# 1.0 Introduction

This project is to develop a simple Python web crawler which periodically scraps the news tittles and corresponding url to json file. Major steps are described in section 2.0. Basically, the procedures can be divided into 2 parts. For part one, build a project in scrapy and change the behaviour in spider according to the data required. For part 2, it will show how to scrap the website locally in window system.

The targeted website is CNBC (<https://www.cnbc.com/world/?region=world>) and the reason of using Scrapy is because it provides a complete framework to scrap.

# 1.1 Objective

Write a simple Python web crawler to retrieve latest news from a news portal periodically.

# 1.2 Tools

* Python 3.6.0
* Scrapy 1.4.0
* Window Task Scheduler
* Sublime Text 3.0
* Pretty JSON

# 2.0 Step 1 : Start project for Scrapy

After deciding CNBC to strap, we have to start working with command prompt (cmd). Generally, two command prompts will be used parallelly, one (scrapy shell) is to check the xpath validation to see whether the xpath command is able to strap the required data and the other is to manipulate the scrapy (e.g. To start a project, generate spider etc.)

$ scrapy shell <https://www.cnbc.com/world/?region=world>

Insert $response to check whether it captured the entire HTML successfully



<200> indicating it is captured successfully

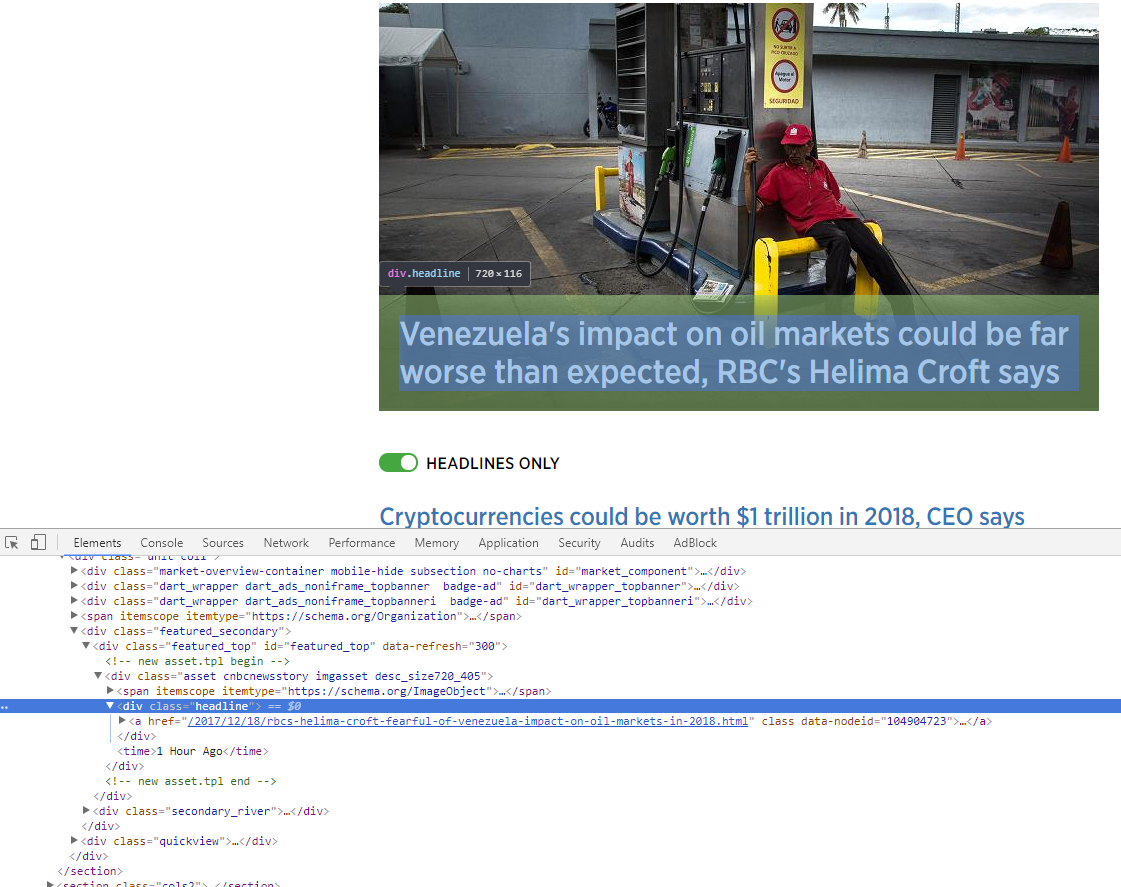
Next, start inspecting the xpath wanted and create project for scrapy (another cmd).

$cd C:\Users\onewa\Desktop\InfoTrie\

$scrapy startproject exercise *#Create a Scrapy project named exercise*.

