,
 <span class="mw-headline" id="Programs">Programs</span>,
 <span class="mw-headline" id="In_popular_culture">In popular culture</span>,
 <span class="mw-headline" id="Grace_Hopper_Celebration_of_Women_in_Computing">Grace Hopper
 <span class="mw-headline" id="Notes">Notes</span>,
 <span class="mw-headline" id="Obituary_notices">Obituary notices</span>,
 <span class="mw-headline" id="See_also">See also</span>,
 <span class="mw-headline" id="References">References</span>,
 <span class="mw-headline" id="Further_reading">Further reading</span>,
 <span class="mw-headline" id="External_links">External links</span>]
```

```
for item in soup.select(".toctext"):
    print(item.text)
```

```
Early life and education
Career
World War II
UNIVAC
COBOL
Standards
Retirement
Post-retirement
Anecdotes
Death
Dates of rank
Awards and honors
Military awards
Other awards
Legacy
Places
Programs
In popular culture
Grace Hopper Celebration of Women in Computing
Notes
Obituary notices
See also
```

```
References
Further reading
External links
```

Example Task 3 - Getting an Image from a Website

Let's attempt to grab the image of the Deep Blue Computer from this wikipedia article: https://en.wikipedia.org/wiki/Deep_Blue_(chess_computer)

```
res = requests.get("https://en.wikipedia.org/wiki/Deep_Blue_(chess_computer)")
```

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

```
image_info = soup.select('.thumbimage')
```

```
image_info
```

```
[<img alt="" class="thumbimage" data-file-height="601" data-file-width="400" decoding="async
 <img alt="" class="thumbimage" data-file-height="600" data-file-width="800" decoding="async
```

```
len(image_info)
```

```
2
```

```
computer = image_info[0]
```

```
type(computer)
```

```
bs4.element.Tag
```

You can make dictionary like calls for parts of the Tag, in this case, we are interested in the **src** , or "source" of the image, which should be its own .jpg or .png link:

```
computer['src']
```

```
'//upload.wikimedia.org/wikipedia/commons/thumb/b/be/Deep_Blue.jpg/220px-Deep_Blue.jpg'
```

We can actually display it with a markdown cell with the following:

```
<img src='https://upload.wikimedia.org/wikipedia/commons/thumb/b/be/Deep_Blue.jpg/220px-Deep
```

Now that you have the actual src link, you can grab the image with requests and get along with the .content attribute. Note how we had to add https:// before the link, if you don't do this, requests will complain (but it gives you a pretty descriptive error code).

```
image_link = requests.get('https://upload.wikimedia.org/wikipedia/commons/thumb/b/be/Deep_Bl
```

```
# The raw content (its a binary file, meaning we will need to use binary read/write methods
image_link.content
```

```
b'\xff\xd8\xff\xe0\x00\x10JFIF\x00\x01\x01\x01\x00H\x00H\x00\x00\xff\xfe\x00CFile source: ht
```

Let's write this to a file:=, not the 'wb' call to denote a binary writing of the file.

```
f = open('my_new_file_name.jpg','wb')
```

```
f.write(image_link.content)
```

```
16806
```

```
f.close()
```

Now we can display this file right here in the notebook as markdown using:

```
<img src="'my_new_file_name.jpg'>
```

Just write the above line in a new markdown cell and it will display the image we just downloaded!

Example Project - Working with Multiple Pages and Items

Let's show a more realistic example of scraping a full site. The website: http://books.toscrape.com/index.html is specifically designed for people to scrape it. Let's try to get the title of every book that has a 2 star rating and at the end just have a Python list with all their titles.

We will do the following:

1. Figure out the URL structure to go through every page
2. Scrap every page in the catalogue
3. Figure out what tag/class represents the Star rating
4. Filter by that star rating using an if statement
5. Store the results to a list

We can see that the URL structure is the following:

```
http://books.toscrape.com/catalogue/page-1.html
```

```
base_url = 'http://books.toscrape.com/catalogue/page-{}.html'
```

We can then fill in the page number with .format()

```
res = requests.get(base_url.format('1'))
```

Now let's grab the products (books) from the get request result:

```
soup = bs4.BeautifulSoup(res.text,"lxml")
```

```
soup.select(".product_pod")
```

```
[<article class="product_pod">
 <div class="image_container">
 <a href="a-light-in-the-attic_1000/index.html"><img alt="A Light in the Attic" class="thumb
 </div>
 <p class="star-rating Three">
 <i class="icon-star"></i>
 <i class="icon-star"></i>
 <i class="icon-star"></i>
```

```
<i class="icon-star"></i>
<i class="icon-star"></i>
</p>
<h3><a href="a-light-in-the-attic_1000/index.html" title="A Light in the Attic">A Light in
<div class="product_price">
<p class="price_color">Â£51.77</p>
<p class="instock availability">
<i class="icon-ok"></i>

        In stock

</p>
<form>
<button class="btn btn-primary btn-block" data-loading-text="Adding..." type="submit">Add t
</form>
</div>
</article>, <article class="product_pod">
<div class="image_container">
<a href="tipping-the-velvet_999/index.html"><img alt="Tipping the Velvet" class="thumbnail"
</div>
<p class="star-rating One">
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
</p>
<h3><a href="tipping-the-velvet_999/index.html" title="Tipping the Velvet">Tipping the Velv
<div class="product_price">
<p class="price_color">Â£53.74</p>
<p class="instock availability">
<i class="icon-ok"></i>

        In stock

</p>
<form>
<button class="btn btn-primary btn-block" data-loading-text="Adding..." type="submit">Add t
</form>
</div>
</article>, <article class="product_pod">
<div class="image_container">
<a href="soumission_998/index.html"><img alt="Soumission" class="thumbnail" src="../media/c
</div>
<p class="star-rating One">
<i class="icon-star"></i>
```

```
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
</p>
<h3><a href="soumission_998/index.html" title="Soumission">Soumission</a></h3>
<div class="product_price">
<p class="price_color">Â£50.10</p>
<p class="instock availability">
<i class="icon-ok"></i>

        In stock

</p>
<form>
<button class="btn btn-primary btn-block" data-loading-text="Adding..." type="submit">Add t
</form>
</div>
</article>, <article class="product_pod">
<div class="image_container">
<a href="sharp-objects_997/index.html"><img alt="Sharp Objects" class="thumbnail" src="../
</div>
<p class="star-rating Four">
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
</p>
<h3><a href="sharp-objects_997/index.html" title="Sharp Objects">Sharp Objects</a></h3>
<div class="product_price">
<p class="price_color">Â£47.82</p>
<p class="instock availability">
<i class="icon-ok"></i>

        In stock

</p>
<form>
<button class="btn btn-primary btn-block" data-loading-text="Adding..." type="submit">Add t
</form>
</div>
</article>, <article class="product_pod">
<div class="image_container">
<a href="sapiens-a-brief-history-of-humankind_996/index.html"><img alt="Sapiens: A Brief Hi
</div>
```

```
<p class="star-rating Five">
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
</p>
<h3><a href="sapiens-a-brief-history-of-humankind_996/index.html" title="Sapiens: A Brief H
<div class="product_price">
<p class="price_color">Â£54.23</p>
<p class="instock availability">
<i class="icon-ok"></i>

        In stock

</p>
<form>
<button class="btn btn-primary btn-block" data-loading-text="Adding..." type="submit">Add t
</form>
</div>
</article>, <article class="product_pod">
<div class="image_container">
<a href="the-requiem-red_995/index.html"><img alt="The Requiem Red" class="thumbnail" src="
</div>
<p class="star-rating One">
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
</p>
<h3><a href="the-requiem-red_995/index.html" title="The Requiem Red">The Requiem Red</a></h
<div class="product_price">
<p class="price_color">Â£22.65</p>
<p class="instock availability">
<i class="icon-ok"></i>

        In stock

</p>
<form>
<button class="btn btn-primary btn-block" data-loading-text="Adding..." type="submit">Add t
</form>
</div>
</article>, <article class="product_pod">
<div class="image_container">
```

```
<a href="the-dirty-little-secrets-of-getting-your-dream-job_994/index.html"><img alt="The D
</div>
<p class="star-rating Four">
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
</p>
<h3><a href="the-dirty-little-secrets-of-getting-your-dream-job_994/index.html" title="The
<div class="product_price">
<p class="price_color">Â£33.34</p>
<p class="instock availability">
<i class="icon-ok"></i>

