**Web Scraping 1: Scraping Table Data**

In this post, we will learn how to scrape table data from the web using Python. Simplified.

[[Kiprono Elijah Koech](https://medium.com/@kiprono_65591?source=post_page-----1665b6b2271c--------------------------------)](https://medium.com/@kiprono_65591?source=post_page-----1665b6b2271c--------------------------------)

[Kiprono Elijah Koech](https://medium.com/@kiprono_65591?source=post_page-----1665b6b2271c--------------------------------)

[Jul 29·4 min read](https://towardsdatascience.com/web-scraping-scraping-table-data-1665b6b2271c?source=post_page-----1665b6b2271c--------------------------------)

Image for post



Photo by [Carlos Muza](https://unsplash.com/@kmuza?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com/?utm_source=medium&utm_medium=referral)

Web Scraping is the most important concept of data collection. In Python, **BeautifulSoup**, **Selenium** and **XPath**are the most important tools that can be used to accomplish the task of web scraping.

In this article, we will focus on BeautifulSoup and how to use it to scrape GDP data from [Wikipedia page](https://en.wikipedia.org/wiki/List_of_countries_by_GDP_(nominal)_per_capita). The data we need on this site is in form of a table.

Definition of Concepts

Take a look at the following image then we can go ahead and define the components of an HTML table

Image for post

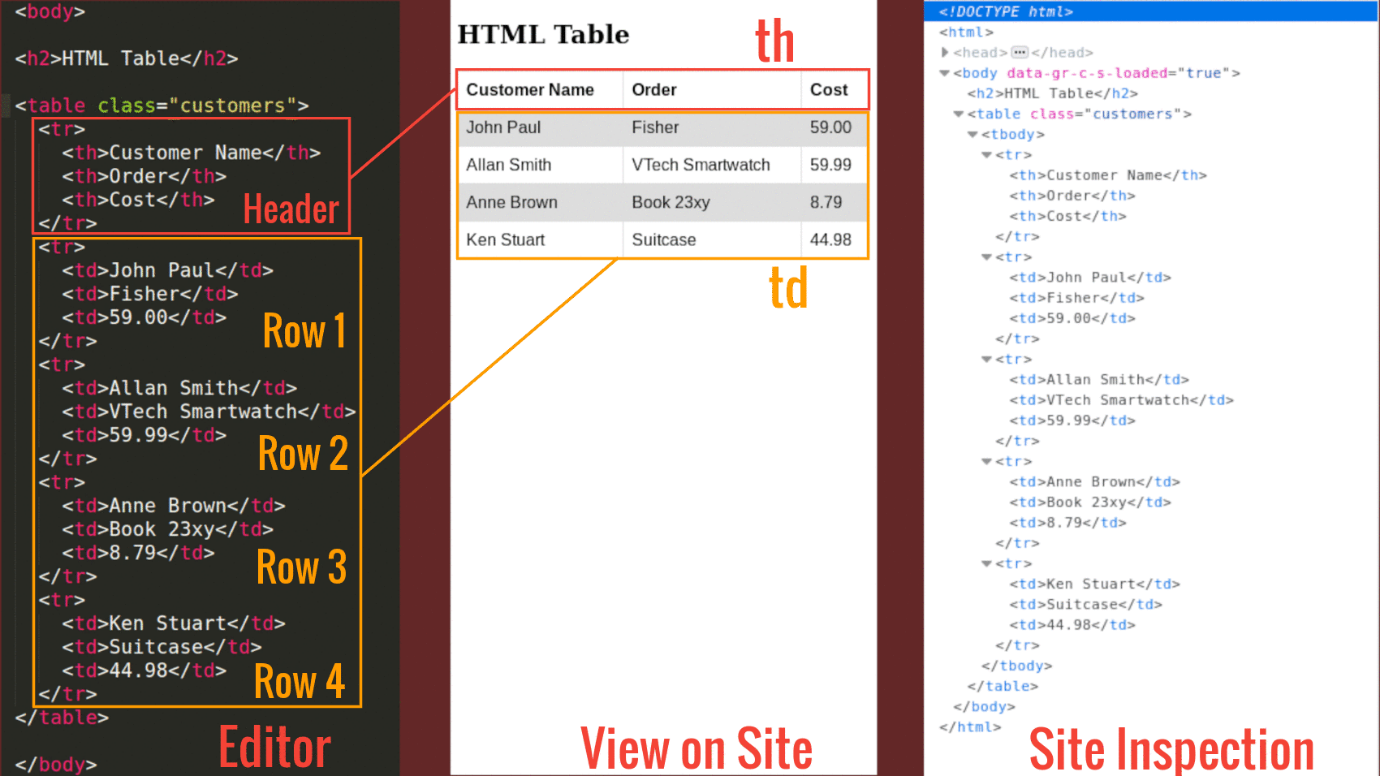


Fig 2

**From the above image we can deduce the following:**

The <table> tag defines an HTML table.

An HTML table consists of one <table> element and one or more <tr>, <th>, and <td> elements.

The <tr> element defines a table row, the <th> element defines a table header, and the <td> element defines a table cell.

An HTML table may also include <caption>, <colgroup>, <thead>, <tfoot>, and <tbody> elements.

Our interest is to inspect the elements of a given site (in this case the site we want to scrap — on the far right of Figure 1 shows the elements of the site). In most computers you visit the site and click **Ctrl+Shift+I**to inspect the page you wish to scrap.

**Note: Elements of a web page are identified by using a class or id options on the tag. Ids are unique but classes are not. This means that a given class can identify more than one web element while one id identifies one and only one element.**

