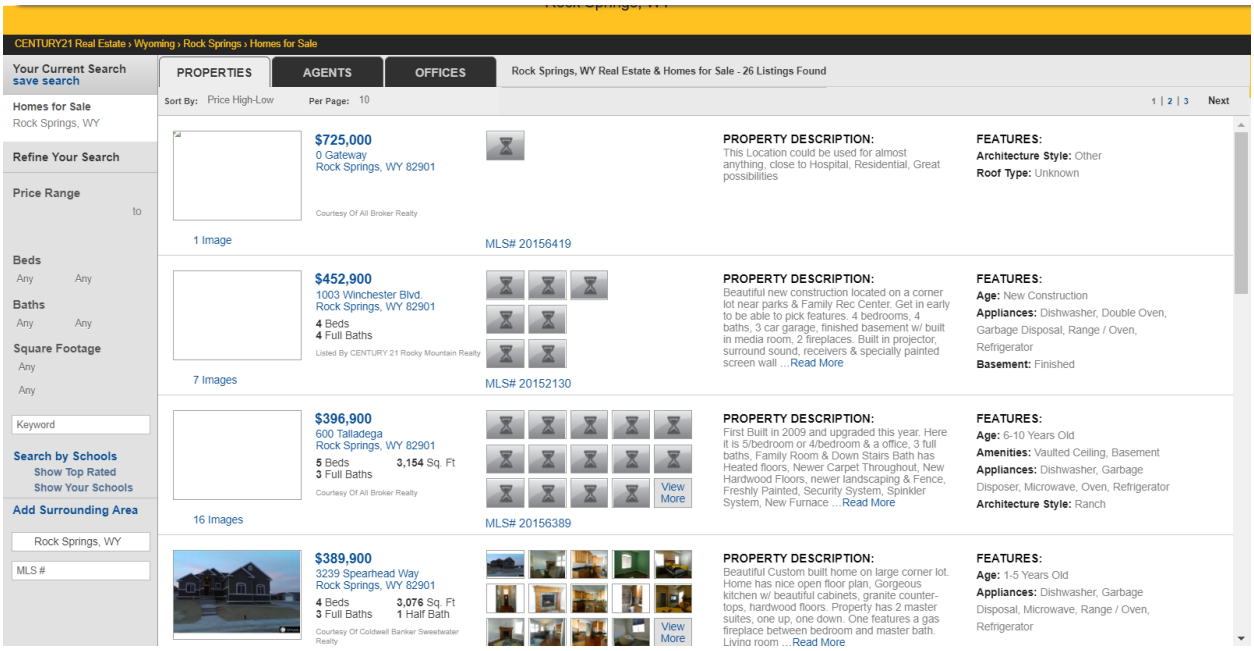
Web Scrapping Project

Introduction:

In this project, I scrape the real estate property data from the **Century 21 website** using Python **HTTP library Requests** and the **Beautiful Soup package.**

The data looks like-



I am going to extract the following attributes-

1) Address
2) Area
3) Locality
4) Beds
5) Full Baths
6) Half Baths
7) Lot size
8) Price

Beautiful Soup:

- Beautiful Soup is a HTML parser which efficiently parses the HTML and XML documents.
- It creates a parse tree for parsed pages and is mainly used to extract the website data from HTML.

Requests:

The requests library is used for making HTTP requests in Python.

1) **Installing packages:**

pip install bs4
pip install requests

2) **Importing the packages:**

```
import requests
from bs4 import BeautifulSoup
```

3) **Requests:**
To load the entire, inspect element script of a website in python, I used the requests library. Requests library allows to give python a URL by making HTTP requests and grab the web content.

```
In [3]: r = requests.get("http://www.pyclass.com/real-estate/rock-springs-wy/LCWYROCKSPRINGS/",
                         headers={'User-agent': 'Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:61.0) Gecko/20100101 Firefox/61.0'})
```

requests.get() is used to make a GET request in order to get or retrieve data from the website. In this case, the web page source code is loaded to the variable r.

4) **Request Headers:**
The request headers let us pass a dictionary of HTTP headers to get response to the request using the "headers" parameter in the request's method.

5) **Content:**
The response of the GET request has some valuable information and this information is stored in a variety of different formats. To see the response content in the **bytes** format we use **.content.**
The content is grabbed from the request data type and stored in another variable c.

```
In [5]: c = r.content
        type(c)

Out[5]: bytes
```