        In stock

</p>
<form>
<button class="btn btn-primary btn-block" data-loading-text="Adding..." type="submit">Add t
</form>
</div>
</article>, <article class="product_pod">
<div class="image_container">
<a href="the-coming-woman-a-novel-based-on-the-life-of-the-infamous-feminist-victoria-woodh
</div>
<p class="star-rating Three">
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
</p>
<h3><a href="the-coming-woman-a-novel-based-on-the-life-of-the-infamous-feminist-victoria-w
<div class="product_price">
<p class="price_color">Â£17.93</p>
<p class="instock availability">
<i class="icon-ok"></i>

        In stock

</p>
<form>
<button class="btn btn-primary btn-block" data-loading-text="Adding..." type="submit">Add t
</form>
</div>
```

```
</article>, <article class="product_pod">
<div class="image_container">
<a href="the-boys-in-the-boat-nine-americans-and-their-epic-quest-for-gold-at-the-1936-berl
</div>
<p class="star-rating Four">
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
</p>
<h3><a href="the-boys-in-the-boat-nine-americans-and-their-epic-quest-for-gold-at-the-1936-
<div class="product_price">
<p class="price_color">Â£22.60</p>
<p class="instock availability">
<i class="icon-ok"></i>

        In stock

</p>
<form>
<button class="btn btn-primary btn-block" data-loading-text="Adding..." type="submit">Add t
</form>
</div>
</article>, <article class="product_pod">
<div class="image_container">
<a href="the-black-maria_991/index.html"><img alt="The Black Maria" class="thumbnail" src="
</div>
<p class="star-rating One">
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
</p>
<h3><a href="the-black-maria_991/index.html" title="The Black Maria">The Black Maria</a></h
<div class="product_price">
<p class="price_color">Â£52.15</p>
<p class="instock availability">
<i class="icon-ok"></i>

        In stock

</p>
<form>
<button class="btn btn-primary btn-block" data-loading-text="Adding..." type="submit">Add t
```

```
</form>
</div>
</article>, <article class="product_pod">
<div class="image_container">
<a href="starving-hearts-triangular-trade-trilogy-1_990/index.html"><img alt="Starving Hear
</div>
<p class="star-rating Two">
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
</p>
<h3><a href="starving-hearts-triangular-trade-trilogy-1_990/index.html" title="Starving Hea
<div class="product_price">
<p class="price_color">Â£13.99</p>
<p class="instock availability">
<i class="icon-ok"></i>

        In stock

</p>
<form>
<button class="btn btn-primary btn-block" data-loading-text="Adding..." type="submit">Add t
</form>
</div>
</article>, <article class="product_pod">
<div class="image_container">
<a href="shakespeares-sonnets_989/index.html"><img alt="Shakespeare's Sonnets" class="thumb
</div>
<p class="star-rating Four">
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
</p>
<h3><a href="shakespeares-sonnets_989/index.html" title="Shakespeare's Sonnets">Shakespeare
<div class="product_price">
<p class="price_color">Â£20.66</p>
<p class="instock availability">
<i class="icon-ok"></i>

        In stock

</p>
```

```
<form>
<button class="btn btn-primary btn-block" data-loading-text="Adding..." type="submit">Add t
</form>
</div>
</article>, <article class="product_pod">
<div class="image_container">
<a href="set-me-free_988/index.html"><img alt="Set Me Free" class="thumbnail" src="../medi
</div>
<p class="star-rating Five">
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
</p>
<h3><a href="set-me-free_988/index.html" title="Set Me Free">Set Me Free</a></h3>
<div class="product_price">
<p class="price_color">Â£17.46</p>
<p class="instock availability">
<i class="icon-ok"></i>

        In stock

</p>
<form>
<button class="btn btn-primary btn-block" data-loading-text="Adding..." type="submit">Add t
</form>
</div>
</article>, <article class="product_pod">
<div class="image_container">
<a href="scott-pilgrims-precious-little-life-scott-pilgrim-1_987/index.html"><img alt="Scot
</div>
<p class="star-rating Five">
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
</p>
<h3><a href="scott-pilgrims-precious-little-life-scott-pilgrim-1_987/index.html" title="Sco
<div class="product_price">
<p class="price_color">Â£52.29</p>
<p class="instock availability">
<i class="icon-ok"></i>

