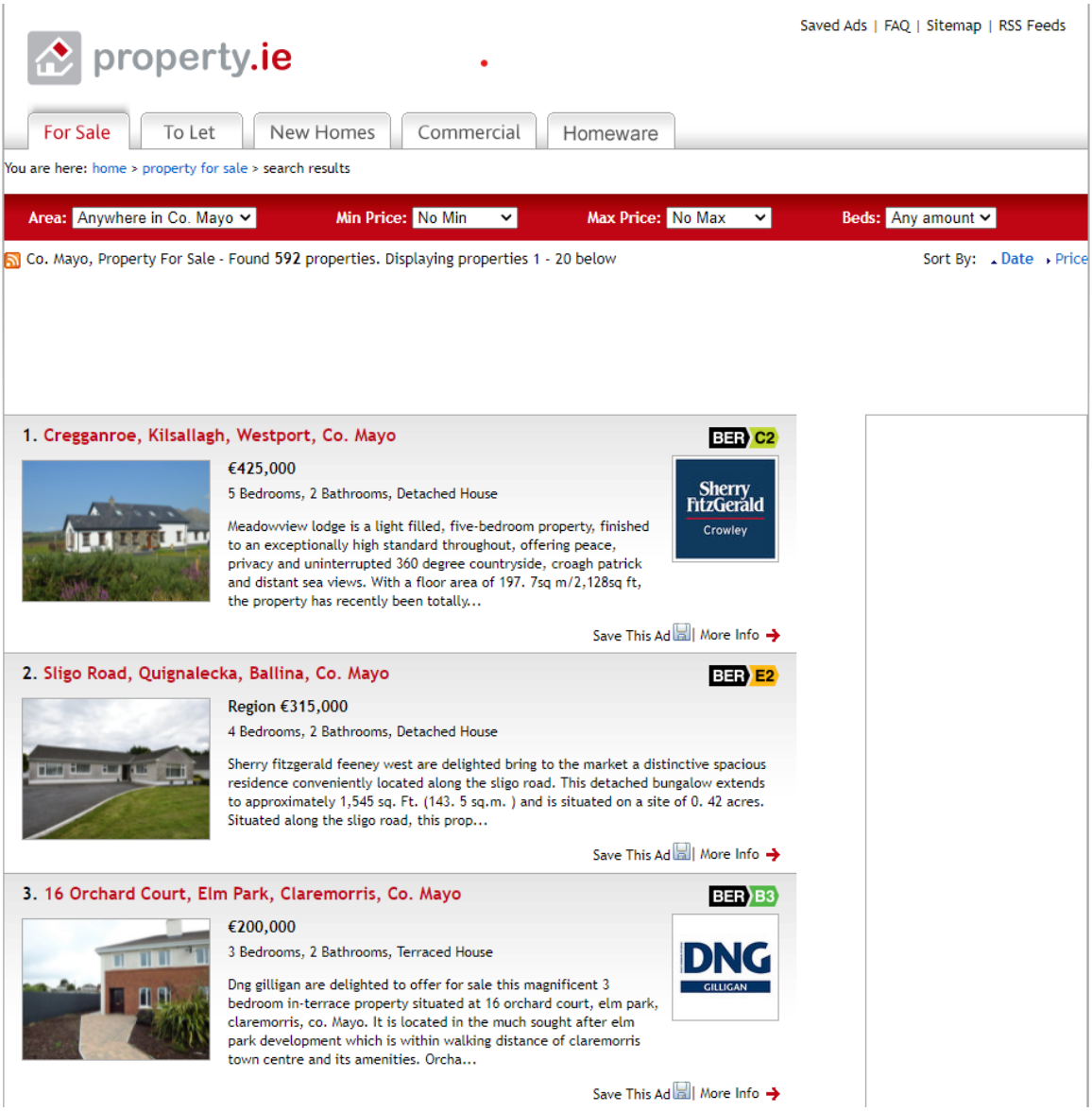
Data Representation
Lab 3b: Web scraping

Lecturer: Andrew Beatty

Extract the house prices and address for the properties for sale in Mayo
From the website https://www.property.ie/



And store the results into a tab delimited file

Prototyping

Ie checking we can do everything we need

1. Test that we can retrieve a web page from the web. save this file at PY01-testRequest.py in a folder called week03-webScraping.

```
import requests
from bs4 import BeautifulSoup
page = requests.get("http://dataquestio.github.io/web-
scraping-pages/simple.html")
print (page)
print("-------------------")
print (page.content)
```

2. Test that BeautifulSoup is installed by modifying the program to read.

```
import requests
from bs4 import BeautifulSoup
page = requests.get("http://dataquestio.github.io/web-
scraping-pages/simple.html")
#print (page)
#print("-------------------")
#print (page.content)
soup1 = BeautifulSoup(page.content, 'html.parser')
print (soup1.prettify())
```

3. Test that you can read a file, we will use the carviewer2.html file that we made last week should be up a directory and in the week02 folder ie ("../week02/carviewer2.html").

```
from bs4 import BeautifulSoup

with open("../week02/carviewer2.html") as fp:
    soup = BeautifulSoup(fp,'html.parser')

print (soup.prettify())
```

(If wish you can save another html file in same directory as this and remove the javascript, I will not be doing this, but it might make the html clearer for you).

