In [1]:
```
from bs4 import BeautifulSoup
import requests
#BeautifulSoup is a Python library for pulling data out of HTML and XML files.

url = 'https://en.wikipedia.org/wiki/List_of_largest_companies_in_the_United_States_by_revenue'
page = requests.get(url)   # this stores the HTML code of the site
soup = BeautifulSoup(page.text)
# here extracting the text, page attribute is used to extract the text content from the response.
```

In [ ]:
```
print(soup)
```

In [2]:
```
table = soup.find_all('table')[1] # here we are storing the 2nd <table> tag from the site to table objec
```

In [ ]:
```
soup.find_all('table', class_ ='wikitable sortable') # checking the class name of the table
```

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

In [5]:
```
world_titles
```

Out[5]:
```
[<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 [14]:
```
world_table_titles = [title.text.strip() for title in world_titles]
# storing the title of table in the variable after formatting it correctly
```

In [ ]:
```
print(world_table_titles)
```

In [7]:
```
import pandas as pd
# pandas particularly useful for working with tabular data, such as spreadsheets or SQL tables.
```

In [8]:
```
df = pd.DataFrame(columns = world_table_titles)
# converting the title into columns by using dataframe function
```

In [10]:
```
df
```

Out[10]:

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

In [9]:
```
column_data = table.find_all('tr')
# storing the row data into the variable
```

In [ ]:
```
for row in column_data[1:]:
    row_data = row.find_all('td')
    individual_row_data = [data.text.strip() for data in row_data]
   # formatting the table row (<td>) data and storing in variable

    length = len(df)
    df.loc[length] = individual_row_data
    # adds a new row to the Pandas DataFrame df. It uses the loc accessor to
    # access the DataFrame at the specified index (length)
```

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

Out[17]:

| | Rank | Name | Industry | Revenue (USD millions) | Revenue growth | Employees | Headquarters |
|---|---|---|---|---|---|---|---|
| **0** | 1 | Walmart | Retail | 611,289 | 6.7% | 2,100,000 | Bentonville, Arkansas |
| **1** | 2 | Amazon | Retail and cloud computing | 513,983 | 9.4% | 1,540,000 | Seattle, Washington |
| **2** | 3 | ExxonMobil | Petroleum industry | 413,680 | 44.8% | 62,000 | Spring, Texas |
| **3** | 4 | Apple | Electronics industry | 394,328 | 7.8% | 164,000 | Cupertino, California |
| **4** | 5 | UnitedHealth Group | Healthcare | 324,162 | 12.7% | 400,000 | Minnetonka, Minnesota |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **95** | 96 | Best Buy | Retail | 46,298 | 10.6% | 71,100 | Richfield, Minnesota |
| **96** | 97 | Bristol-Myers Squibb | Pharmaceutical industry | 46,159 | 0.5% | 34,300 | New York City, New York |
| **97** | 98 | United Airlines | Airline | 44,955 | 82.5% | 92,795 | Chicago, Illinois |
| **98** | 99 | Thermo Fisher Scientific | Laboratory instruments | 44,915 | 14.5% | 130,000 | Waltham, Massachusetts |
| **99** | 100 | Qualcomm | Technology | 44,200 | 31.7% | 51,000 | San Diego, California |

100 rows × 7 columns

In [18]:
```
df.to_csv(r"C:\Users\shashank verma\Downloads\Projects\Companies.csv", index = False)
# creating csv file and storing in memory
```

In [ ]:
```

```