        In stock
```

```
</p>
<form>
<button class="btn btn-primary btn-block" data-loading-text="Adding..." type="submit">Add t
</form>
</div>
</article>, <article class="product_pod">
<div class="image_container">
<a href="rip-it-up-and-start-again_986/index.html"><img alt="Rip it Up and Start Again" cla
</div>
<p class="star-rating Five">
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
</p>
<h3><a href="rip-it-up-and-start-again_986/index.html" title="Rip it Up and Start Again">Ri
<div class="product_price">
<p class="price_color">Â£35.02</p>
<p class="instock availability">
<i class="icon-ok"></i>

        In stock

</p>
<form>
<button class="btn btn-primary btn-block" data-loading-text="Adding..." type="submit">Add t
</form>
</div>
</article>, <article class="product_pod">
<div class="image_container">
<a href="our-band-could-be-your-life-scenes-from-the-american-indie-underground-1981-1991_9
</div>
<p class="star-rating Three">
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
</p>
<h3><a href="our-band-could-be-your-life-scenes-from-the-american-indie-underground-1981-19
<div class="product_price">
<p class="price_color">Â£57.25</p>
<p class="instock availability">
<i class="icon-ok"></i>
```

```
        In stock

</p>
<form>
<button class="btn btn-primary btn-block" data-loading-text="Adding..." type="submit">Add t
</form>
</div>
</article>, <article class="product_pod">
<div class="image_container">
<a href="olio_984/index.html"><img alt="Olio" class="thumbnail" src="../media/cache/55/33/5
</div>
<p class="star-rating One">
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
</p>
<h3><a href="olio_984/index.html" title="Olio">Olio</a></h3>
<div class="product_price">
<p class="price_color">Â£23.88</p>
<p class="instock availability">
<i class="icon-ok"></i>

        In stock

</p>
<form>
<button class="btn btn-primary btn-block" data-loading-text="Adding..." type="submit">Add t
</form>
</div>
</article>, <article class="product_pod">
<div class="image_container">
<a href="mesaerion-the-best-science-fiction-stories-1800-1849_983/index.html"><img alt="Mes
</div>
<p class="star-rating One">
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
</p>
<h3><a href="mesaerion-the-best-science-fiction-stories-1800-1849_983/index.html" title="Me
<div class="product_price">
<p class="price_color">Â£37.59</p>
```

```
<p class="instock availability">
<i class="icon-ok"></i>

        In stock

</p>
<form>
<button class="btn btn-primary btn-block" data-loading-text="Adding..." type="submit">Add t
</form>
</div>
</article>, <article class="product_pod">
<div class="image_container">
<a href="libertarianism-for-beginners_982/index.html"><img alt="Libertarianism for Beginner
</div>
<p class="star-rating Two">
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
</p>
<h3><a href="libertarianism-for-beginners_982/index.html" title="Libertarianism for Beginne
<div class="product_price">
<p class="price_color">Â£51.33</p>
<p class="instock availability">
<i class="icon-ok"></i>

        In stock

</p>
<form>
<button class="btn btn-primary btn-block" data-loading-text="Adding..." type="submit">Add t
</form>
</div>
</article>, <article class="product_pod">
<div class="image_container">
<a href="its-only-the-himalayas_981/index.html"><img alt="It's Only the Himalayas" class="t
</div>
<p class="star-rating Two">
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
</p>
<h3><a href="its-only-the-himalayas_981/index.html" title="It's Only the Himalayas">It's On
```

```
 <div class="product_price">
 <p class="price_color">Â£45.17</p>
 <p class="instock availability">
 <i class="icon-ok"></i>

         In stock

 </p>
 <form>
 <button class="btn btn-primary btn-block" data-loading-text="Adding..." type="submit">Add t
 </form>
 </div>
 </article>]
```

Now we can see that each book has the product_pod class. We can select any tag with this class, and then further reduce it by its rating.

```
products = soup.select(".product_pod")
```

```
example = products[0]
```

```
type(example)
```

```
bs4.element.Tag
```

```
example.attrs
```

```
{'class': ['product_pod']}
```

Now by inspecting the site we can see that the class we want is class=‘star-rating Two’ , if you click on this in your browser, you’ll notice it displays the space as a . , so that means we want to search for “.star-rating.Two”

```
list(example.children)
```

```
['\n', <div class="image_container">
 <a href="a-light-in-the-attic_1000/index.html"><img alt="A Light in the Attic" class="thumb
 </div>, '\n', <p class="star-rating Three">
 <i class="icon-star"></i>
 <i class="icon-star"></i>
 <i class="icon-star"></i>
 <i class="icon-star"></i>
 <i class="icon-star"></i>
 </p>, '\n', <h3><a href="a-light-in-the-attic_1000/index.html" title="A Light in the Attic'
 <p class="price_color">Â£51.77</p>
 <p class="instock availability">
 <i class="icon-ok"></i>

