

```
In [1]: # Programmer: Neja Gurung, Laxmi Gurung
# Course: CIS 3120-ETRA (34795)
# Project: WebScarping a website,
#         converting a list of strings to integers and finding an average
# Date: 09/29/2021
```

```
In [2]: #To request in to the website, importing requests library
#To get the html(raw) data from the website, importing beautiful soup library

import requests
from bs4 import BeautifulSoup

url = "http://www.theus50.com/fastfacts/area.php"
r = requests.get(url) # r is the response object that retrieves context.
src = r.content
soup = BeautifulSoup(src, 'html.parser')

# 1. The heading of the page that describes the content of the table.
pageTitle = soup.find_all('h4')
print("The Content of Table.")
print(pageTitle)

#There is only one table tag in the html document of the given website, therefore we can
#find or find_all
allTable = soup.find('table')
#print(allTable)
```

The Content of Table.
[<h4 class="pageTitle">(State Areas)</h4>]

```
In [3]: # Create an empty list to store the field name of the table
stateList = []
headers = []

# To iterate over the first column which is the field names of the table
# To find the first table row, we are using find method
# There is also only one <th> tag.
# Using find_all method to extract the table headings.

for header in allTable.find("tr").find_all("th"):
    headers.append(header.text)

print(headers)
```

['Rank', 'State Name', 'Area (Sq Miles)']

```
In [4]: # After finding the field names which was at position 0 in allTable
# Assigning all the table rows in a variable rows
rows = allTable.find_all("tr")[1:]
#print(rows)

#To iterate over rows and extracting the table data storing in
#an empty list values[]
for row in rows:
    values = []
```

```

for col in row.find_all('td'):
    values.append(col.text)# .text returns the data in string format

if values:
    stateDict = {} #Creating an empty dictionary

    # To store all the data from values list in a list of dictionary
    for i in range(len(values)):
        #print(headers[i])
        #print(values[i])
        stateDict[headers[i]]= values[i]
    stateList.append(stateDict)
print(stateDict)

```

```

{'Rank': '1', 'State Name': 'Alaska', 'Area (Sq Miles)': '571,951'}
{'Rank': '2', 'State Name': 'Texas', 'Area (Sq Miles)': '261,797'}
{'Rank': '3', 'State Name': 'California', 'Area (Sq Miles)': '155,959'}
{'Rank': '4', 'State Name': 'Montana', 'Area (Sq Miles)': '145,552'}
{'Rank': '5', 'State Name': 'New Mexico', 'Area (Sq Miles)': '121,356'}
{'Rank': '6', 'State Name': 'Arizona', 'Area (Sq Miles)': '113,635'}
{'Rank': '7', 'State Name': 'Nevada', 'Area (Sq Miles)': '109,826'}
{'Rank': '8', 'State Name': 'Colorado', 'Area (Sq Miles)': '103,718'}
{'Rank': '9', 'State Name': 'Wyoming', 'Area (Sq Miles)': '97,100'}
{'Rank': '10', 'State Name': 'Oregon', 'Area (Sq Miles)': '95,997'}
{'Rank': '11', 'State Name': 'Idaho', 'Area (Sq Miles)': '82,747'}
{'Rank': '12', 'State Name': 'Utah', 'Area (Sq Miles)': '82,144'}
{'Rank': '13', 'State Name': 'Kansas', 'Area (Sq Miles)': '81,815'}
{'Rank': '14', 'State Name': 'Minnesota', 'Area (Sq Miles)': '79,610'}
{'Rank': '15', 'State Name': 'Nebraska', 'Area (Sq Miles)': '76,872'}
{'Rank': '16', 'State Name': 'South Dakota', 'Area (Sq Miles)': '75,885'}
{'Rank': '17', 'State Name': 'North Dakota', 'Area (Sq Miles)': '68,976'}
{'Rank': '18', 'State Name': 'Missouri', 'Area (Sq Miles)': '68,886'}
{'Rank': '19', 'State Name': 'Oklahoma', 'Area (Sq Miles)': '68,667'}
{'Rank': '20', 'State Name': 'Washington', 'Area (Sq Miles)': '66,544'}
{'Rank': '21', 'State Name': 'Georgia', 'Area (Sq Miles)': '57,906'}
{'Rank': '22', 'State Name': 'Michigan', 'Area (Sq Miles)': '56,804'}
{'Rank': '23', 'State Name': 'Iowa', 'Area (Sq Miles)': '55,869'}
{'Rank': '24', 'State Name': 'Illinois', 'Area (Sq Miles)': '55,584'}
{'Rank': '25', 'State Name': 'Wisconsin', 'Area (Sq Miles)': '54,310'}
{'Rank': '26', 'State Name': 'Florida', 'Area (Sq Miles)': '53,927'}
{'Rank': '27', 'State Name': 'Arkansas', 'Area (Sq Miles)': '52,068'}
{'Rank': '28', 'State Name': 'Alabama', 'Area (Sq Miles)': '50,744'}
{'Rank': '29', 'State Name': 'North Carolina', 'Area (Sq Miles)': '48,711'}
{'Rank': '30', 'State Name': 'New York', 'Area (Sq Miles)': '47,214'}
{'Rank': '31', 'State Name': 'Mississippi', 'Area (Sq Miles)': '46,907'}
{'Rank': '32', 'State Name': 'Pennsylvania', 'Area (Sq Miles)': '44,817'}
{'Rank': '33', 'State Name': 'Louisiana', 'Area (Sq Miles)': '43,562'}
{'Rank': '34', 'State Name': 'Tennessee', 'Area (Sq Miles)': '41,217'}
{'Rank': '35', 'State Name': 'Ohio', 'Area (Sq Miles)': '40,948'}
{'Rank': '36', 'State Name': 'Kentucky', 'Area (Sq Miles)': '39,728'}
{'Rank': '37', 'State Name': 'Virginia', 'Area (Sq Miles)': '39,594'}
{'Rank': '38', 'State Name': 'Indiana', 'Area (Sq Miles)': '35,867'}
{'Rank': '39', 'State Name': 'Maine', 'Area (Sq Miles)': '30,862'}
{'Rank': '40', 'State Name': 'South Carolina', 'Area (Sq Miles)': '30,109'}
{'Rank': '41', 'State Name': 'West Virginia', 'Area (Sq Miles)': '24,078'}
{'Rank': '42', 'State Name': 'Maryland', 'Area (Sq Miles)': '9,774'}
{'Rank': '43', 'State Name': 'Vermont', 'Area (Sq Miles)': '9,250'}
{'Rank': '44', 'State Name': 'New Hampshire', 'Area (Sq Miles)': '8,968'}
{'Rank': '45', 'State Name': 'Massachusetts', 'Area (Sq Miles)': '7,840'}
{'Rank': '46', 'State Name': 'New Jersey', 'Area (Sq Miles)': '7,417'}
{'Rank': '47', 'State Name': 'Hawaii', 'Area (Sq Miles)': '6,423'}
{'Rank': '48', 'State Name': 'Connecticut', 'Area (Sq Miles)': '4,845'}
{'Rank': '49', 'State Name': 'Delaware', 'Area (Sq Miles)': '1,954'}
{'Rank': '50', 'State Name': 'Rhode Island', 'Area (Sq Miles)': '1,045'}

