In [55]:
```
import json
```

This assignment I will present in Jupiter Notebook format. I will run these Markdown cells in the same way I run code cells. However, when I run a Markdown cell, the text formatted using Markdown syntax will be rendered as stylized text and not as a code. After that, I will present the code as a HTML file, convert it into a PDF document and upload through my GitHub Repository.

I will use comments in order to:

1. explain Python code
2. make the code more readable
3. prevent execution when testing code

As can be seen, first I have to import JSON file which has been already provided. A few important facts about this type of file:

JSON stands for JavaScript Object Notation. it is a text format for storing and transporting data JSON is "self-describing" and easy to understand Also, the requests module allows me to send HTTP requests using Python.

In [94]:
```
# Task 1 Read in a data file of all counties in the US.
# Make a list of unique county names.

with open("G:\My Drive\JSON\gz_2010_us_050_00_20m.json", 'r') as f:
    data = json.load(f)
    list2=list()
    list1=list()
    for i in range(len(data['features'])):
        for j in range(10):
            countyname = data['features'][i]['properties']['NAME']
            statefips = data['features'][i]['properties']['STATE']
            countyfips = data['features'][i]['properties']['COUNTY']
            countyarea = data['features'][i]['properties']['CENSUSAREA']
        list2.append(countyname)
        list2.append(countyfips)
        list2.append(statefips)
        list2.append(countyarea)
        list1.append(list2)
        list2=list()
    print("County Data FIRST50: \n", list1[0:50])
```

```
County Data FIRST50:
 [['Autauga', '001', '01', 594.436], ['Blount', '009', '01', 644.776], ['Chambers',
'017', '01', 596.531], ['Chilton', '021', '01', 692.854], ['Colbert', '033', '01', 59
2.619], ['Dale', '045', '01', 561.15], ['Elmore', '051', '01', 618.485], ['Hale', '06
5', '01', 643.943], ['Lawrence', '079', '01', 690.678], ['Limestone', '083', '01', 55
9.936], ['Monroe', '099', '01', 1025.675], ['Pickens', '107', '01', 881.408], ['Talla
dega', '121', '01', 736.775], ['Bethel', '050', '02', 40570.004], ['Hoonah-Angoon',
'105', '02', 7524.915], ['Kenai Peninsula', '122', '02', 16075.331], ['Kodiak Islan
d', '150', '02', 6549.579], ['Lake and Peninsula', '164', '02', 23652.009], ['Nome',
'180', '02', 22961.761], ['Northwest Arctic', '188', '02', 35572.584], ['Prince of Wa
les-Hyder', '198', '02', 3922.873], ['Van Buren', '141', '05', 708.143], ['White', '1
45', '05', 1035.075], ['Amador', '005', '06', 594.583], ['Glenn', '021', '06', 1313.9
47], ['Lake', '033', '06', 1256.464], ['Mariposa', '043', '06', 1448.816], ['Napa',
'055', '06', 748.362], ['Shasta', '089', '06', 3775.402], ['Stanislaus', '099', '06',
1494.827], ['Yuba', '115', '06', 631.839], ['Alamosa', '003', '08', 722.643], ['Bould
er', '013', '08', 726.289], ['Broomfield', '014', '08', 33.034], ['Crowley', '025',
'08', 787.421], ['Denver', '031', '08', 153.0], ['Douglas', '035', '08', 840.248],
['Gilpin', '047', '08', 149.896], ['Marion', '083', '12', 1584.546], ['Monroe', '08
7', '12', 983.282], ['Orange', '095', '12', 903.429], ['St. Lucie', '111', '12', 571.
926], ['Sumter', '119', '12', 546.933], ['Union', '125', '12', 243.556], ['Appling',
'001', '13', 507.081], ['Barrow', '013', '13', 160.309], ['Bryan', '029', '13', 435.9
67], ['Candler', '043', '13', 243.044], ['Chattooga', '055', '13', 313.338], ['Clark
e', '059', '13', 119.2]]
```

Sometimes, instead of our local file we can use url linke. A Uniform Resource Locator (URL) is actually a web address to open a specific site. Sometimes while working in python we need to fetch data from a website, for this I have to open the url of a specific website. So, to open a URL in python I need to import the specified module and perform some steps to open that URL.

