In [1]:
```
from bs4 import BeautifulSoup
import requests
```

In [2]:
```
url = 'https://en.wikipedia.org/wiki/List_of_largest_companies_in_the_United_States
```

In [3]:
```
requests.get(url)
```

Out[3]:
```
<Response [200]>
```

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

In [5]:
```
BeautifulSoup(page.text, 'html')
```

```
---------------------------------------------------------------------------
FeatureNotFound                           Traceback (most recent call last)
Cell In[5], line 1
----> 1 BeautifulSoup(page.text, 'htmlll')

File ~\anaconda3\lib\site-packages\bs4\__init__.py:248, in BeautifulSoup.__init__
(self, markup, features, builder, parse_only, from_encoding, exclude_encodings, el
ement_classes, **kwargs)
    246     builder_class = builder_registry.lookup(*features)
    247     if builder_class is None:
--> 248         raise FeatureNotFound(
    249             "Couldn't find a tree builder with the features you "
    250             "requested: %s. Do you need to install a parser library?"
    251             % ",".join(features))
    253 # At this point either we have a TreeBuilder instance in
    254 # builder, or we have a builder_class that we can instantiate
    255 # with the remaining **kwargs.
    256 if builder is None:

FeatureNotFound: Couldn't find a tree builder with the features you requested: htm
lll. Do you need to install a parser library?
```

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

In [7]:
```
# to find the DOM data using tag
soup.find('table')
```

In [8]:
```
# returns list of matching data from DOM
soup.find_all('table')
```

Out[8]:
```
[]
```

In [9]:
```
#soup.find('table', class_ = 'wikitable sortable jquery-tablesorter') -- entire cla
soup.find('table', class_ = 'wikitable sortable')
```

In [13]:
```
table = soup.find('table', class_ = 'wikitable sortable')
```

In [16]:
```
world_titles = table.find_all('th')
```

In [17]:
```
print(world_titles)
```

```
[<th>Rank
</th>, <th>Name
</th>, <th>Industry
</th>, <th>Revenue <br/>(USD millions)
</th>, <th>Revenue growth
</th>, <th>Employees
</th>, <th>Headquarters
</th>]
```

In [18]:

```
world_table_titles = [title.text for title in world_titles]

print(world_table_titles)
```

```
['Rank\n', 'Name\n', 'Industry\n', 'Revenue (USD millions)\n', 'Revenue growth\n',
'Employees\n', 'Headquarters\n']
```

In [19]:

```
world_table_titles = [title.text.strip() for title in world_titles]

print(world_table_titles)
```

```
['Rank', 'Name', 'Industry', 'Revenue (USD millions)', 'Revenue growth', 'Employee
s', 'Headquarters']
```

In [21]:

```
import pandas as pd

df = pd.DataFrame(columns = world_table_titles)

df
```

Out[21]:

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

In [22]:

```
column_data = table.find_all('tr')
```

In [10]:

```
for row in column_data:
    print(row.find_all('td'))
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[10], line 1
----> 1 for row in column_data:
      2     print(row.find_all('tdd'))

NameError: name 'column_data' is not defined
```

In [11]:

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

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[11], line 1
----> 1 for row in column_data:
      2     row_data = row.find_all('td')
      3     individual_row_data = [data.text.strip() for data in row_data]

NameError: name 'column_data' is not defined
```

In [12]:

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

    # insert the row data in to df
    length = len(df)
    df.loc[length] = individual_row_data
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[12], line 1
----> 1 for row in column_data:
      2     row_data = row.find_all('td')
      3     individual_row_data = [data.text.strip() for data in row_data]

NameError: name 'column_data' is not defined
```

In [26]:

```
# error due to an empty row in the individual_row_data in the begining
```

In [32]:

```
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 [33]:

```
df
```

Out[33]:

| | Rank | Name | Industry | Revenue (USD millions) | Revenue growth | Employees | Headquarters |
|---|---|---|---|---|---|---|---|
| **length** | 100 | Qualcomm | Technology | 44,200 | 31.7% | 51,000 | San Diego, California |
| **1** | 1 | Walmart | Retail | 611,289 | 6.7% | 2,100,000 | Bentonville, Arkansas |
| **2** | 2 | Amazon | Retail and cloud computing | 513,983 | 9.4% | 1,540,000 | Seattle, Washington |
| **3** | 3 | ExxonMobil | Petroleum industry | 413,680 | 44.8% | 62,000 | Spring, Texas |
| **4** | 4 | Apple | Electronics industry | 394,328 | 7.8% | 164,000 | Cupertino, California |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **96** | 96 | Best Buy | Retail | 46,298 | 10.6% | 71,100 | Richfield, Minnesota |
| **97** | 97 | Bristol-Myers Squibb | Pharmaceutical industry | 46,159 | 0.5% | 34,300 | New York City, New York |
| **98** | 98 | United Airlines | Airline | 44,955 | 82.5% | 92,795 | Chicago, Illinois |
| **99** | 99 | Thermo Fisher Scientific | Laboratory instruments | 44,915 | 14.5% | 130,000 | Waltham, Massachusetts |
| **100** | 100 | Qualcomm | Technology | 44,200 | 31.7% | 51,000 | San Diego, California |

101 rows × 7 columns

In [34]:

```
df.to_csv(r'C:\Users\kallzz\Desktop\Data Analytics Stuff\Data Analyst - Boot Camp\P
```

In [36]:

```
# to remove index from the data
df.to_csv(r'C:\Users\kallzz\Desktop\Data Analytics Stuff\Data Analyst - Boot Camp\P
```

In [ ]: