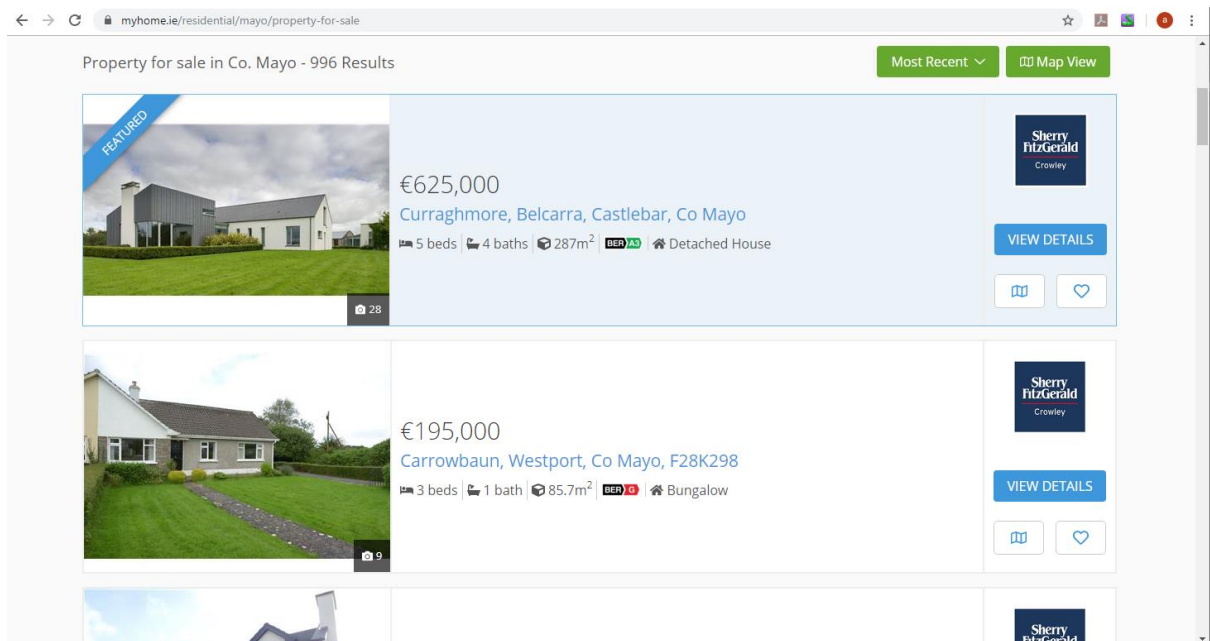


# Data Representation

## Lab 3: Web Scraping

Lecturer: Andrew Beatty

Extract house prices from myhome.ie and store in a TabSV :



Will be stored in a file like

Price	address
625,000	Curraghmore, Belcarra, Castlebar, Co Mayo
195,000	Croghan, Killala, Mayo

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
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("-----")
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)

## Reading the data from our html file

4. Extract the first <tr> 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)
```

This is the first <tr>. This is not what we want

5. Modify the program to get all the <tr>

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

6. Now for each row let's get the contents of the TD

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

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

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

8. 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', 'IT', 'March'])

employee_file.close()
```

Look at the directory and check if an employee\_file.csv was made

### Bring it together

9. 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()
```

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

Tricky bit: Read data from myhome.ie

11. Open myhome.ie in a browser and search for houses for sale in a county, you will see that the search results are in multiple pages navigate to the second page and note the URL

<https://www.myhome.ie/residential/mayo/property-for-sale?page=1>

12. Create a file called `py06-myhome.py`, and write the code to read the page. (this may take a little time to run)

```
<import requests

from bs4 import BeautifulSoup
page = requests.get("https://www.myhome.ie/residential/
mayo/property-for-sale?page=1")

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

13. Look at the do a view page source, search for some text you recognise, I did a search for “belcarra” and looked at the containing divs and saw that the listing are in a containing div with class=“`PropertyListingCard`”

```

66=class="SearchResults_Properties container">
67=
68=
69=
70=
71=
72=
73=
74=
75=
76=
77=
78=
79=
80=
81=
82=
83=
84=
85=
86=
87=
88=
89=
90=
91=
92=
93=
94=
95=
96=
97=
98=
99=
100=
101=
102=
103=
104=
105=
106=
107=
108=
109=
110=
111=
112=
113=
114=
115=
116=
117=
118=
119=
120=
121=
122=
123=
124=
125=
126=
127=
128=
129=
130=
131=
132=
133=
134=
135=
136=
137=
138=
139=
140=
141=
142=
143=
144=
145=
146=
147=
148=
149=
150=
151=
152=
153=
154=
155=
156=
157=
158=
159=
160=
161=
162=
163=
164=
165=
166=
167=
168=
169=
170=
171=
172=
173=
174=
175=
176=
177=
178=
179=
180=
181=
182=
183=
184=
185=
186=
187=
188=
189=
190=
191=
192=
193=
194=
195=
196=
197=
198=
199=
200=
201=
202=
203=
204=
205=
206=
207=
208=
209=
210=
211=
212=
213=
214=
215=
216=
217=
218=
219=
220=
221=
222=
223=
224=
225=
226=
227=
228=
229=
230=
231=
232=
233=
234=
235=
236=
237=
238=
239=
240=
241=
242=
243=
244=
245=
246=
247=
248=
249=
250=
251=
252=
253=
254=
255=
256=
257=
258=
259=
260=
261=
262=
263=
264=
265=
266=
267=
268=
269=
270=
271=
272=
273=
274=
275=
276=
277=
278=
279=
280=
281=
282=
283=
284=
285=
286=
287=
288=
289=
290=
291=
292=
293=
294=
295=
296=
297=
298=
299=
300=
301=
302=
303=
304=
305=
306=
307=
308=
309=
310=
311=
312=
313=
314=
315=
316=
317=
318=
319=
320=
321=
322=
323=
324=
325=
326=
327=
328=
329=
330=
331=
332=
333=
334=
335=
336=
337=
338=
339=
340=
341=
342=
343=
344=
345=
346=
347=
348=
349=
350=
351=
352=
353=
354=
355=
356=
357=
358=
359=
360=
361=
362=
363=
364=
365=
366=
367=
368=
369=
370=
371=
372=
373=
374=
375=
376=
377=
378=
379=
380=
381=
382=
383=
384=
385=
386=
387=
388=
389=
390=
391=
392=
393=
394=
395=
396=
397=
398=
399=
400=
401=
402=
403=
404=
405=
406=
407=
408=
409=
410=
411=
412=
413=
414=
415=
416=
417=
418=
419=
420=
421=
422=
423=
424=
425=
426=
427=
428=
429=
430=
431=
432=
433=
434=
435=
436=
437=
438=
439=
440=
441=
442=
443=
444=
445=
446=
447=
448=
449=
450=
451=
452=
453=
454=
455=
456=
457=
458=
459=
460=
461=
462=
463=
464=
465=
466=
467=
468=
469=
470=
471=
472=
473=
474=
475=
476=
477=
478=
479=
480=
481=
482=
483=
484=
485=
486=
487=
488=
489=
490=
491=
492=
493=
494=
495=
496=
497=
498=
499=
500=
501=
502=
503=
504=
505=
506=
507=
508=
509=
510=
511=
512=
513=
514=
515=
516=
517=
518=
519=
520=
521=
522=
523=
524=
525=
526=
527=
528=
529=
530=
531=
532=
533=
534=
535=
536=
537=
538=
539=
540=
541=
542=
543=
544=
545=
546=
547=
548=
549=
550=
551=
552=
553=
554=
555=
556=
557=
558=
559=
560=
561=
562=
563=
564=
565=
566=
567=
568=
569=
570=
571=
572=
573=
574=
575=
576=
577=
578=
579=
580=
581=
582=
583=
584=
585=
586=
587=
588=
589=
590=
591=
592=
593=
594=
595=
596=
597=
598=
599=
600=
601=
602=
603=
604=
605=
606=
607=
608=
609=
610=
611=
612=
613=
614=
615=
616=
617=
618=
619=
620=
621=
622=
623=
624=
625=
626=
627=
628=
629=
630=
631=
632=
633=
634=
635=
636=
637=
638=
639=
640=
641=
642=
643=
644=
645=
646=
647=
648=
649=
650=
651=
652=
653=
654=
655=
656=
657=
658=
659=
660=
661=
662=
663=
664=
665=
666=
667=
668=
669=
670=
671=
672=
673=
674=
675=
676=
677=
678=
679=
680=
681=
682=
683=
684=
685=
686=
687=
688=
689=
690=
691=
692=
693=
694=
695=
696=
697=
698=
699=
700=
701=
702=
703=
704=
705=
706=
707=
708=
709=
710=
711=
712=
713=
714=
715=
716=
717=
718=
719=
720=
721=
722=
723=
724=
725=
726=
727=
728=
729=
730=
731=
732=
733=
734=
735=
736=
737=
738=
739=
740=
741=
742=
743=
744=
745=
746=
747=
748=
749=
750=
751=
752=
753=
754=
755=
756=
757=
758=
759=
760=
761=
762=
763=
764=
765=
766=
767=
768=
769=
770=
771=
772=
773=
774=
775=
776=
777=
778=
779=
780=
781=
782=
783=
784=
785=
786=
787=
788=
789=
790=
791=
792=
793=
794=
795=
796=
797=
798=
799=
800=
801=
802=
803=
804=
805=
806=
807=
808=
809=
810=
811=
812=
813=
814=
815=
816=
817=
818=
819=
820=
821=
822=
823=
824=
825=
826=
827=
828=
829=
830=
831=
832=
833=
834=
835=
836=
837=
838=
839=
840=
841=
842=
843=
844=
845=
846=
847=
848=
849=
850=
851=
852=
853=
854=
855=
856=
857=
858=
859=
860=
861=
862=
863=
864=
865=
866=
867=
868=
869=
870=
871=
872=
873=
874=
875=
876=
877=
878=
879=
880=
881=
882=
883=
884=
885=
886=
887=
888=
88
```

