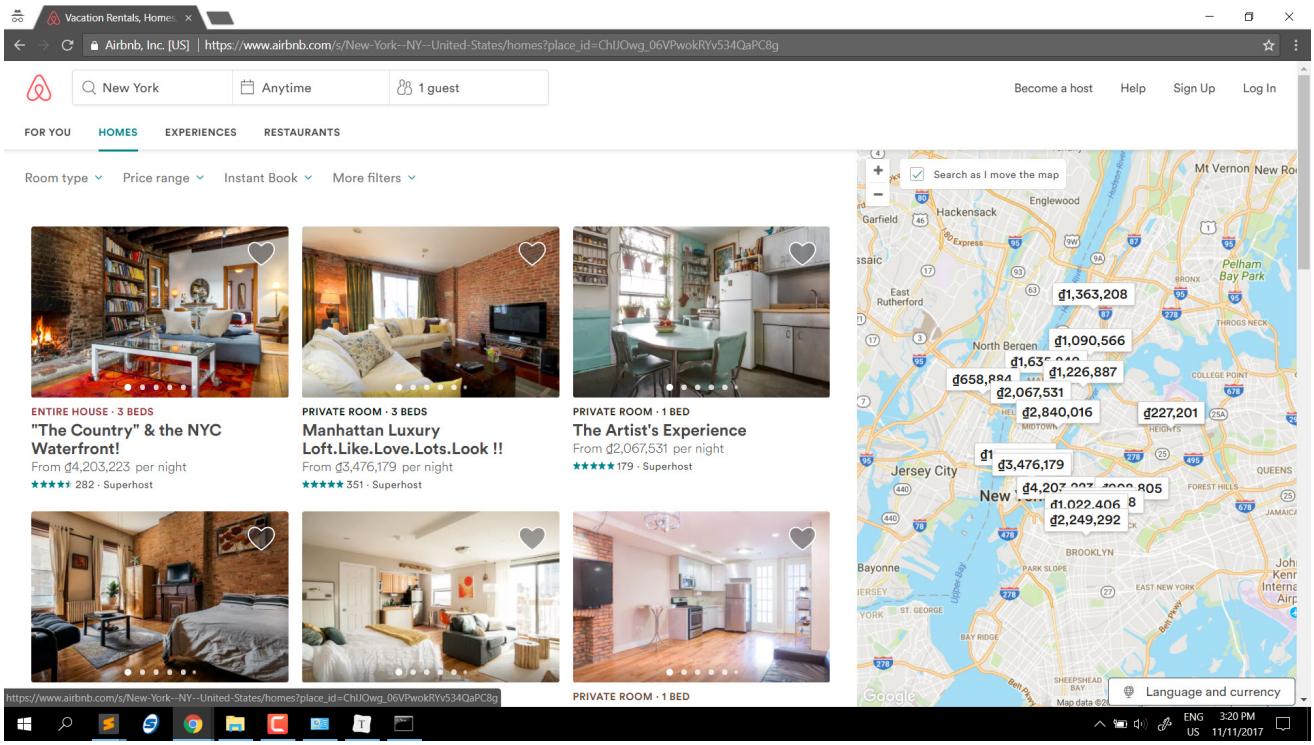
https://tanpham.org

Explain how to get data from **javascript, dynamic** site like AirBnB.

Objective

This tutorial will show you how to get housing data from AirBnb. Please access this link for example of housing data Newyork House for Rent from AirBnB



