Web Scraping Tables using Pandas

Estimated Effort: 5 mins

The Pandas library in Python contains a function read_html() that can be used to extract tabular information from any web page.

Consider the following example:

Let us assume we want to extract the list of the largest banks in the world by market capitalization, from the following link:

URL = 'https://en.wikipedia.org/wiki/List_of_largest_banks'

We may use pandas.read_html() function in python to extract all the tables in the web page directly.

A snapshot of the webpage is shown below.















Search

Contents [hide]

(Top)

By market capitalization

By total assets

Banks by country or territory

See also

References

List of largest banks

Article Talk

From Wikipedia, the free encyclopedia

The following are lists of the largest banks in the world, as measured by market capitalization

By market capitalization [edit]

The list is based on Forbes.com's ranking as of August 2023 based on an analysis of the bar the global economy.^[1]

Rank ¢	Bank name \$	Market cap [hide] (US\$ billion)
1	JPMorgan Chase	419.25
2	Bank of America	231.52
3	Industrial and Commercial Bank of China	194.56
4	Agricultural Bank of China	160.68
5	HDFC Bank	157.91
6	Wells Fargo	155.87
7	HSBC Holdings PLC	148.90
8	Morgan Stanley	140.83
9	China Construction Bank	139.82
10	Bank of China	136.81

We can see that the required table is the first one in the web page.

Note: This is a live web page and it may get updated over time. The image shown above has been captured in November 2023. The process of data extraction remains the same.

about:blank 1/4

We may execute the following lines of code to extract the required table from the web page.

```
import pandas as pd
URL = 'https://en.wikipedia.org/wiki/List_of_largest_banks'
tables = pd.read_html(URL)
df = tables[0]
print(df)
```

This will extract the required table as a dataframe df. The output of the print statement would look as shown below.

	Rank	Bank name	Market cap(US\$ billion)
0	1	JPMorgan Chase	419.25
1	2	Bank of America	231.52
2	3	Industrial and Commercial Bank of China	194.56
3	4	Agricultural Bank of China	160.68
4	5	HDFC Bank	157.91
5	6	Wells Fargo	155.87
6	7	HSBC Holdings PLC	148.90
7	8	Morgan Stanley	140.83
8	9	China Construction Bank	139.82
9	10	Bank of China	136.81

Although convenient, this method comes with its own set of limitations.

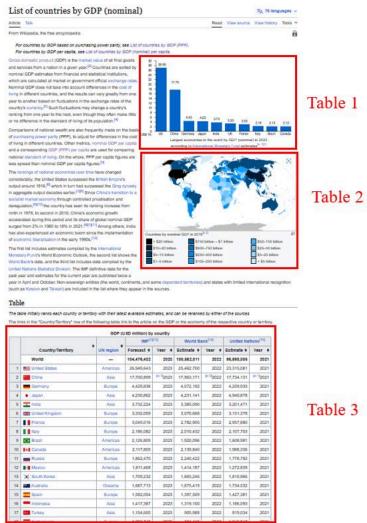
Firstly, web pages may have content saved in them as tables but they may not appear as tables on the web page.

For instance, consider the following URL showing the list of countries by GDP (nominal).

URL = 'https://en.wikipedia.org/wiki/List_of_countries_by_GDP_(nominal)'

about:blank 2/4

The images on the web page are also saved in tabular format. A snapshot of the web page is shared below.



Secondly, the contents of the tables in the web pages may contain elements such as hyperlink text and other denoters, which are also scraped directly using the pandas method. This may lead to a requirement of further cleaning of data.

A closer look at table 3 in the image shown above indicates that there are many hyperlink texts which are also going to be treated as information by the pandas function.

about:blank 3/4

GDP (USD million) by country World Bank [14] United Nations [15] IMF [1][13] UN Country/Territory Forecast \$ Estimate \$ Year ♦ Estimate \$ region Year + Year \$ 96.698.005 World 104,476,432 2023 100.562.011 2022 2021 United States 25,462,700 1 Americas 26,949,643 2023 2022 23,315,081 2021 [n 1]2023 [n 3]2022 17,700,899 17,963,171 17,734,131 [n 1]2021 2 China Asia 3 4,429,838 2023 4,072,192 2022 4,259,935 2021 Germany Europe 4,230,862 2023 4,231,141 2022 4,940,878 2021 4 Japan Asia 3,201,471 5 India Asia 3,732,224 2023 3,385,090 2022 2021 3,332,059 2023 3,070,668 2022 3,131,378 2021 6 United Kingdom Europe 7 2023 2,782,905 2022 2,957,880 2021 France Europe 3.049.016 2023 2,010,432 2022 2,107,703 8 Italy Europe 2,186,082 2021 1,608,981 9 Brazil Americas 2,126,809 2023 1.920.096 2022 2021 10 ■◆■ Canada Americas 2,117,805 2023 2,139,840 2022 1,988,336 2021 2023 2021 11 Russia Europe 1,862,470 2,240,422 2022 1,778,782 2023 12 Mexico Americas 1,811,468 1,414,187 2022 1,272,839 2021 2021 13 South Korea Asia 1,709,232 2023 1,665,246 2022 1,810,966 14 Australia Australia Oceania 1,687,713 2023 1,675,419 2022 1,734,532 2021

1,582,054

2023

1,397,509

2022

1,427,381

2021

We can extract the table using the code shown below.

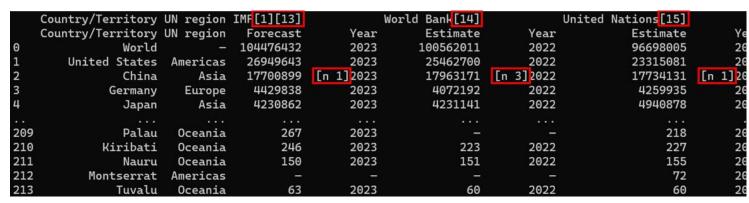
15

Spain

```
import pandas as pd
URL = 'https://en.wikipedia.org/wiki/List_of_countries_by_GDP_(nominal)'
tables = pd.read_html(URL)
df = tables[2] # the required table will have index 2
print(df)
```

Europe

The output of the print statement is shown below.



Note that the hyperlink texts have also been retained in the code output.

It is further prudent to point out, that this method exclusively operates only on tabular data extraction. BeautifulSoup library still remains the default method of extracting any kind of information from web pages.

Author(s)

Abhishek Gagneja

about:blank 4/4