Output of c:

```
In [6]: c

Out[6]: b'<!DOCTYPE html>\n<!-- saved from url=(0110)http://web.archive.org/web/20160127020422/http://www.century21.com/real-estate/r
ock-springs-wy/LCWYROCKSPRINGS -->\n<html lang="en" style="margin: 0px;overflow:hidden"><script async="" src="./LCWYROCKSPRIN
GS1_files/beacon.js"></script><script src="chrome-extension://pkljnnogdmlajgaoodihioopfdkpgjgg/Kernel.js?0.3685073930846756">
</script><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8">\n\n\n<script type="text/javascript" src
="./LCWYROCKSPRINGS1_files/analytics.js"></script>\n<script type="text/javascript">archive_analytics.values.server_name="wwwb
-app17.us.archive.org";archive_analytics.values.server_ms=227;</script>\n<link type="text/css" rel="stylesheet" href="./LCWYR
OCKSPRINGS1_files/banner-styles.css">\n\n\n      <title>Rock Springs Real Estate | Find Houses &amp; Homes for Sale in Rock S
prings, WY</title>\n\n      <meta name="title" content="Rock Springs Real Estate | Find Houses &amp; Homes for Sale in Rock S
prings, WY">\n      <meta name="description" content="Search Rock Springs real estate property listings to find homes for sal
e in Rock Springs, WY. Browse houses for sale in Rock Springs today!">\n      <meta name="keywords" content="Rock Springs rea
l estate, Rock Springs homes, Rock Springs homes for sale, Rock Springs properties, Rock Springs listings, Rock Springs house
s for sale, WY real estate, WY homes, WY homes for sale, WY properties, WY listings, WY houses for sale">\n\n\n\n      <scrip
t>if(window.innerWidth && (innerWidth<769)){location.href+=(location.href.match("\\\\?"))?"&v=0":"?v=0";}</script>\n\n\n
<link rel="alternate" hreflang="en-US" href="http://www.century21.com/real-estate/rock-springs-wy/LCWYROCKSPRINGS/">\n\n\n
<link rel="alternate" hreflang="es" href="http://espanol.century21.com/propiedades-en-venta/rock-springs-wy/LCWYROCKSPRING
S/">\n\n\n\n\n\n\n                <link rel="canonical" href="http://www.century21.com/real-estate/rock-springs-wy/LCWYROCKSPRIN
GS/">\n\n\n\n\n\n         <link rel="publisher" href="https://plus.google.com/+Century21">\n\n      <meta http-equiv="X-UA-Comp
atible" content="IE=8, IE=9">\n\n          <link rel="stylesheet" href="./LCWYROCKSPRINGS1_files/advanceLiquidMapCSS.css">\n\n
\n      <link rel="stylesheet" href="./LCWYROCKSPRINGS1_files/liquidmappingPhone.css" media="screen and (max-width: 767px)">
```

The output of c is scrambled and so we use Beautiful Soup. The Beautiful Soup library creates a parse tree, and this helps in making the webpage content more readable. This parse tree is created from the parsed page using the python's built-in **html.parser.** This is assigned to the variable **soup.**

```
In [7]: soup = BeautifulSoup(c, "html.parser")
```

```
In [20]: soup

<title>Rock Springs Real Estate | Find Houses &amp; Homes for Sale in Rock Springs, WY</title>
<meta content="Rock Springs Real Estate | Find Houses &amp; Homes for Sale in Rock Springs, WY" name="title"/>
<meta content="Search Rock Springs real estate property listings to find homes for sale in Rock Springs, WY. Browse houses fo
r sale in Rock Springs today!" name="description"/>
<meta content="Rock Springs real estate, Rock Springs homes, Rock Springs homes for sale, Rock Springs properties, Rock Sprin
gs listings, Rock Springs houses for sale, WY real estate, WY homes, WY homes for sale, WY properties, WY listings, WY houses
for sale" name="keywords"/>
<script>if(window.innerWidth && (innerWidth<769)){location.href+=(location.href.match("\\?"))?"&v=0":"?v=0";}</script>
<link href="http://www.century21.com/real-estate/rock-springs-wy/LCWYROCKSPRINGS/" hreflang="en-US" rel="alternate"/>
<link href="http://espanol.century21.com/propiedades-en-venta/rock-springs-wy/LCWYROCKSPRINGS/" hreflang="es" rel="alternat
e"/>
<link href="http://www.century21.com/real-estate/rock-springs-wy/LCWYROCKSPRINGS/" rel="canonical"/>
<link href="https://plus.google.com/+Century21" rel="publisher"/>
<meta content="IE=8, IE=9" http-equiv="X-UA-Compatible"/>
<link href="./LCWYROCKSPRINGS1_files/advanceLiquidMapCSS.css" rel="stylesheet"/>
<link href="./LCWYROCKSPRINGS1_files/liquidmappingPhone.css" media="screen and (max-width: 767px)" rel="stylesheet"/>
<script>


window.m_aDeferredMethods = [];
```

The Beautiful Soup parse tree is then formatted to a Unicode string using the Beautiful Soup's **prettify()** method.

```
In [8]: print(soup.prettify())
```

```
<!DOCTYPE html>
<!-- saved from url=(0110)http://web.archive.org/web/20160127020422/http://www.century21.com/real-estate/rock-springs-wy/LCW
YROCKSPRINGS -->
<html lang="en" style="margin: 0px;overflow:hidden">
 <script async="" src="./LCWYROCKSPRINGS1_files/beacon.js">
 </script>
 <script src="chrome-extension://pkljnnogdmlajgaoodihioopfdkpgjgg/Kernel.js?0.3685073930846756">
 </script>
 <head>
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
  <script src="./LCWYROCKSPRINGS1_files/analytics.js" type="text/javascript">
  </script>
  <script type="text/javascript">
   archive_analytics.values.server_name="wwwb-app17.us.archive.org";archive_analytics.values.server_ms=227;
  </script>
  <link href="./LCWYROCKSPRINGS1_files/banner-styles.css" rel="stylesheet" type="text/css"/>
  <title>
   Rock Springs Real Estate | Find Houses &amp; Homes for Sale in Rock Springs, WY
  </title>
  <meta content="Rock Springs Real Estate | Find Houses &amp; Homes for Sale in Rock Springs, WY" name="title"/>
```