4. Extract the first from the file (make a file called (PY03-readOutFile.py).

```
from bs4 import BeautifulSoup

with open("../week02/carviewer2.html") as fp:
    soup = BeautifulSoup(fp,'html.parser')

print (soup.tr)
```

5. This is the first . This is not what we want.
6. Modify the program to get all the <tr>

```
#print (soup.tr)
rows = soup.findAll("tr")
for row in rows:
    print("-------")
    print(row)
```

7. Now for each row let's get the contents of the <TD>

```
for row in rows:
    #print(row)
    dataList = []
    cols = row.findAll("td")
    for col in cols:
        print(col.text)
```

8. Modify this so that the text in the columns are stored in a list

```
    dataList = []
    cols = row.findAll("td")
    for col in cols:
        dataList.append(col.text)
    print (dataList)
```

Write to CSV

9. We want to write this to a CSV file for that we will need the csv package, lets test it. Write a file called PY04-testCSV.py.

```
import csv

employee_file = open('employee_file.csv', mode='w')
employee_writer = csv.writer(employee_file, delimiter=',', quotechar='"'
, quoting=csv.QUOTE_MINIMAL)

employee_writer.writerow(['John Smith', 'Accounting', 'November'])
employee_writer.writerow(['Erica Meyers, what', 'IT', 'March'])

employee_file.close()
```

10. Make a file called PY05-readFileFinal.py, copy in the code from PY03-readOutFile.py, that brings it all together.

```
from bs4 import BeautifulSoup
import csv

with open("../week02/carviewer2.html") as fp:
    soup = BeautifulSoup(fp,'html.parser')

#print (soup.tr)
employee_file = open('week02data.csv', mode='w')
employee_writer = csv.writer(employee_file, delimiter=',', quotechar='"'
, quoting=csv.QUOTE_MINIMAL)

rows = soup.findAll("tr")
for row in rows:

    cols = row.findAll("td")
    dataList = []
    for col in cols:
        dataList.append(col.text)
    employee_writer.writerow(dataList)
employee_file.close()
```

11. How would you modify the code so that the update and delete text is not outputted?

For real lets get the property prices

Explore the site:

1. Navigate to the page that has the properties for sale in mayo
https://www.property.ie/property-for-sale/mayo/

2. Find the data we want, ie the price, right click and select inspect element.
3. Note the element the price is in and the class of the div that that is in and the class of the div that contains all the data for this property

4. Check that the site does not have any dynamic functionality that might affect us, by right clicking the page and selecting view source. and checking that the elements and classes are the same.

5. Write a program called py06-myhome.py, that reads in from the URL you want.

https://www.property.ie/property-for-sale/mayo/

```
import requests
import csv
from bs4 import BeautifulSoup
url = "https://www.property.ie/property-for-sale/mayo/"
page = requests.get(url)

soup = BeautifulSoup(page.content, 'html.parser')
print(soup.prettify())
```

6. Check that the source that is outputted contains the divs we want
7. Modify the program to find the divs that contain each of the results (ie that have the class=”search_result”).

```
soup = BeautifulSoup(page.content, 'html.parser')
#print(soup.prettify())
listings = soup.findAll("div", class_="search_result")

for listing in listings:
    print(listing)
```

8. Now let’s find the price for each.

```
price = listing.find(class_="sresult_description").find("h3").text
print(price)
```

9. Now let’s find the address for each

```
I am leaving this to you
```

10. Now lets put the results into a teb delimitated file, NOTE the delimiter is set to "\t"

```
import requests
import csv
from bs4 import BeautifulSoup
url = "https://www.property.ie/property-for-sale/mayo/"
page = requests.get(url)

soup = BeautifulSoup(page.content, 'html.parser')
property_file = open('week03property.csv', mode='w')
property_writer = csv.writer(property_file, delimiter='\t',
                        quotechar='"', quoting=csv.QUOTE_MINIMAL)

listings = soup.findAll("div", class_="search_result")

for listing in listings:
    entryList = []
    address = listing.find(class_="sresult_address").find("h2").find("a"
).text
    entryList.append(address)
    price = listing.find(class_="sresult_description").find("h3").text
    entryList.append(price)

    property_writer.writerow(entryList)
property_file.close()
```

11. Look at the file, I would strip the strings that contain the address and price (use strip())

That's it, take a break