         In stock

 </p>
 <form>
 <button class="btn btn-primary btn-block" data-loading-text="Adding..." type="submit">Add t
 </form>
 </div>, '\n']
```

```
example.select('.star-rating.Three')
```

```
[<p class="star-rating Three">
 <i class="icon-star"></i>
 <i class="icon-star"></i>
 <i class="icon-star"></i>
 <i class="icon-star"></i>
 <i class="icon-star"></i>
 </p>]
```

But we are looking for 2 stars, so it looks like we can just check to see if something was returned

```
example.select('.star-rating.Two')
```

```
[]
```

Alternatively, we can just quickly check the text string to see if “star-rating Two” is in it. Either approach is fine (there are also many other alternative approaches!)

Now let’s see how we can get the title if we have a 2-star match:

```
example.select('a')
```

```
[<a href="a-light-in-the-attic_1000/index.html"><img alt="A Light in the Attic" class="thumb
 <a href="a-light-in-the-attic_1000/index.html" title="A Light in the Attic">A Light in the
```

```
example.select('a')[1]
```

```
<a href="a-light-in-the-attic_1000/index.html" title="A Light in the Attic">A Light in the
```

```
example.select('a')[1]['title']
```

```
'A Light in the Attic'
```

Okay, let’s give it a shot by combining all the ideas we’ve talked about! (this should take about 20-60 seconds to complete running. Be aware a firwall may prevent this script from running. Also if you are getting a no response error, maybe try adding a sleep step with time.sleep(1).

```
two_star_titles = []

for n in range(1,51):

    scrape_url = base_url.format(n)
    res = requests.get(scrape_url)
    soup = bs4.BeautifulSoup(res.text,"lxml")
    books = soup.select(".product_pod")

    for book in books:
        if len(book.select('.star-rating.Two')) != 0:
            two_star_titles.append(book.select('a')[1]['title'])
```