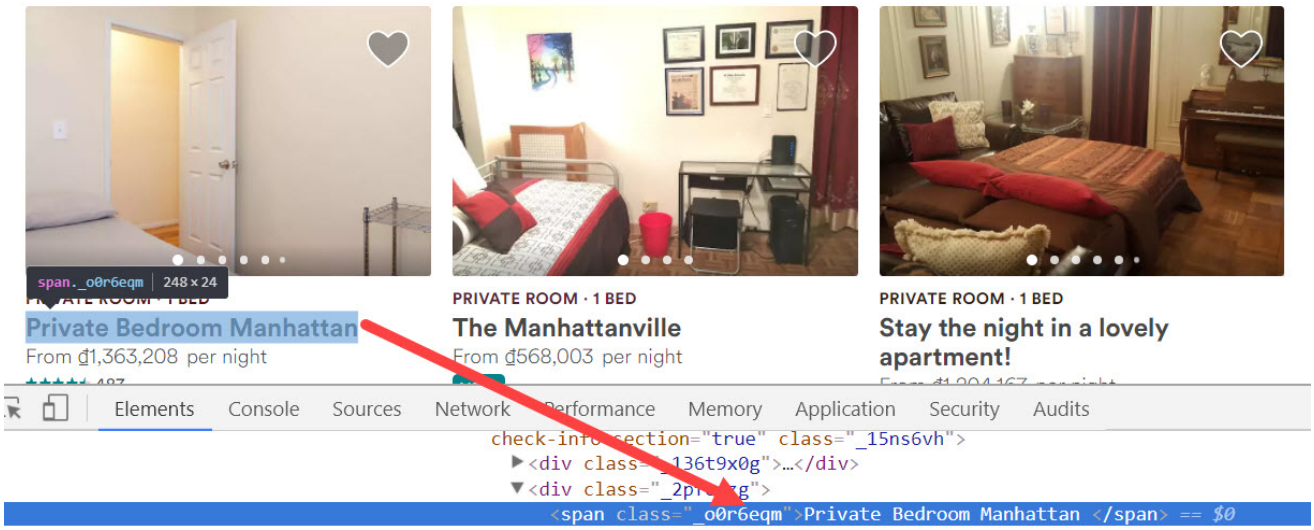
Dynamic Site with Javascript

Let start the `shell` to understand more about this page

```
scrapy shell
```

```
fetch('https://www.airbnb.com/s/New-York--NY--United-States/homes?
place_id=ChIJOwg_06VPwokRYv534QaPC8g')
```

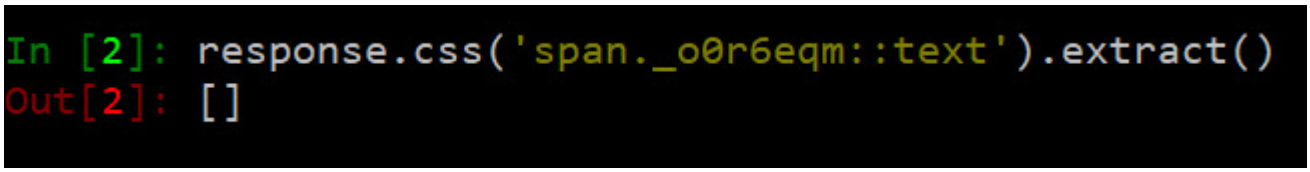
Do investigation with inspection tool, we see that to extract room title, we need to extract `span` tag which has class is `_o0r6eqm`



Let try this with `shell`

```
response.css('span._o0r6eqm::text').extract()
```

and you got NOTHING

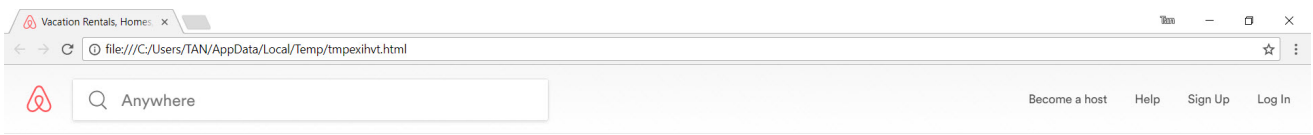


Why that ? What going on here ?

The important point is this site is a dynamic Javascript site, mean the content is generate dynamic at time of browser loading. The Scrapy request could not load this dynamic site. That is reason why we see nothing when we do query above.

Let try to see the HTML source which inside `response` by command, seem nothing in the page

```
view(response)
```



So how to deal with this kind of site ?

Selenium

One of solution is use library call `selenium` , this library has object call `webdriver` , this object allow us to load dynamic webpage as we load in a real browser and return HTML result as expected.