Lets now see the image of the site we want to scrape

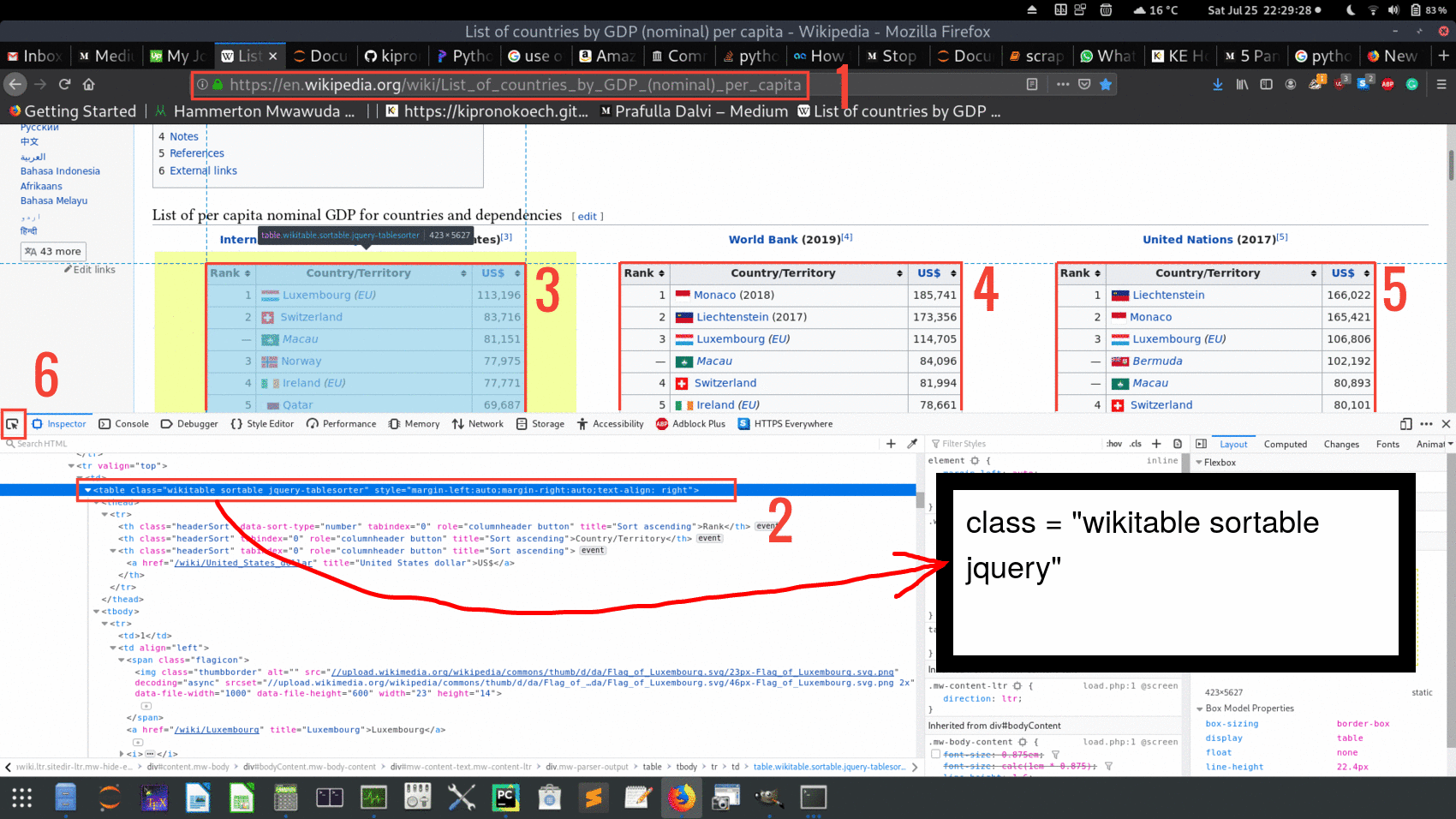


Fig 3

From this Figure note the following:

1. This is the Uniform Resource Locator (**URL**). We need this.
2. Tag element for our object of interest. The object is defined by **class** and not **id**: class = “wikitable sortable jquery”.Note that the tag element contains 3 classes identifying one table (classes are separated by white space). Apart from general reference of a site element as we will use here, **classes** and **ids** are used as reference to support styling using languages like CSS.
3. In the site there are 3 tables numbered **3**, **4** and **5** in the Figure above. For the sake of this article we will go through how to scrape table **3** and you can easily figure out of how to do **4** and **5**.
4. The button labelled 6 is very important when you are hovering through the page to identify the elements of your interest. Once your object of interest is highlighted the tag element will also be highlighted. e.g for our case label **2** is matching label **3**.

Actual Scraping

Required packages: **bs4, lxml, pandas and requests.**

Once you have the said packages we can now go through the code.

In this snippet, we import necessary packages and parse HTML content of the site.

Snippet 1

Upon inspection of the site elements and/or soup variable in snippet 1 you will realize that the three tables shown in Fig 3 belongs to the same class wikitable .

Snippet 2

OUTPUT:   
Number of tables on site: 3

To scrap the first table we just need to get the 0th index from gdp. The body variable in the snippet below contains all the rows of the table including the header. The headers will be in body[0] and all the others rows will be in the list body[1:]

Snippet 3

OUTPUT:  
['Rank', 'Country/Territory', 'US$']

The headers looks right.

Next, we need to loop through rest of the rows as below

Snippet 4

At this point we now have the headers on headings and all the other rows on all\_rows and from here we can pass this data and headers as a Pandas DataFrame

Image for post

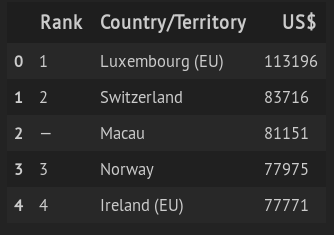


Fig 4

[Full code](https://github.com/kipronokoech/Web-Scraper/tree/master/gdp-data)

Caution

Web content are protected by copyright which, in most cases, restrict copy, distribution, transmission and adoption of the content. When you intent to scrap such a website you may need to get express and explicit permission for the content owners otherwise your act of scraping without proper consent may land you in trouble of breach.

Some sites, however, provides its content under [Creative Commons License](https://creativecommons.org/) allowing you to copy, modify and distribute there content without any problem though some may provide restriction on commercialization of such content.

The bottom line is that before web scraping, you need to be sure that you have the right to do so.

Conclusion

Web scraping is the most important tool you can use to collect data from the web. To learn more about web scraping you can go through [this repository](https://github.com/kipronokoech/Web-Scraper) which contains more examples.

Thanks for reading.