```
two_star_titles
```

```
['Starving Hearts (Triangular Trade Trilogy, #1)',
 'Libertarianism for Beginners',
 "It's Only the Himalayas",
 'How Music Works',
 'Maude (1883-1993):She Grew Up with the country',
 "You can't bury them all: Poems",
 'Reasons to Stay Alive',
 'Without Borders (Wanderlove #1)',
 'Soul Reader',
 'Security',
 'Saga, Volume 5 (Saga (Collected Editions) #5)',
 'Reskilling America: Learning to Labor in the Twenty-First Century',
 'Political Suicide: Missteps, Peccadilloes, Bad Calls, Backroom Hijinx, Sordid Pasts, Rotte
 'Obsidian (Lux #1)',
 'My Paris Kitchen: Recipes and Stories',
 'Masks and Shadows',
 'Lumberjanes, Vol. 2: Friendship to the Max (Lumberjanes #5-8)',
 'Lumberjanes Vol. 3: A Terrible Plan (Lumberjanes #9-12)',
 'Judo: Seven Steps to Black Belt (an Introductory Guide for Beginners)',
 'I Hate Fairyland, Vol. 1: Madly Ever After (I Hate Fairyland (Compilations) #1-5)',
 'Giant Days, Vol. 2 (Giant Days #5-8)',
 'Everydata: The Misinformation Hidden in the Little Data You Consume Every Day',
 "Don't Be a Jerk: And Other Practical Advice from Dogen, Japan's Greatest Zen Master",
 'Bossypants',
 'Bitch Planet, Vol. 1: Extraordinary Machine (Bitch Planet (Collected Editions))',
 'Avatar: The Last Airbender: Smoke and Shadow, Part 3 (Smoke and Shadow #3)',
 'Tuesday Nights in 1980',
 'The Psychopath Test: A Journey Through the Madness Industry',
 'The Power of Now: A Guide to Spiritual Enlightenment',
 "The Omnivore's Dilemma: A Natural History of Four Meals",
 'The Love and Lemons Cookbook: An Apple-to-Zucchini Celebration of Impromptu Cooking',
 'The Girl on the Train',
 'The Emerald Mystery',
 'The Argonauts',
 'Suddenly in Love (Lake Haven #1)',
 'Soft Apocalypse',
 "So You've Been Publicly Shamed",
 'Shoe Dog: A Memoir by the Creator of NIKE',
 'Louisa: The Extraordinary Life of Mrs. Adams',
 'Large Print Heart of the Pride',
 'Grumbles',
 'Chasing Heaven: What Dying Taught Me About Living',
 'Becoming Wise: An Inquiry into the Mystery and Art of Living',
 'Beauty Restored (Riley Family Legacy Novellas #3)',
 'Batman: The Long Halloween (Batman)',
 "Ayumi's Violin",
 'Wild Swans',
 "What's It Like in Space?: Stories from Astronauts Who've Been There",
 'Until Friday Night (The Field Party #1)',
 'Unbroken: A World War II Story of Survival, Resilience, and Redemption',
 'Twenty Yawns',
 'Through the Woods',
 'This Is Where It Ends',
 'The Year of Magical Thinking',
 'The Last Mile (Amos Decker #2)',
 'The Immortal Life of Henrietta Lacks',
 'The Hidden Oracle (The Trials of Apollo #1)',
 'The Guilty (Will Robie #4)',
 'Red Hood/Arsenal, Vol. 1: Open for Business (Red Hood/Arsenal #1)',
 'Once Was a Time',
 'No Dream Is Too High: Life Lessons From a Man Who Walked on the Moon',
 'Naruto (3-in-1 Edition), Vol. 14: Includes Vols. 40, 41 & 42 (Naruto: Omnibus #14)',
 'More Than Music (Chasing the Dream #1)',
 'Lowriders to the Center of the Earth (Lowriders in Space #2)',
 'Eat Fat, Get Thin',
 'Doctor Sleep (The Shining #2)',
 'Crazy Love: Overwhelmed by a Relentless God',
 'Carrie',
 'Batman: Europa',
 'Angels Walking (Angels Walking #1)',
 'Adulthood Is a Myth: A "Sarah\'s Scribbles" Collection',
 'A Study in Scarlet (Sherlock Holmes #1)',
 'A Series of Catastrophes and Miracles: A True Story of Love, Science, and Cancer',
 "A People's History of the United States",
 'My Kitchen Year: 136 Recipes That Saved My Life',
 'The Lonely City: Adventures in the Art of Being Alone',
 'The Dinner Party',
 'Stars Above (The Lunar Chronicles #4.5)',
 'Love, Lies and Spies',
 'Troublemaker: Surviving Hollywood and Scientology',
 'The Widow',
 'Setting the World on Fire: The Brief, Astonishing Life of St. Catherine of Siena',
 'Mothering Sunday',
 'Lilac Girls',
 '10% Happier: How I Tamed the Voice in My Head, Reduced Stress Without Losing My Edge, and
 'Underlying Notes',
 'The Flowers Lied',
 'Modern Day Fables',
 "Chernobyl 01:23:40: The Incredible True Story of the World's Worst Nuclear Disaster",
 '23 Degrees South: A Tropical Tale of Changing Whether...',
 'When Breath Becomes Air',
 'Vagabonding: An Uncommon Guide to the Art of Long-Term World Travel',
 'The Martian (The Martian #1)',
 "Miller's Valley",
 "Love That Boy: What Two Presidents, Eight Road Trips, and My Son Taught Me About a Parent'