```

In [5]:

```
print(len(stateList))
```

50

In [6]:

```
# Create an empty list to store the Area (Sq miles) column's data.
```

```
stateAreas = []
```

```
# Iterating over the list of dictionary
```

```
for n in stateList:
```

```
    areaList =n['Area (Sq Miles)']
```

```
    print(areaList)
```

```
    stateAreas.append(areaList)
```

571,951

261,797

155,959

145,552

121,356

113,635

109,826

103,718

97,100

95,997

82,747

82,144

81,815

79,610

76,872

75,885

68,976

68,886

68,667

66,544

57,906

56,804

55,869

55,584

54,310

53,927

52,068

50,744

48,711

47,214

46,907

44,817

43,562

41,217

40,948

39,728

39,594

35,867

30,862

30,109

24,078

9,774

9,250

8,968

7,840

7,417

6,423
4,845
1,954
1,045

In [7]:

```
# Again creating an empty list to store the area of the states after forming  
# data.  
# We removed the ',' from each data, using the for loop, if statement  
#and replace method.
```

```
newAreaList = []  
for x in stateAreas:  
    if ',' in x:  
        formatArea= x.replace(',','')  
        newAreaList.append(formatArea)  
print(newAreaList)
```

```
['571951', '261797', '155959', '145552', '121356', '113635', '109826', '103718', '9710  
0', '95997', '82747', '82144', '81815', '79610', '76872', '75885', '68976', '68886', '68  
667', '66544', '57906', '56804', '55869', '55584', '54310', '53927', '52068', '50744',  
'48711', '47214', '46907', '44817', '43562', '41217', '40948', '39728', '39594', '3586  
7', '30862', '30109', '24078', '9774', '9250', '8968', '7840', '7417', '6423', '4845',  
'1954', '1045']
```

In [8]:

```
# Checking the data type of the data  
print(type(stateAreas[0]))  
  
# Created an empty list to store the list of integers after converting  
# the string data type using int()  
FinalAreaList = []  
for area in newAreaList:  
    FinalAreaList.append(int(area))  
  
print(FinalAreaList)  
  
# Converting the list of strings into integers using map()  
#FinalAreaList = list(map(int,newAreaList))  
#print(FinalAreaList)
```

```
<class 'str'>  
[571951, 261797, 155959, 145552, 121356, 113635, 109826, 103718, 97100, 95997, 82747, 82  
144, 81815, 79610, 76872, 75885, 68976, 68886, 68667, 66544, 57906, 56804, 55869, 55584,  
54310, 53927, 52068, 50744, 48711, 47214, 46907, 44817, 43562, 41217, 40948, 39728, 3959  
4, 35867, 30862, 30109, 24078, 9774, 9250, 8968, 7840, 7417, 6423, 4845, 1954, 1045]
```

In [9]:

```
# Assign a value 0 to the variable sum  
sum = 0  
  
# Using for to iterate over the list of areas and adding it to the variable sum  
for num in FinalAreaList:  
    sum += num # everytime the loops runs, the value will be added and stored in sum.  
  
# Calculating the average of area of all the states.  
average = sum/len(FinalAreaList)  
#print(len(FinalAreaList))  
# Displaying the result.  
print(f"The average of all the areas is {average} square miles.")
```

The average of all the areas is 70747.58 square miles.