Above I firsly defined variables (countyname, statefips, countyfips, countyarea) by comparing their names in our json file.

List is the most common, the most important and the most used built-in structure in Python. In addition, for the purpose of this assignemnt I will use for and range() loops. A for loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string). It works more like an iterator method as found in other object-orientated programming languages. With the for loop we can execute a set of statements, once for each item in a list, tuple, set etc. The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and stops before a specified number.

In [ ]:
```
# Below I had a problem with the code which affected the rest of my work.
```

In [6]:
```
with open("G:\My Drive\JSON\FipsToState.json", 'r') as f:
    data_new = json.load(f)
    list3=list()
for j in list3:
    x = j[2]
    y = data_new[x]
    j.append(y)
    list3.append(j)
print("County Names Including State Names FIRST50: \n", list3[0:50])
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [6], in <cell line: 1>()
      1 with open("G:\My Drive\JSON\FipsToState.json", 'r') as f:
----> 2     data_new = json.load(f)
      3     list3=list()
      4 for j in list3:

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

In [86]:
```
s1=set()
for i in list3:
    x = i[0] + ", " + i[4]
    s1.add(x)
```

In [105…
```
area2=dict()
for i in list3:
    x = i[4]
    y = area2.get(x)
    if y == None:
        area2[x] = 1
    else:
        area2[x] = area2[x] + 1
print("Number of Counties by State \n", area2)
```

```
Number of Counties by State
 {'Alabama': 13, 'Alaska': 10, 'Arkansas': 16, 'California': 8, 'Colorado': 16, 'Flor
ida': 13, 'Geogia': 35, 'Connecticut': 1, 'Arizona': 2, 'Hawaii': 1, 'Idaho': 7, 'Ill
inois': 24, 'Indiana': 19, 'Iowa': 25, 'Kansas': 22, 'Kentucky': 27, 'Louisiana': 11,
'Minnesota': 18, 'Mississippi': 19, 'Maine': 2, 'Maryland': 10, 'Massachusetts': 1,
'Michigan': 24, 'Missouri': 24, 'Montana': 9, 'Nebraska': 22, 'Nevada': 3, 'New Jerse
y': 3, 'New Mexico': 7, 'North Carolina': 20, 'Ohio': 19, 'Oklahoma': 16, 'North Dako
ta': 10, 'New York': 11, 'Pennsylvania': 13, 'South Carolina': 8, 'Tennessee': 19, 'S
outh Dakota': 14, 'Oregon': 5, 'Texas': 60, 'Utah': 1, 'Virginia': 18, 'Washington':
2, 'Wisconsin': 13, 'Wyoming': 4}
```

In [103…
```
# Task 2 Find the three most common names of the counties.
# Derive the numbers of counties that use these three names, respectively.
# For each of state list their county name and state code.
# Firstly, I will sort it according the lenght of the list.

area1 = dict()
count = 0
for i in list3:
    x = i[4]
    z = i[0]
    if area1.get(z) == None:
        area1[z] = 1
    else:
        area1[z] = area1[z] + 1
```

i and j are variables, before and after iteration. area1, area2, area3, area4, area5 and area6 are dictionaries. list1, list2, list3, list4, list5 and rest6 are lists.

SOME KEY FEATURES AS A RESULT OF MY LEARNING PROCESS:

Dictionaries are collections of keys and values. They are commonly used because they are a very fast data type. Dictionary is used for counting items, or for updating data (example: stock

change prices). Key for dictionary has to be unique and immutable. Only immutable data type can be used as a key. Touple, bolean, float, integers and string can be used as a key. List can not be used as a key, becuse is a MUTABLE data type.

In [104…
```
def v(j):
    return (j[1])
most_common_names=list()
t1 = sorted(d1.items(), key=v, reverse=True)
for j in range(3):
    most_common_names.append(t1[j])
top_3_states=list()
for i in most_common_names:
    match = i[0]
    count = i[1]
    if count == match:
        x = (county + ", " + state)
        top_3_states.append(x)
print("Top Three Most Common Names of Counties: \n", top_3_states)
```