14. Modify the code to retrieve the first `<div>` with `class="PropertyListingCard"`

```
listings = soup.find("div", class_="PropertyListingCard" )
print (listings)
```

15. Modify that code to get the price

```
class="PropertyListingCard__Price"
```

```
price = listings.find(class_="PropertyListingCard__Price").text  
  
print (price)
```

16. Also get the address

```
price = listings.find(class_="PropertyListingCard__Price").text  
  
print (price)
```

17. Now get all the entries in the page and store each in a list

```
listings = soup.findAll("div", class_="PropertyListingCard" )  
  
for listing in listings:  
    entry = []  
  
    price = listing.find(class_="PropertyListingCard__Price").text  
    entry.append(price)  
    address = listing.find(class_="PropertyListingCard__Address").text  
    entry.append(address)  
  
    print(entry)
```

18. Now output into a CSV with tabs as delimiters, (you can open this in excel if you wish)

```
import requests
import csv
from bs4 import BeautifulSoup
page = requests.get("https://www.myhome.ie/residential/mayo/property-for-sale?page=1")

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

home_file = open('week03MyHome.csv', mode='w')
home_writer = csv.writer(home_file, delimiter='\t', quotechar='\"', quoting=csv.QUOTE_MINIMAL)

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

for listing in listings:
    entryList = []

    price = listing.find(class_="PropertyListingCard__Price").text
    entryList.append(price)
    address = listing.find(class_="PropertyListingCard__Address").text
    entryList.append(address)

    home_writer.writerow(entryList)
home_file.close()
```