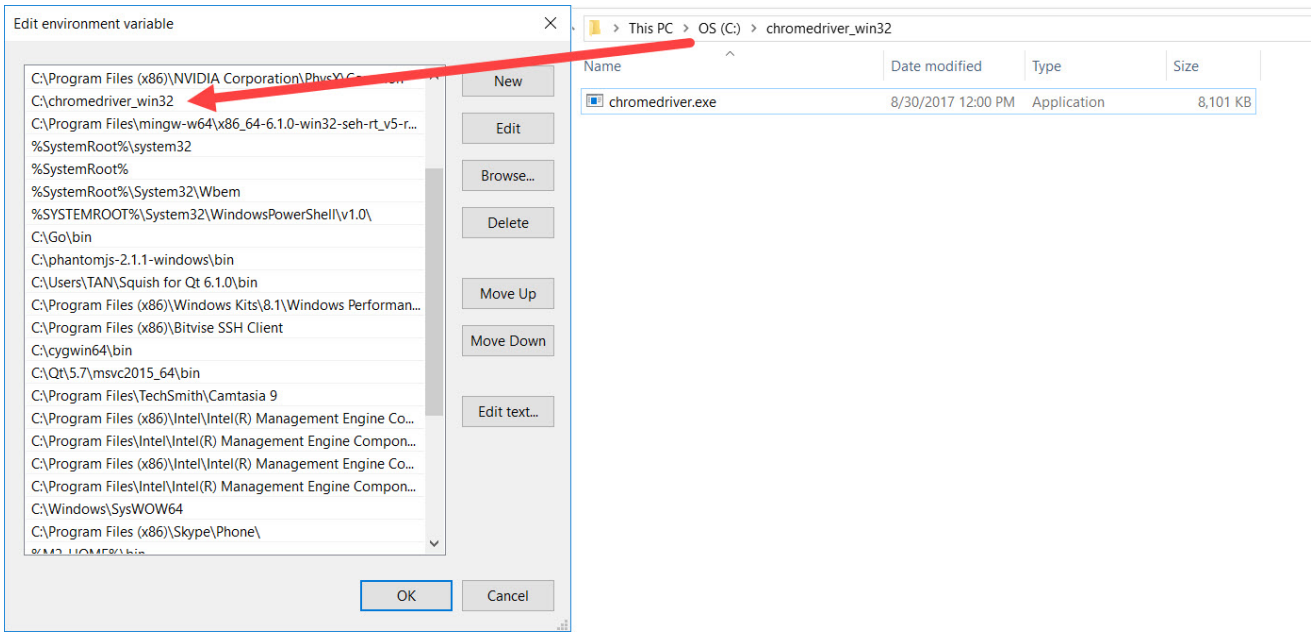
To install `selenium` with `pip` execute following command

```
pip install selenium
```

Google Chrome Driver

Selenium want to load webpage, it need browser driver, in this session you will know how to install **google chrome driver**. To download chrome driver, please access https://chromedriver.storage.googleapis.com/index.html?path=2.33/] (https://chromedriver.storage.googleapis.com/index.html?path=2.33/)

Save driver to local and then add the path to executable file to your **PATH** environment variable.



That it, you complete install **chrome driver**.