```
Top Three Most Common Names of Counties:
 []
```

In [99]:
```
# Task 3 Basic statistics by state
# For each state, find
# 1. The number of counties
# 2. The name and size (census area) of the biggest and smallest county by area
# 3. The total and average area of counties
```

In [111…
```
# In the following step I will find out the total area for each state
area1 = dict()
for i in list3:
    x = i[4]
    z = i[3]
    y = area1.get(x)
    if y == None:
        area1[x] = 0
    else:
        area1[x] = area1[x] + z
print("Total Areas by State", area1)
```

```
Total Areas by State {'Alabama': 8244.83, 'Alaska': 167580.365, 'Arkansas': 11217.65
4, 'California': 10669.657, 'Colorado': 16766.835, 'Florida': 8167.63, 'Geogia': 1205
6.258000000002, 'Connecticut': 0, 'Arizona': 9200.143, 'Hawaii': 0, 'Idaho': 5206.004
000000001, 'Illinois': 11532.470000000001, 'Indiana': 6549.177999999998, 'Iowa': 1302
5.745999999997, 'Kansas': 17830.647999999997, 'Kentucky': 8732.735000000002, 'Louisia
na': 6626.47, 'Minnesota': 10697.002, 'Mississippi': 10977.736, 'Maine': 2562.66, 'Ma
ryland': 3220.9469999999997, 'Massachusetts': 0, 'Michigan': 15641.853000000001, 'Mis
souri': 14095.439, 'Montana': 12722.475, 'Nebraska': 13471.195, 'Nevada': 12628.49199
9999998, 'New Jersey': 506.59799999999996, 'New Mexico': 25815.21, 'North Carolina':
9560.64, 'Ohio': 8499.892, 'Oklahoma': 13294.977999999997, 'North Dakota': 10280.519,
'New York': 6266.007, 'Pennsylvania': 8159.923999999999, 'South Carolina': 4460.676,
'Tennessee': 7724.8600000000015, 'South Dakota': 13262.705000000002, 'Oregon': 12375.
305, 'Texas': 54033.15300000002, 'Utah': 0, 'Virginia': 3082.3149999999996, 'Washingt
on': 1242.171, 'Wisconsin': 9913.431999999999, 'Wyoming': 7006.29}
```

In [3]:
```
# In this step I will count an average county area by state
area4 = dict()
for i in list3:
    x = i[4]
```

```
    y = area4.get(x)
    if y == None:
        area4[x] = "No Data"
    else:
        area4[x] = area3[x] / area2[x]
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [3], in <cell line: 3>()
      1 # In this step I will count an average county area by state
      2 area4 = dict()
----> 3 for i in list3:
      4     x = i[4]
      5     y = area4.get(x)

NameError: name 'list3' is not defined
```

In [117…

```
# The Smallest Area Counties by State
area5=dict()
list5 = list()
for i in list3:
    a = i[3]
    x = i[4]
    z = i[0]
    y = area5.get(x)
    if y == None:
        area5[x] = 0
    else:
        area5[x] = a
    if float(area5[x]) > a:
        area5[x] = a
for j in list3:
    a = j[3]
    x = j[4]
    z = j[0]
    if area5[x] == a:
        list5.append([x, z, a])
    else:
        continue
print("The Smallest Counties by State: \n", list5)
```

