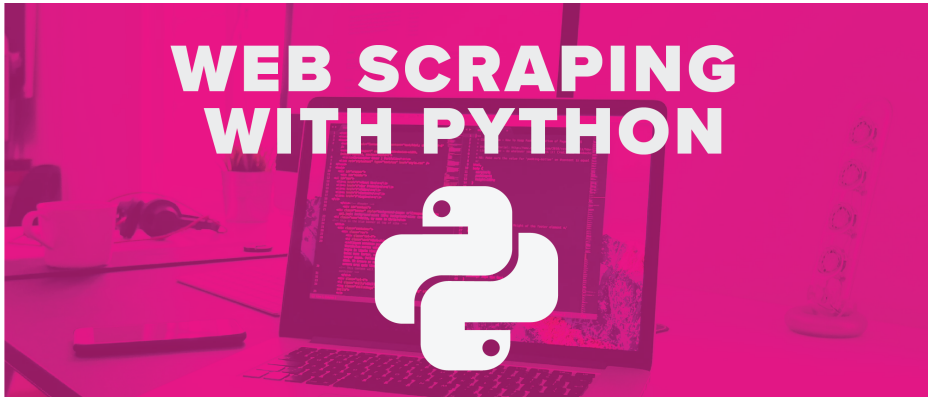


Web scraping using BeautifulSoup and Requests libraries (Rate My Professors Website)

Let's scrape the data from the web using Python.



Today we will be scraping "[Rate My Professor](#)" website. A little insight about Rate My Professor website, it is a website that contains a rating of school, professors and universities. You can search for any professor or school and get their ratings before taking or joining to their courses. It's a handy feature which helps to know more about your professor or the university that you want to join. In this tutorial, we shall see how to scrape and to extract a specific professor's tag. I warn you guys this is not illegal but the mass scraping of data from the website can lead your IP address being blocked. Just do it once or twice, but don't just foolishly put it in a loop and try to put request inside the loop.



What is Web Scraping ?

Web Scraping (also termed as Scraping, Data Extraction, Data Harvesting, etc.) is a technique used to extract data from the websites. Sometimes web scraping can be very useful wherein we can get the data that we are looking for straight from the web, but sometimes it's a bad way to do it, because it's like stealing the precious data from the website without their permission, but limit your scraping process to once or twice so that this can avoid you from falling in trouble.

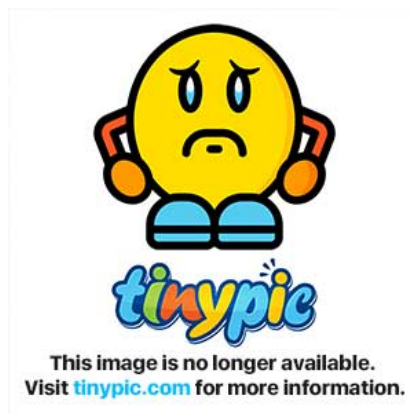
The most useful libraries required for web scraping are:

1. [Beautiful Soup](#).
2. [Requests](#).

These are the steps that we would be following throughout this tutorial:

1. Importing the required libraries.
2. Getting the URL and storing it in a variable.
3. Making a request to the website using the requests library.
4. Using the BeautifulSoup library to get the HTML (raw) data from the website.
5. Using soup.findAll method to get the respected tag that we are looking for.
6. Removing all the HTML tags and converting it to a plain text format.

You might be wondering what tags to extract, well in the Rate My Professor website every professor will have his/her respected tags such as (hilarious, heavy homework, study hard or fail, etc.), we will just try to extra these tags in this tutorials as shown below.



Before we begin make sure to scrape the data at a slow pace, and you can also use a VPN service to get a different IP address, this prevents your IP address from banning, but I hope you guys will follow the

instructions. Here is an [article](#) that will let you know how to scrape a website without getting blacklisted. One important thing in this tutorial there is no point of me explaining each and every line of code, which is not needed here because python code is self-explanatory. However, I will try not to confuse you, and make things clear in an easy way. So I wrote this tutorial in such a way that everybody can understand irrespective of their programming background. Moreover, the entire source code can be found in my GitHub. There might be numerous tutorials available on the internet, but this tutorial is easy to understand because I have tried to explain the code as much as possible, some parts are a mechanical process, wherein you just have to follow them, just let me know if you have any doubt in the comments section down below.

