Web scraping data from wikipedia using pandas and BeautifulSoup

Following code scrapes data from the wikipedia page of Largest companies in the us based on revenue with
url='https://en.wikipedia.org/wiki/List_of_largest_companies_in_the_United_States_by_revenue'

step1

-> import the libraries Beautiful Soup(used to scrape information from webpage) and requests(makes HTTP requests that makes it easy to send and recieve information from websites providing a uniform interface)

In [2]:
```
from bs4 import BeautifulSoup

import requests
```

step2

-> retrieve the data from a resource using .get() method.

-> print out the page to get a response to make sure the command is working prefferably 200.

In [3]:
```
url = 'https://en.wikipedia.org/wiki/List_of_largest_companies_in_the_United_States_by_
```

In [4]:
```
page= requests.get(url)
```

In [5]:
```
print(page)
```

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

step3

-> use BeautifulSoup to parse the page

-> find the html tags associated with the table we want to extract using the inspect in the given url [.find('table')]

In [6]:
```
soup = BeautifulSoup(page.text,'html')
```

In [7]:
```
soup.find('table')
```

Out[7]:
```
<table class="box-More_citations_needed plainlinks metadata ambox ambox-content ambox-Re
fimprove" role="presentation"><tbody><tr><td class="mbox-image"><div class="mbox-image-d
iv"><a class="image" href="/wiki/File:Question_book-new.svg"><img alt="" data-file-heigh
t="399" data-file-width="512" decoding="async" height="39" src="//upload.wikimedia.org/w
ikipedia/en/thumb/9/99/Question_book-new.svg/50px-Question_book-new.svg.png" srcset="//u
pload.wikimedia.org/wikipedia/en/thumb/9/99/Question_book-new.svg/75px-Question_book-ne
w.svg.png 1.5x, //upload.wikimedia.org/wikipedia/en/thumb/9/99/Question_book-new.svg/100
px-Question_book-new.svg.png 2x" width="50"/></a></div></td><td class="mbox-text"><div c
lass="mbox-text-span">This article <b>needs additional citations for <a href="/wiki/Wiki
pedia:Verifiability" title="Wikipedia:Verifiability">verification</a></b>.<span class="h
ide-when-compact"> Please help <a class="external text" href="https://en.wikipedia.org/
w/index.php?title=List_of_largest_companies_in_the_United_States_by_revenue&amp;action=e
dit">improve this article</a> by <a href="/wiki/Help:Referencing_for_beginners" title="H
elp:Referencing for beginners">adding citations to reliable sources</a>. Unsourced mater
ial may be challenged and removed.<br/><small><span class="plainlinks"><i>Find sources:
</i> <a class="external text" href="https://www.google.com/search?as_eq=wikipedia&amp;q
=%22List+of+largest+companies+in+the+United+States+by+revenue%22" rel="nofollow">"List o
f largest companies in the United States by revenue"</a> – <a class="external text" href
="https://www.google.com/search?tbm=nws&amp;q=%22List+of+largest+companies+in+the+United
+States+by+revenue%22+-wikipedia&amp;tbs=ar:1" rel="nofollow">news</a> <b>·</b> <a class
="external text" href="https://www.google.com/search?&amp;q=%22List+of+largest+companies
+in+the+United+States+by+revenue%22&amp;tbs=bkt:s&amp;tbm=bks" rel="nofollow">newspapers
</a> <b>·</b> <a class="external text" href="https://www.google.com/search?tbs=bks:1&am
p;q=%22List+of+largest+companies+in+the+United+States+by+revenue%22+-wikipedia" rel="nof
ollow">books</a> <b>·</b> <a class="external text" href="https://scholar.google.com/scho
lar?q=%22List+of+largest+companies+in+the+United+States+by+revenue%22" rel="nofollow">sc
holar</a> <b>·</b> <a class="external text" href="https://www.jstor.org/action/doBasicS
earch?Query=%22List+of+largest+companies+in+the+United+States+by+revenue%22&amp;acc=on&
amp;wc=on" rel="nofollow">JSTOR</a></span></small></span> <span class="date-container">
<i>(<span class="date">June 2020</span>)</i></span><span class="hide-when-compact"><i>
(<small><a href="/wiki/Help:Maintenance_template_removal" title="Help:Maintenance templ
ate removal">Learn how and when to remove this template message</a></small>)</i></span>
</div></td></tr></tbody></table>
```

In [8]:
```
Table=soup.find_all('table')[1]
```

step 4

-> find all 'th' tags (for the heading of the table)

-> use.strip() to remove the spaces and /n from the output

In [12]:
```
world_titles=Table.find_all('th')
```

In [10]:
```
world_table_titles= [title.text.strip() for title in world_titles]
print(world_table_titles)
```

```
['Rank', 'Name', 'Industry', 'Revenue (USD millions)', 'Revenue growth', 'Employees', 'H
eadquarters']
```

step 5

-> import the pandas library

->put the extracted column names into a dataframe

In [13]:
```
import pandas as pd
```

In [14]:
```
df = pd.DataFrame(columns=world_table_titles)

df
```

Out[14]:

Rank	Name	Industry	Revenue (USD millions)	Revenue growth	Employees	Headquarters

step 6

-> find all the 'tr' tags which gives the row data

-> start from the second row as the first row has no data

-> find all 'td' which gives out the individual row data

-> insert all the individual data into the dataframe

In [15]:
```
column_data=Table.find_all('tr')
```

In [16]:
```
for row in column_data[1:]:
    row_data=row.find_all('td')
    individual_row_data = [data.text.strip() for data in row_data]

    length=len(df)
    df.loc[length]=individual_row_data
```

In [17]:
```
df
```

Out[17]:

	Rank	Name	Industry	Revenue (USD millions)	Revenue growth	Employees	Headquarters
0	1	Walmart	General merchandisers	572,754	2.4%	2,300,000	Bentonville, Arkansas
1	2	Amazon	Retail and Cloud Computing	469,822	21.7%	1,608,000	Seattle, Washington
2	3	Apple	Electronics industry	365,817	33.2%	154,000	Cupertino, California
3	4	CVS Health	Healthcare	292,111	32.0%	258,000	Woonsocket, Rhode Island
4	5	UnitedHealth	Healthcare	287,597	11.8%	350,000	Minnetonka,

	Rank	Name	Industry	Revenue (USD millions)	Revenue growth	Employees	Headquarters
		Group					Minnesota
...	...	...	...	...	...	...	...
95	96	General Dynamics	Airspace and defense	38,469	8.7%	103,100	Reston, Virginia
96	97	CHS	Agriculture cooperative	38,448	1.4%	9,941	Inver Grove Heights, Minnesota
97	98	USAA	Financials	37,470	3.2%	37,335	San Antonio, Texas
98	99	Northwestern Mutual	Insurance	36,751	8.8%	7,585	Milwaukee, Wisconsin
99	100	Nucor	Metals	36,484	81.2%	28,800	Charlotte, North Carolina

100 rows × 7 columns

step7

-> save the scraped data in a csv file using df.to_csv(r'path')

In [ ]:
```
df.to_csv(r'E:\python')
```

In [ ]:
```

```