Build Spider Use Selenium and Chrome Driver

Start `airbnb` project

```
scrapy startproject airbnb
```

Change directory to `airbnb` then create `home` spider with basic type

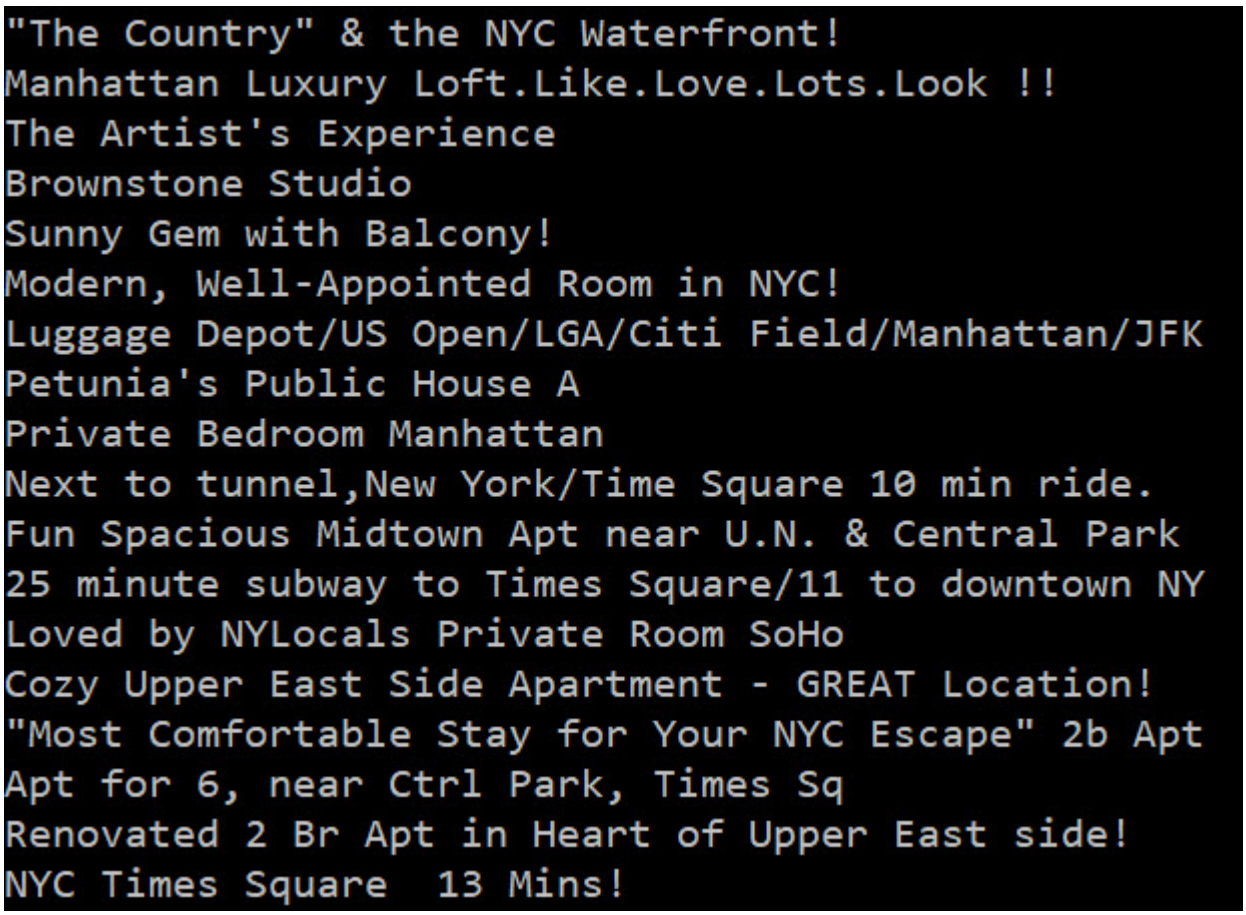
```
cd airbnb
scrapy genspider home
```

Change `home.py` with following code

```
# -*- coding: utf-8 -*-
import scrapy
from scrapy import Selector
# import webdriver
from selenium import webdriver

class HomeSpider(scrapy.Spider):

    name = 'home'
    allowed_domains = ['airbnb.com']
    start_urls = ['https://www.airbnb.com/s/New-York--NY--United-States/homes?
place_id=ChIJOwg_06VPwokRYv534QaPC8g&refinement_path=%2Fhomes&allow_override%5B%5D=&s_tag=pOehHf
Zr']

    def __init__(self):
        # init the driver with Chrome driver
        self.driver = webdriver.Chrome()


    def parse(self, response):
        # request the start url with chrome driver and all dynamic content is generate
        self.driver.get(response.url)
        # build Selector object for parsing
        sel = Selector(text=self.driver.page_source)
        for room in sel.css('span._o0r6eqm::text').extract():
            print room
```

Running the crawl and you can see Chrome open and request to page.

```
scrapy crawl home
```



Headless with PhantomJS

Now you do not want a Chrome browser pop up, have a thing call `PhantomJS` which also a browser but it running with out UI. To download PhantomJS go to this link http://phantomjs.org/download.html , put phantomjs.exe some where and add it's path to **PATH** environment variable.

In spider, instead of Chrome, we will use PhantomJS as follow:

```
# -*- coding: utf-8 -*-
import scrapy
from scrapy import Selector
# import webdriver
from selenium import webdriver

class HomeSpider(scrapy.Spider):

    name = 'home'
    allowed_domains = ['airbnb.com']
    start_urls = ['https://www.airbnb.com/s/New-York--NY--United-States/homes?
place_id=ChIJOwg_06VPwokRYv534QaPC8g&refinement_path=%2Fhomes&allow_override%5B%5D=&s_tag=pOehHf
Zr']

    def __init__(self):
        # headless with PhantomJS
        self.driver = webdriver.PhantomJS()


    def parse(self, response):
        # request the start url with chrome driver and all dynamic content is generate
        self.driver.get(response.url)
        # build Selector object for parsing
        sel = Selector(text=self.driver.page_source)
        for room in sel.css('span._o0r6eqm::text').extract():
            print room
```

Now you could get same information with out see any browser open up.