Let's get started.

Step 1: Importing the required libraries

Let us import few important libraries such as Requests and BeautifulSoup.

In [0]:

```
import requests
from bs4 import BeautifulSoup
```

Step 2: Getting the URL and storing it in a variable.

Let us store the URL of the professor in the variable named "url". The URL of the website can be found here: ["Rate My Professor"](#).

In [0]:

```
url = 'https://www.ratemyprofessors.com/ShowRatings.jsp?tid=1986099'
```

Step 3: Making a request to the website using the requests library.

Here we use the requests library by passing "url" as a parameter, be careful don't run this multiple times. If you get like Response 200 then its success, if you get something else then there is something wrong with maybe the code or your browser I don't know.

In [0]:

```
page = requests.get(url)
```

In [5]:

```
page
```

Out[5]:

```
<Response [200]>
```

Step 4: Using the BeautifulSoup library to get the HTML (raw) data from the website.

Here we use the BeautifulSoup by passing the page.text as a parameter and using the HTML parser. You can try to print the soup, but printing the soup doesn't give you the answer, rather it contains huge chunks of HTML data, so I decided not to show it here.

In [0]:

```
soup = BeautifulSoup(page.text, "html.parser")
```

Step 5: Using soup.findAll method to get the respected tag that we are looking for.

Here is the place where you shall add the tags that you are looking for, to get the tag name all you have to do is to right click on the respected tag or click Ctrl-Shift-I on the tag in the webpage, then a page with selected tag will open for you to your right-hand side as shown below:



You can then copy the HTML tag and class if any, and then place it inside the soup.findAll method. In this case, the HTML tag is "span" and class is "tag-box-choosetags"

In [7]:

```
proftags = soup.findAll("span", {"class": "tag-box-choosetags" })
proftags
```

Out[7]:

```
[<span class="tag-box-choosetags"> Hilarious <b>(11)</b></span>,
<span class="tag-box-choosetags"> Caring <b>(9)</b></span>,
<span class="tag-box-choosetags"> Accessible outside class <b>(8)</b></spa
<span class="tag-box-choosetags"> Amazing lectures <b>(5)</b></span>,
<span class="tag-box-choosetags"> Clear grading criteria <b>(4)</b></span>
<span class="tag-box-choosetags"> Inspirational <b>(4)</b></span>,
<span class="tag-box-choosetags"> Group projects <b>(3)</b></span>,
<span class="tag-box-choosetags"> Respected <b>(3)</b></span>,
<span class="tag-box-choosetags"> Gives good feedback <b>(2)</b></span>,
<span class="tag-box-choosetags"> EXTRA CREDIT <b>(2)</b></span>,
<span class="tag-box-choosetags"> Participation matters <b>(1)</b></span>,
<span class="tag-box-choosetags"> Lecture heavy <b>(1)</b></span>,
<span class="tag-box-choosetags"> Test heavy <b>(1)</b></span>,
<span class="tag-box-choosetags"> So many papers <b>(1)</b></span>]
```

Step 6: Removing all the HTML tags and converting it to a plain text format.

Here we remove all the HTML tags and convert it to a text format, this can be done with the help of `get_text` method placed inside a for loop. This converts the HTML into the text format.

In [8]:

```
for mytag in proftags:
    print(mytag.get_text())
```

```
Hilarious (11)
Caring (9)
Accessible outside class (8)
Amazing lectures (5)
Clear grading criteria (4)
Inspirational (4)
Group projects (3)
Respected (3)
Gives good feedback (2)
EXTRA CREDIT (2)
Participation matters (1)
Lecture heavy (1)
Test heavy (1)
So many papers (1)
```