6) find_all()

The find_all method is used to find the tags from the inspect element script of the webpage to scrape that particular data.

```
In [9]: all = soup.find_all("div", {"class":"propertyRow"})
        all
```

```
Out[9]: [<div class="propertyRow" id="propertyRowREN021201395" onclick="Track.doEvent('Hybrid Mapping', 'Property Center Lane', 'Sele
ct a property with brand REN to view details'); document.location.href='/property/0-gateway-rock-springs-wy-82901-REN02120139
5';">
 <div class="CenterLaneCardBg CardWrapper propertyCard" id="propertyREN021201395">
 <div class="CenterLaneCard propertyCard">
 <div class="CardThumb">
 <div class="landscapeThumbContainer">
 <a href="http://web.archive.org/web/20160127020422/http://www.century21.com/property/0-gateway-rock-springs-wy-82901-REN0212
01395"><img class="lazyLoad" rel="http://www.century21.com/listhub/thumb/130x90/photos.listhub.net/FCBRWY/20156419/1?lm=20151
103T230938" src="./LCWYROCKSPRINGS1_files/1"/></a>
 </div>
 </div>
 <h4 class="propPrice">


            $725,000
```

This returns all classes with values “propertyRow”

```
In [14]: all[0]
```

```
Out[14]: <div class="propertyRow" id="propertyRowREN021201395" onclick="Track.doEvent('Hybrid Mapping', 'Property Center Lane', 'Selec
t a property with brand REN to view details'); document.location.href='/property/0-gateway-rock-springs-wy-82901-REN02120139
5';">
<div class="CenterLaneCardBg CardWrapper propertyCard" id="propertyREN021201395">
<div class="CenterLaneCard propertyCard">
<div class="CardThumb">
<div class="landscapeThumbContainer">
<a href="http://web.archive.org/web/20160127020422/http://www.century21.com/property/0-gateway-rock-springs-wy-82901-REN02120
1395"><img class="lazyLoad" rel="http://www.century21.com/listhub/thumb/130x90/photos.listhub.net/FCBRWY/20156419/1?lm=201511
03T230938" src="./LCWYROCKSPRINGS1_files/1"/></a>
</div>
</div>
<h4 class="propPrice">


            $725,000


<span class="IconPropertyFavorite16"></span>
```

```
In [15]: all[0].find_all("h4", {"class":"propPrice"})

Out[15]: [<h4 class="propPrice">

                    $725,000


            <span class="IconPropertyFavorite16"></span>
        </h4>]
```

This just returns the property price by accessing the h4 tag having the class name "propPrice" from the div tag having the class name "propertyRow".

7) .text

.text ignores the HTML syntax and only returns the text part of the output.

```
In [10]: all[0].find("h4", {"class":"propPrice"}).text

Out[10]: '\n\n\n            $725,000\n\n\n\n      \n'
```

The type() method gives me the data type of price.

```
In [11]: type(all[0].find("h4", {"class":"propPrice"}).text)

Out[11]: str
```

Since it is string, I used the string methods to remove the spaces and the next line characters.

```
In [15]: all[0].find("h4", {"class":"propPrice"}).text.replace("\n", "").replace(" ","")

Out[15]: '$725,000'
```

Then I grabbed the page number of the last page to find out how many pages I need to scrape using the following code.

```
In [17]: page_num = soup.find_all("a",{"class":"Page"})[-1].text #grab the last page
         print(page_num)
         print(type(page_num))

         3
         <class 'str'>
```

8) Next, I tried to print the property price of all 10 property listings on the first page using the for loop.

```
In [18]: for item in all:
             print(item.find("h4", {"class":"propPrice"}).text.replace("\n", "").replace(" ",""))
```

```
$725,000
$452,900
$396,900
$389,900
$254,000
$252,900
$210,000
$209,000
$199,900
$196,900
```

9) Similarly, I find the other attributes that need to be scrapped from each listing.
There are multiple occurrences of the tag so if I use find then it will just return the first occurrence so I use the find_all() method. Also, the "propAddressCollapse" class has two attributes- address and locality. In order to grab both I print both positions 0 and 1.

```
In [19]: for item in all:
             print(item.find("h4", {"class":"propPrice"}).text.replace("\n", "").replace(" ",""))
             print(item.find_all("span",{"class":"propAddressCollapse"})[0].text)
             print(item.find_all("span",{"class":"propAddressCollapse"})[1].text)
```

```
$725,000
0 Gateway
Rock Springs, WY 82901
$452,900
1003 Winchester Blvd.
Rock Springs, WY 82901
$396,900
600 Talladega
Rock Springs, WY 82901
$389,900
3239 Spearhead Way
Rock Springs, WY 82901
$254,000
522 Emerald Street
Rock Springs, WY 82901
$252,900
1302 Veteran's Drive
Rock Springs, WY 82901
$210,000
1021 Cypress Cir
Rock Springs, WY 82901
$209,000
913 Madison Dr
Rock Springs, WY 82901
$199,900
1344 Teton Street
Rock Springs, WY 82901
$196,900
4 Minnies Lane
Rock Springs, WY 82901
```

10) Then I scrape the No. of beds using the "infoBed" class. However, this gives an error since there are "None" values.

```
In [21]: for item in all:
             print(item.find("h4", {"class":"propPrice"}).text.replace("\n", "").replace(" ",""))
             print(item.find_all("span",{"class":"propAddressCollapse"})[0].text)
             print(item.find_all("span",{"class":"propAddressCollapse"})[1].text)
             print(item.find("span",{"class":"infoBed"}).text)
             print(" ")
```

```
$725,000
0 Gateway
Rock Springs, WY 82901

---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-21-5367e75125dd> in <module>()
      3     print(item.find_all("span",{"class":"propAddressCollapse"})[0].text)
      4     print(item.find_all("span",{"class":"propAddressCollapse"})[1].text)
----> 5     print(item.find("span",{"class":"infoBed"}).text)
      6     print(" ")

AttributeError: 'NoneType' object has no attribute 'text'
```

To fix this I use try-catch error handling method.

```
In [22]: for item in all:
             print(item.find("h4", {"class":"propPrice"}).text.replace("\n", "").replace(" ",""))
             print(item.find_all("span",{"class":"propAddressCollapse"})[0].text)
             print(item.find_all("span",{"class":"propAddressCollapse"})[1].text)
             try:
                 print(item.find("span",{"class":"infoBed"}).text)
             except:
                 pass
             print(" ")
```

```
$725,000
0 Gateway
Rock Springs, WY 82901

$452,900
1003 Winchester Blvd.
Rock Springs, WY 82901
4 Beds

$396,900
600 Talladega
Rock Springs, WY 82901
5 Beds

$389,900
3239 Spearhead Way
Rock Springs, WY 82901
4 Beds

$254,000
522 Emerald Street
Rock Springs, WY 82901
3 Beds

$252,900
1302 Veteran's Drive
```

This works but gives an output “4 Beds”. I want this column entry to be of integer datatype. So, I extract from the <b> tag inside the tag. Also, instead of just “pass” in the except block I print “None” to maintain uniformity throughout.

In [23]:
```
for item in all:
    print(item.find("h4", {"class":"propPrice"}).text.replace("\n", "").replace(" ",""))
    print(item.find_all("span",{"class":"propAddressCollapse"})[0].text)
    print(item.find_all("span",{"class":"propAddressCollapse"})[1].text)
    try:
        print(item.find("span",{"class":"infoBed"}).find("b").text)
    except:
        print("None")
    print(" ")
```

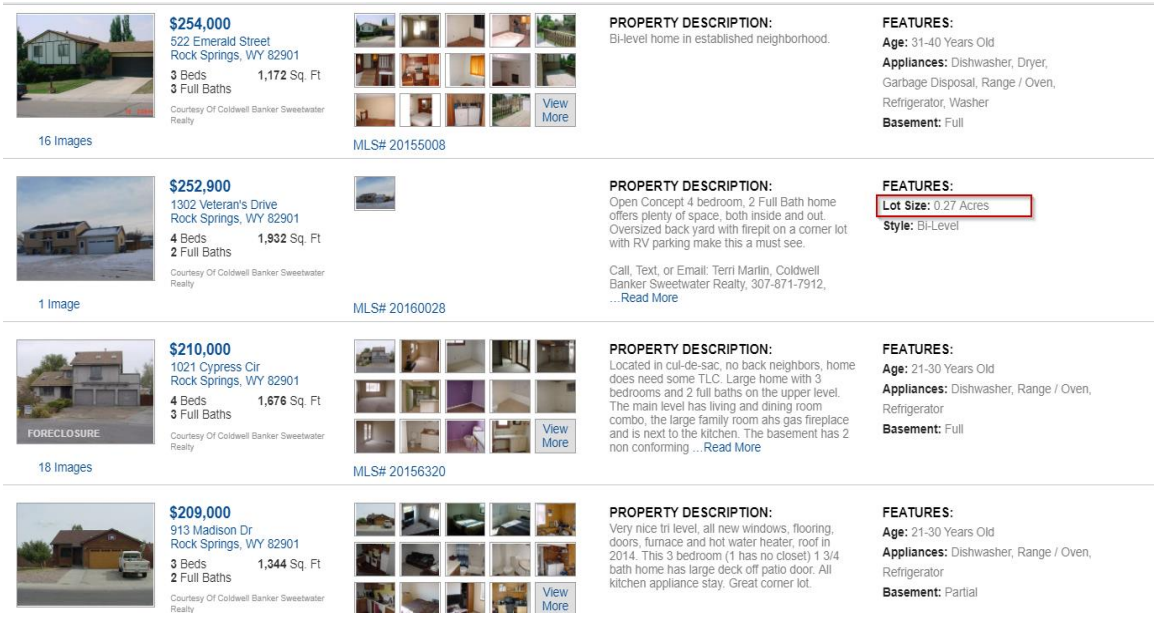
```
$725,000
0 Gateway
Rock Springs, WY 82901
None

$452,900
1003 Winchester Blvd.
Rock Springs, WY 82901
4

$396,900
600 Talladega
Rock Springs, WY 82901
5

$389,900
3239 Spearhead Way
Rock Springs, WY 82901
4

$254,000
522 Emerald Street
Rock Springs, WY 82901
3

$252,900
```

11) Similarly, I scrape the values for square feet info, no. of full baths and no. of half baths.

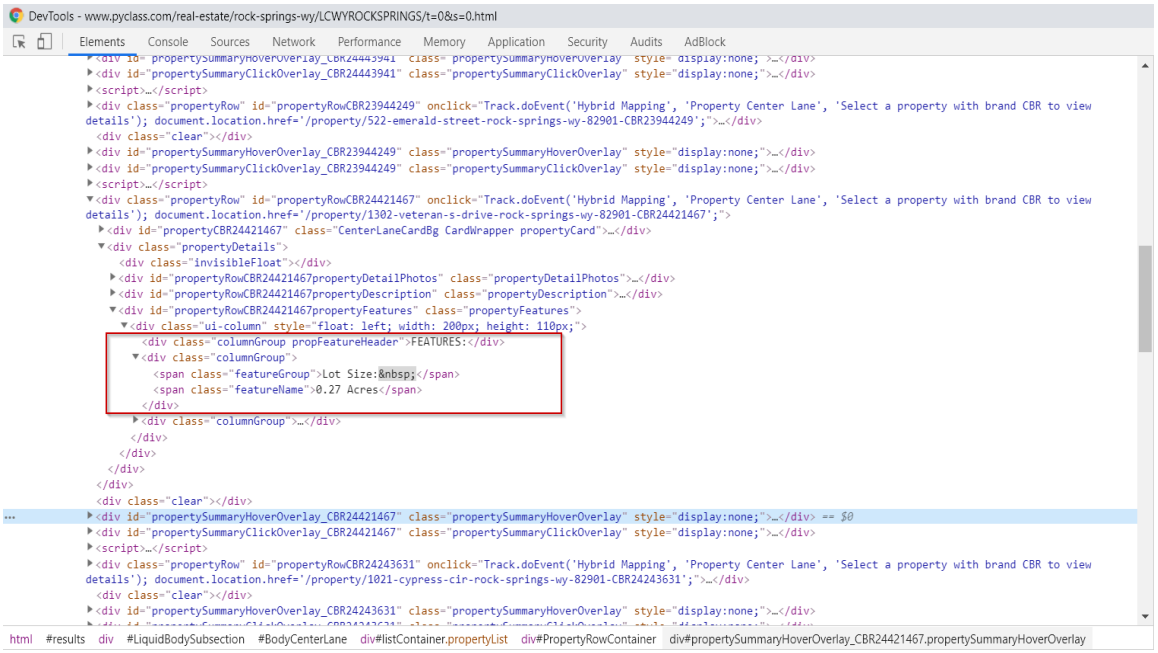
In [24]:
```
for item in all:
    print(item.find("h4", {"class":"propPrice"}).text.replace("\n", "").replace(" ",""))
    print(item.find_all("span",{"class":"propAddressCollapse"})[0].text)
    print(item.find_all("span",{"class":"propAddressCollapse"})[1].text)
    try:
        print(item.find("span",{"class":"infoBed"}).find("b").text)
    except:
        print("None")
    try:
        print(item.find("span",{"class":"infoSqFt"}).find("b").text)
    except:
        print("None")

    try:
        print(item.find("span",{"class":"infoValueFullBath"}).find("b").text)
    except:
        print("None")

    try:
        print(item.find("span",{"class":"infoValueHalfBath"}).find("b").text)
    except:
        print("None")
    print(" ")
```

```
$725,000
0 Gateway
Rock Springs, WY 82901
None
None
None
None

$452,900
1003 Winchester Blvd.
Rock Springs, WY 82901
```

12) Then I find the "Lot Size".



"Lot Size" is a part of "Features" and "Features" are referred as "Feature group" and "Feature name".



```python
try:
    print(item.find("span",{"class":"infoValueHalfBath"}).find("b").text)
except:
    print("None")
for column_group in item.find_all("div", {"class":"columnGroup"}):
    for feature_group, feature_name in zip(column_group.find_all("span",{"class":"featureGroup"}),
                                           column_group.find_all("span",{"class":"featureName"})):
        if "Lot Size" in feature_group.text:
            print(feature_name.text)
print(" ")
```

```
$725,000
0 Gateway
Rock Springs, WY 82901
None
None
None
None

$452,900
1003 Winchester Blvd.
Rock Springs, WY 82901
4
None
4
None
0.21 Acres

$396,900
600 Talladega
Rock Springs, WY 82901
5
3,154
3
None
```

13) Now I need all this data in the form of panda's data frame. One solution could be to iterate through the data frame but that is a very costly and time-consuming solution. Instead its better to create a data frame out of a python dictionary or a list of python dictionaries. In this case I would need 10 dictionaries to store the key-value pairs. I also need to store these dictionaries somewhere in order to access them later. So, I store them in a list. So, I will start every iteration with a empty dictionary so that I can add key-value pairs to an empty dictionary. Once the empty dictionary is created, I replace all the print statements in the for loop.

```
In [27]: l = []
for item in all:
    d = {}
    d["Address"]= item.find_all("span",{"class":"propAddressCollapse"})[0].text
    d["Locality"]= item.find_all("span",{"class":"propAddressCollapse"})[1].text
    d["Price"]= item.find("h4", {"class":"propPrice"}).text.replace("\n", "").replace(" ","")
    try:
        d["Beds"]= item.find("span",{"class":"infoBed"}).find("b").text
    except:
        d["Beds"]= None

    try:
        d["Area"]= item.find("span",{"class":"infoSqFt"}).find("b").text
    except:
        d["Area"]= None

    try:
        d["Full Baths"]= item.find("span",{"class":"infoValueFullBath"}).find("b").text
    except:
        d["Full Baths"]= None

    try:
        d["Half Baths"]= item.find("span",{"class":"infoValueHalfBath"}).find("b").text
    except:
        d["Half Baths"]= None

    for column_group in item.find_all("div", {"class":"columnGroup"}):
            for feature_group, feature_name in zip(column_group.find_all("span",{"class":"featureGroup"}),
                                                   column_group.find_all("span",{"class":"featureName"})):
                if "Lot Size" in feature_group.text:
                    d["Lot Size"]= feature_name.text
    l.append(d)
```

```
In [28]: l
```

```
Out[28]: [{'Address': '0 Gateway',
           'Locality': 'Rock Springs, WY 82901',
           'Price': '$725,000',
           'Beds': None,
           'Area': None,
           'Full Baths': None,
           'Half Baths': None},
          {'Address': '1003 Winchester Blvd.',
           'Locality': 'Rock Springs, WY 82901',
           'Price': '$452,900',
           'Beds': '4',
           'Area': None,
           'Full Baths': '4',
           'Half Baths': None,
           'Lot Size': '0.21 Acres'},
          {'Address': '600 Talladega',
           'Locality': 'Rock Springs, WY 82901',
           'Price': '$396,900',
           'Beds': '5',
           'Area': '3,154',
           'Full Baths': '3',
           'Half Baths': None},
          {'Address': '3239 Spearhead Way',
           'Locality': 'Rock Springs, WY 82901',
           'Price': '$389,900',
           'Beds': '4',
           'Area': '3,076',
           'Full Baths': '3',
           'Half Baths': '1',
           'Lot Size': 'Under 1/2 Acre, '},
          {'Address': '522 Emerald Street',
           'Locality': 'Rock Springs, WY 82901',
           'Price': '$254,000',
```

The length of the list is 10 which is the number of listings on the page 1.

```
In [29]: len(l)
```

```
Out[29]: 10
```

14) Then I store this scrapped data in a panda's data frame.

```
In [30]: import pandas as pd
         df = pd.DataFrame(l)
```

```
In [31]: df
```

Out[31]:

	Address	Area	Beds	Full Baths	Half Baths	Locality	Lot Size	Price
0	0 Gateway	None	None	None	None	Rock Springs, WY 82901	NaN	$725,000
1	1003 Winchester Blvd.	None	4	4	None	Rock Springs, WY 82901	0.21 Acres	$452,900
2	600 Talladega	3,154	5	3	None	Rock Springs, WY 82901	NaN	$396,900
3	3239 Spearhead Way	3,076	4	3	1	Rock Springs, WY 82901	Under 1/2 Acre,	$389,900
4	522 Emerald Street	1,172	3	3	None	Rock Springs, WY 82901	Under 1/2 Acre,	$254,000
5	1302 Veteran's Drive	1,932	4	2	None	Rock Springs, WY 82901	0.27 Acres	$252,900
6	1021 Cypress Cir	1,676	4	3	None	Rock Springs, WY 82901	Under 1/2 Acre,	$210,000
7	913 Madison Dr	1,344	3	2	None	Rock Springs, WY 82901	Under 1/2 Acre,	$209,000
8	1344 Teton Street	1,920	3	2	None	Rock Springs, WY 82901	Under 1/2 Acre,	$199,900
9	4 Minnies Lane	1,664	3	2	None	Rock Springs, WY 82901	2.02 Acres	$196,900

15) Now, I need to do this same procedure to extract the data from all 3 pages. In order to do that I loop through all 3 pages and find a pattern in the web URL.

Following are the URL's of the three pages-

www.pyclass.com/real-estate/rock-springs-wy/LCWYROCKSPRINGS/t=0&s=0.html

www.pyclass.com/real-estate/rock-springs-wy/LCWYROCKSPRINGS/t=0&s=10.html

www.pyclass.com/real-estate/rock-springs-wy/LCWYROCKSPRINGS/t=0&s=20.html

There is a pattern in the URL. The "s" part changes by 10 for each page. Thus, our base URL will be-

```
base_url = "http://www.pyclass.com/real-estate/rock-springs-wy/LCWYROCKSPRINGS/t=0&s="
```

I applied a for loop to iterate through the three pages using the pattern found.

```
In [27]: l = []
for page in range(0,int(page_num)*10,10):
    print(base_url+str(page)+".html")
    r = requests.get(base_url+str(page)+".html", headers={'User-agent': 'Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:61.0)
                                                          Gecko/20100101 Firefox/61.0'})
    c = r.content
    soup = BeautifulSoup(c, "html.parser")
    all = soup.find_all("div", {"class":"propertyRow"})

    for item in all:
        d = {}
        d["Address"]= item.find_all("span",{"class":"propAddressCollapse"})[0].text
        d["Locality"]= item.find_all("span",{"class":"propAddressCollapse"})[1].text
        d["Price"]= item.find("h4", {"class":"propPrice"}).text.replace("\n", "").replace(" ","")
        try:
            d["Beds"]= item.find("span",{"class":"infoBed"}).find("b").text
        except:
            d["Beds"]= None

        try:
            d["Area"]= item.find("span",{"class":"infoSqFt"}).find("b").text
        except:
            d["Area"]= None

        try:
            d["Full Baths"]= item.find("span",{"class":"infoValueFullBath"}).find("b").text
        except:
            d["Full Baths"]= None

        try:
            d["Half Baths"]= item.find("span",{"class":"infoValueHalfBath"}).find("b").text
        except:
            d["Half Baths"]= None

        for column_group in item.find_all("div", {"class":"columnGroup"}):
```

This gives the URL's of all 3 pages.

```
http://www.pyclass.com/real-estate/rock-springs-wy/LCWYROCKSPRINGS/t=0&s=0.html
http://www.pyclass.com/real-estate/rock-springs-wy/LCWYROCKSPRINGS/t=0&s=10.html
http://www.pyclass.com/real-estate/rock-springs-wy/LCWYROCKSPRINGS/t=0&s=20.html
```

```
In [36]: l

Out[36]: [{'Address': '0 Gateway',
           'Locality': 'Rock Springs, WY 82901',
           'Price': '$725,000',
           'Beds': None,
           'Area': None,
           'Full Baths': None,
           'Half Baths': None},
          {'Address': '1003 Winchester Blvd.',
           'Locality': 'Rock Springs, WY 82901',
           'Price': '$452,900',
           'Beds': '4',
           'Area': None,
           'Full Baths': '4',
           'Half Baths': None,
           'Lot Size': '0.21 Acres'},
          {'Address': '600 Talladega',
           'Locality': 'Rock Springs, WY 82901',
           'Price': '$396,900',
           'Beds': '5',
           'Area': '3,154'
```

Now the length of list is 37 which is the exact number of listings we have in the 3 pages.

```
In [37]: len(l)

Out[37]: 37
```

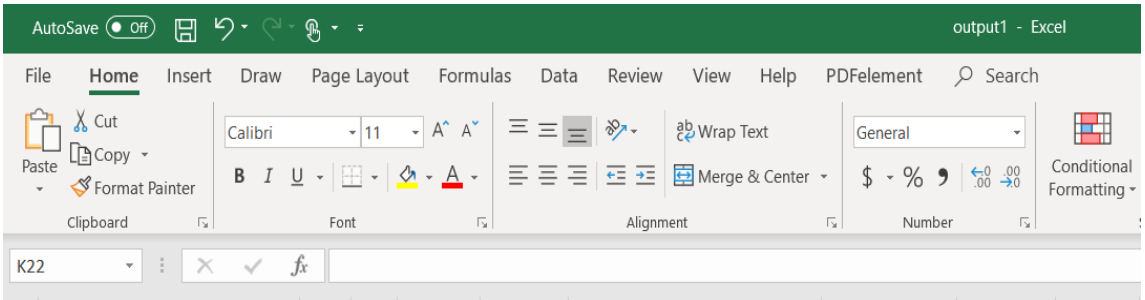
16) Importing the data in a panda's data frame.

```
In [77]: import pandas as pd
         df = pd.DataFrame(l)
```

	Address	Area	Beds	Full Baths	Half Baths	Locality	Lot Size	Price
0	0 Gateway	None	None	None	None	Rock Springs, WY 82901	NaN	$725,000
1	1003 Winchester Blvd.	None	4	4	None	Rock Springs, WY 82901	0.21 Acres	$452,900
2	600 Talladega	3,154	5	3	None	Rock Springs, WY 82901	NaN	$396,900
3	3239 Spearhead Way	3,076	4	3	1	Rock Springs, WY 82901	Under 1/2 Acre,	$389,900
4	522 Emerald Street	1,172	3	3	None	Rock Springs, WY 82901	Under 1/2 Acre,	$254,000
5	1302 Veteran's Drive	1,932	4	2	None	Rock Springs, WY 82901	0.27 Acres	$252,900
6	1021 Cypress Cir	1,676	4	3	None	Rock Springs, WY 82901	Under 1/2 Acre,	$210,000
7	913 Madison Dr	1,344	3	2	None	Rock Springs, WY 82901	Under 1/2 Acre,	$209,000
8	1344 Teton Street	1,920	3	2	None	Rock Springs, WY 82901	Under 1/2 Acre,	$199,900
9	4 Minnies Lane	1,664	3	2	None	Rock Springs, WY 82901	2.02 Acres	$196,900
10	9339 Sd 26900	2,560	None	None	None	Rocksprings, TX 78880	NaN	$1,700,000
11	RR674P13 Hwy 377	2,000	None	None	None	Rocksprings, TX 78880	NaN	$1,100,000
12	0 Hwy 41	None	None	None	None	Rocksprings, TX 78880	NaN	$1,080,000
13	9339 Sd 26900	2,560	None	None	None	Rocksprings, TX 78880	NaN	$908,350
14	CR450 Hwy 377	None	None	None	None	Rocksprings, TX 78880	NaN	$905,000
15	Cr 240 Cr 240	1,398	None	None	None	Rocksprings, TX 78880	NaN	$695,000
16	RR674 Hwy 377	1,738	None	None	None	Rocksprings, TX 78880	NaN	$605,000
17	9770a Sd 26900	1,080	None	None	None	Rocksprings, TX 78880	NaN	$559,805
18	Lot17 CR 2630	None	None	None	None	Rocksprings, TX 78880	NaN	$504,000
19	Tr12,16 CR 520	None	None	None	None	Rocksprings, TX 78880	NaN	$410,000
20	32575 S Shadow Mountain Road	2,318	3	2	None	Black Canyon City, AZ 85324	NaN	$299,900
21	32750 S Shangrila Drive	2,120	3	2	None	Black Canyon City, AZ 85324	NaN	$167,500
22	0000 Black Canyon Highway	None	None	None	None	Black Canyon City, AZ 85324	5 Acres	$150,000
23	34775 S CHOLLA Drive	1,220	3	2	None	Black Canyon City, AZ 85324	NaN	$129,500
24	33403 S. HA-WA-SI TERRACE	2,000	4	2	None	BLACK CANYON CITY, AZ 85324	NaN	$129,000
25	34263 S Bertha Street	2,260	5	2	None	Black Canyon City, AZ 85324	NaN	$80,000
26	33160 S Canyon Road	1,248	3	2	None	Black Canyon City, AZ 85324	NaN	$77,900
27	19421 E Todd Evans Road	1,404	3	2	None	Black Canyon City, AZ 85324	NaN	$70,500
28	18688 E AGUA Vista	None	None	None	None	Black Canyon City, AZ 85324	0.7 Acres	$70,000
29	50600 N Old Black Canyon Road	None	None	None	None	Black Canyon City, AZ 85324	3 Acres	$67,500
30	20101 E SQUAW VALLEY Road	None	None	None	None	Black Canyon City, AZ 85324	NaN	$54,900
31	33259 S Canyon Road	1,056	3	1	None	Black Canyon City, AZ 85324	NaN	$45,600
32	34558 S ROADRUNNER RD	784	2	1	None	Black Canyon City, AZ 85324	Under 1/2 Acre	$40,000
33	19260 E Scenic Loop Road	None	None	None	None	Black Canyon City, AZ 85324	2.35 Acres	$30,000
34	19000 E MAREN Avenue	None	None	None	None	Black Canyon City, AZ 85324	2.05 Acres	$29,000
35	19350 E SAGUARO Drive	None	None	None	None	Black Canyon City, AZ 85324	0.73 Acres	$28,995
36	20650 E Amethyst Place	None	None	None	None	Black Canyon City, AZ 85324	0.31 Acres	$15,000