 'Left Behind (Left Behind #1)',
 'Howl and Other Poems',
 "Heaven is for Real: A Little Boy's Astounding Story of His Trip to Heaven and Back",
 "Brazen: The Courage to Find the You That's Been Hiding",
 '32 Yolks',
 'Wildlife of New York: A Five-Borough Coloring Book',
 'Unreasonable Hope: Finding Faith in the God Who Brings Purpose to Your Pain',
 'The Art Book',
 'Steal Like an Artist: 10 Things Nobody Told You About Being Creative',
 'Raymie Nightingale',
 'Like Never Before (Walker Family #2)',
 'How to Be a Domestic Goddess: Baking and the Art of Comfort Cooking',
 'Finding God in the Ruins: How God Redeems Pain',
 'Chronicles, Vol. 1',
 'A Summer In Europe',
 'The Rise and Fall of the Third Reich: A History of Nazi Germany',
 'The Makings of a Fatherless Child',
 'The Fellowship of the Ring (The Lord of the Rings #1)',
 "Tell the Wolves I'm Home",
 'In the Woods (Dublin Murder Squad #1)',
 'Give It Back',
 'Why Save the Bankers?: And Other Essays on Our Economic and Political Crisis',
 'The Raven King (The Raven Cycle #4)',
 'The Expatriates',
 'The 5th Wave (The 5th Wave #1)',
 'Peak: Secrets from the New Science of Expertise',
 'Logan Kade (Fallen Crest High #5.5)',
 "I Know Why the Caged Bird Sings (Maya Angelou's Autobiography #1)",
 'Drama',
 "America's War for the Greater Middle East: A Military History",
 'A Game of Thrones (A Song of Ice and Fire #1)',
 "The Pilgrim's Progress",
 'The Hound of the Baskervilles (Sherlock Holmes #5)',
 "The Geography of Bliss: One Grump's Search for the Happiest Places in the World",
 'The Demonists (Demonist #1)',
 'The Demon Prince of Momochi House, Vol. 4 (The Demon Prince of Momochi House #4)',
 'Misery',
 'Far From True (Promise Falls Trilogy #2)',
 'Confessions of a Shopaholic (Shopaholic #1)',
 'Vegan Vegetarian Omnivore: Dinner for Everyone at the Table',
 'Two Boys Kissing',
 'Twilight (Twilight #1)',
 'Twenties Girl',
 'The Tipping Point: How Little Things Can Make a Big Difference',
 'The Stand',
 'The Picture of Dorian Gray',
 'The Name of God is Mercy',
 "The Lover's Dictionary",
 'The Last Painting of Sara de Vos',
 'The Guns of August',
 'The Girl Who Played with Fire (Millennium Trilogy #2)',
 'The Da Vinci Code (Robert Langdon #2)',
 'The Cat in the Hat (Beginner Books B-1)',
 'The Book Thief',
 'The Autobiography of Malcolm X',
 "Surely You're Joking, Mr. Feynman!: Adventures of a Curious Character",
 'Soldier (Talon #3)',
 'Shopaholic & Baby (Shopaholic #5)',
 'Seven Days in the Art World',
 'Rework',
 'Packing for Mars: The Curious Science of Life in the Void',
 'Orange Is the New Black',
 'One for the Money (Stephanie Plum #1)',
 'Midnight Riot (Peter Grant/ Rivers of London - books #1)',
 'Me Talk Pretty One Day',
 'Manuscript Found in Accra',
 'Lust & Wonder',
 "Life, the Universe and Everything (Hitchhiker's Guide to the Galaxy #3)",
 'Life After Life',
 'I Am Malala: The Girl Who Stood Up for Education and Was Shot by the Taliban',
 'House of Lost Worlds: Dinosaurs, Dynasties, and the Story of Life on Earth',
 'Horrible Bear!',
 'Holidays on Ice',
 'Girl in the Blue Coat',
 'Fruits Basket, Vol. 3 (Fruits Basket #3)',
 'Cosmos',
 'Civilization and Its Discontents',
 "Catastrophic Happiness: Finding Joy in Childhood's Messy Years",
 'Career of Evil (Cormoran Strike #3)',
 'Born to Run: A Hidden Tribe, Superathletes, and the Greatest Race the World Has Never Seen
 "Best of My Love (Fool's Gold #20)",
 'Beowulf',
 'Awkward',
 'And Then There Were None',
 'A Storm of Swords (A Song of Ice and Fire #3)',
 'The Suffragettes (Little Black Classics, #96)',
 'Vampire Girl (Vampire Girl #1)',
 'Three Wishes (River of Time: California #1)',
 'The Wicked + The Divine, Vol. 1: The Faust Act (The Wicked + The Divine)',
 'The Little Prince',
 'The Last Girl (The Dominion Trilogy #1)',
 'Taking Shots (Assassins #1)',
 'Settling the Score (The Summer Games #1)',
 'Rhythm, Chord & Malykhin',
 'One Second (Seven #7)',
 "Old Records Never Die: One Man's Quest for His Vinyl and His Past",
 'Of Mice and Men',
 'My Perfect Mistake (Over the Top #1)',
 'Meditations',
 'Frankenstein',
 'Emma']
```

** Excellent! You should now have the tools necessary to scrape any websites that interest you! Keep in mind, the more complex the website, the harder it will be to scrape. Always ask for permission! **