```
The Smallest Counties by State:
 [['Alabama', 'Talladega', 736.775], ['California', 'Yuba', 631.839], ['Colorado', 'W
eld', 3987.238], ['Florida', 'Leon', 666.852], ['Alaska', 'Wade Hampton', 17081.433],
['Arizona', 'Maricopa', 9200.143], ['Arkansas', 'Searcy', 666.095], ['Idaho', 'Teto
n', 449.456], ['Illinois', 'Woodford', 527.799], ['Indiana', 'DeKalb', 362.824], ['Ge
ogia', 'Washington', 678.452], ['Kentucky', 'Morgan', 381.127], ['Iowa', 'Worth', 40
0.123], ['Kansas', 'Butler', 1429.863], ['Louisiana', 'Winn', 950.086], ['Maine', 'Wa
shington', 2562.66], ['Maryland', 'Talbot', 268.538], ['Michigan', 'Wexford', 565.00
2], ['Minnesota', 'Kandiyohi', 796.785], ['Nevada', 'White Pine', 8875.648], ['New Je
rsey', 'Passaic', 184.593], ['Montana', 'Wheatland', 1423.195], ['Nebraska', 'McPhers
on', 858.976], ['Mississippi', 'Washington', 724.741], ['Missouri', 'Hickory', 399.09
1], ['North Carolina', 'Yadkin', 334.829], ['North Dakota', 'Wells', 1271.047], ['Ohi
o', 'Fayette', 406.357], ['New Mexico', 'Socorro', 6646.679], ['New York', 'Yates', 3
38.143], ['South Carolina', 'Saluda', 452.778], ['South Dakota', 'Ziebach', 1961.27
2], ['Oklahoma', 'Woodward', 1242.399], ['Oregon', 'Wheeler', 1714.749], ['Pennsylvan
ia', 'Armstrong', 653.203], ['Tennessee', 'Wilson', 570.826], ['Texas', 'Nolan', 911.
997], ['Virginia', 'Waynesboro', 15.039], ['Washington', 'Franklin', 1242.171], ['Wis
consin', 'Wood', 793.116], ['Wyoming', 'Weston', 2398.089]]
```

In [118…

```
# The Largest Area Counties by State
area6=dict()
list6 = list()
for i in list3:
    a = i[3]
    x = i[4]
    z = i[0]
    y = area6.get(x)
    if y == None:
        area6[x] = 0
    else:
        area6[x] = a
    if float(area6[x]) < a:
        area6[x] = a
for j in list3:
    a = j[3]
    x = j[4]
    z = j[0]
    if area6[x] == a:
        list6.append([x, z, a])
print("The Largest Area Counties by State: \n", list6)
```

```
The Largest Area Counties by State:
 [['Alabama', 'Talladega', 736.775], ['California', 'Yuba', 631.839], ['Colorado', 'W
eld', 3987.238], ['Connecticut', 'Windham', 512.91], ['Florida', 'Leon', 666.852],
['Alaska', 'Wade Hampton', 17081.433], ['Arizona', 'Maricopa', 9200.143], ['Arkansa
s', 'Searcy', 666.095], ['Hawaii', 'Maui', 1161.521], ['Idaho', 'Teton', 449.456],
['Illinois', 'Woodford', 527.799], ['Indiana', 'DeKalb', 362.824], ['Geogia', 'Washin
gton', 678.452], ['Kentucky', 'Morgan', 381.127], ['Iowa', 'Worth', 400.123], ['Kansa
s', 'Butler', 1429.863], ['Louisiana', 'Winn', 950.086], ['Maine', 'Washington', 256
2.66], ['Maryland', 'Talbot', 268.538], ['Massachusetts', 'Franklin', 699.319], ['Mic
higan', 'Wexford', 565.002], ['Minnesota', 'Kandiyohi', 796.785], ['Nevada', 'White P
ine', 8875.648], ['New Jersey', 'Passaic', 184.593], ['Montana', 'Wheatland', 1423.19
5], ['Nebraska', 'McPherson', 858.976], ['Mississippi', 'Washington', 724.741], ['Mis
souri', 'Hickory', 399.091], ['North Carolina', 'Yadkin', 334.829], ['North Dakota',
'Wells', 1271.047], ['Ohio', 'Fayette', 406.357], ['New Mexico', 'Socorro', 6646.67
9], ['New York', 'Yates', 338.143], ['South Carolina', 'Saluda', 452.778], ['South Da
kota', 'Ziebach', 1961.272], ['Oklahoma', 'Woodward', 1242.399], ['Oregon', 'Wheele
r', 1714.749], ['Pennsylvania', 'Armstrong', 653.203], ['Utah', 'Davis', 298.778],
['Tennessee', 'Wilson', 570.826], ['Texas', 'Nolan', 911.997], ['Virginia', 'Waynesbo
ro', 15.039], ['Washington', 'Franklin', 1242.171], ['Wisconsin', 'Wood', 793.116],
['Wyoming', 'Weston', 2398.089]]
```