17) Lastly, I store it in .csv format.

```
In [76]: df.to_csv("output1.csv")
```



	A	B	C	D	E	F	G	H	I
1	**Address**	**Area**	**Beds**	**Full Baths**	**Half Baths**	**Locality**	**Lot Size**	**Price**	
2	0 Gateway					Rock Springs, WY 82901		$725,000	
3	1003 Winchester Blvd.		4	4		Rock Springs, WY 82901	0.21 Acres	$452,900	
4	600 Talladega	3,154	5	3		Rock Springs, WY 82901		$396,900	
5	3239 Spearhead Way	3,076	4	3	1	Rock Springs, WY 82901	Under 1/2 Acre,	$389,900	
6	522 Emerald Street	1,172	3	3		Rock Springs, WY 82901	Under 1/2 Acre,	$254,000	
7	1302 Veteran's Drive	1,932	4	2		Rock Springs, WY 82901	0.27 Acres	$252,900	
8	1021 Cypress Cir	1,676	4	3		Rock Springs, WY 82901	Under 1/2 Acre,	$210,000	
9	913 Madison Dr	1,344	3	2		Rock Springs, WY 82901	Under 1/2 Acre,	$209,000	
10	1344 Teton Street	1,920	3	2		Rock Springs, WY 82901	Under 1/2 Acre,	$199,900	
11	4 Minnies Lane	1,664	3	2		Rock Springs, WY 82901	2.02 Acres	$196,900	
12	9339 Sd 26900	2,560				Rocksprings, TX 78880		$1,700,000	
13	RR674P13 Hwy 377	2,000				Rocksprings, TX 78880		$1,100,000	
14	0 Hwy 41					Rocksprings, TX 78880		$1,080,000	
15	9339 Sd 26900	2,560				Rocksprings, TX 78880		$908,350	
16	CR450 Hwy 377					Rocksprings, TX 78880		$905,000	
17	Cr 240 Cr 240	1,398				Rocksprings, TX 78880		$695,000	
18	RR674 Hwy 377	1,738				Rocksprings, TX 78880		$605,000	
19	9770a Sd 26900	1,080				Rocksprings, TX 78880		$559,805	
20	Lot17 CR 2630					Rocksprings, TX 78880		$504,000	
21	Tr12,16 CR 520					Rocksprings, TX 78880		$410,000	
22	32575 S Shadow Mountain Road	2,318	3	2		Black Canyon City, AZ 85324		$299,900	
23	32750 S Shangrila Drive	2,120	3	2		Black Canyon City, AZ 85324		$167,500	
24	0000 Black Canyon Highway					Black Canyon City, AZ 85324	5 Acres	$150,000	
25	34775 S CHOLLA Drive	1,220	3	2		Black Canyon City, AZ 85324		$129,500	
26	33403 S. HA-WA-SI TERRACE	2,000	4	2		BLACK CANYON CITY, AZ 85324		$129,000	
27	34263 S Bertha Street	2,260	5	2		Black Canyon City, AZ 85324		$80,000	
28	33160 S Canyon Road	1,248	3	2		Black Canyon City, AZ 85324		$77,900	
29	19421 E Todd Evans Road	1,404	3	2		Black Canyon City, AZ 85324		$70,500	