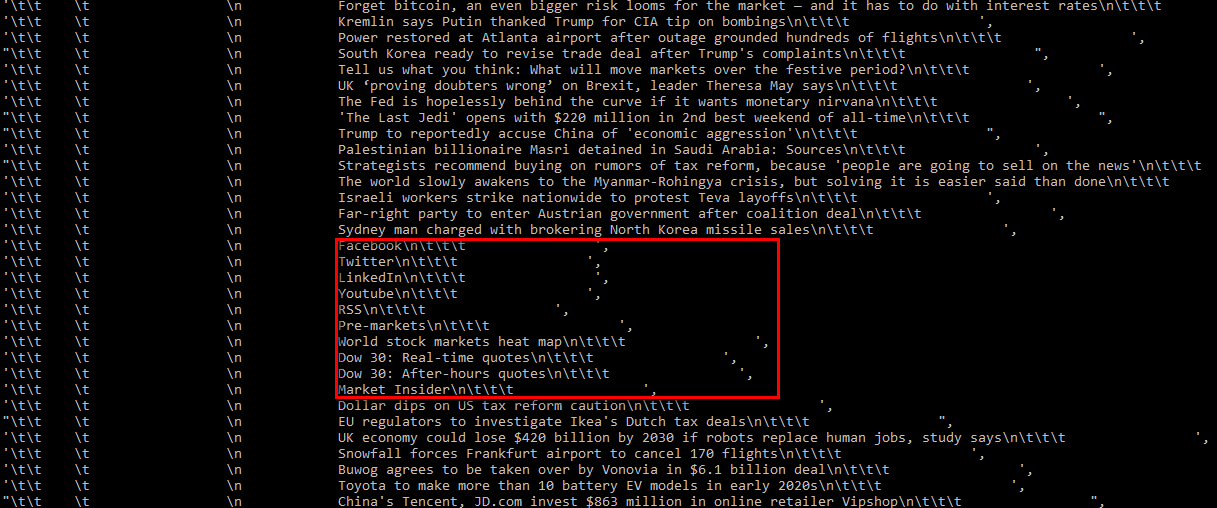
$cd exercise

$scrapy genspider news\_scraping <https://www.cnbc.com/world/?region=world> *#Generate a spider named news\_scraping.*

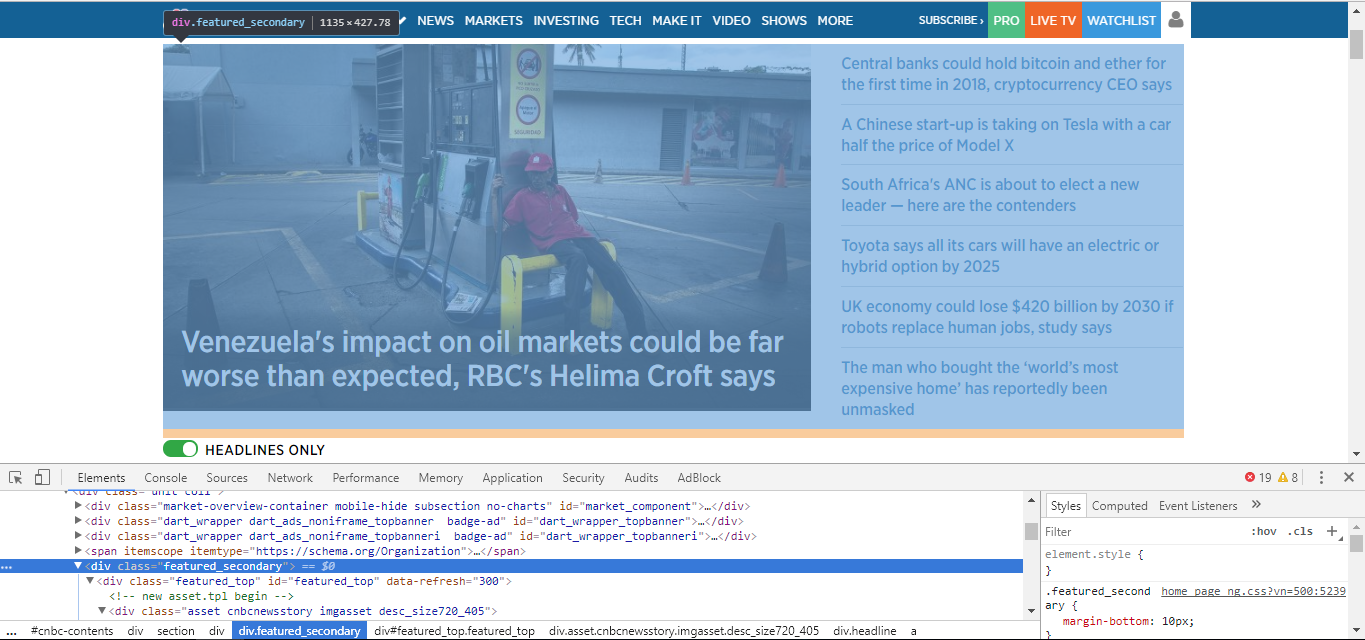


class=”headline” is found check whether it is the class that provides all useful information to scrap

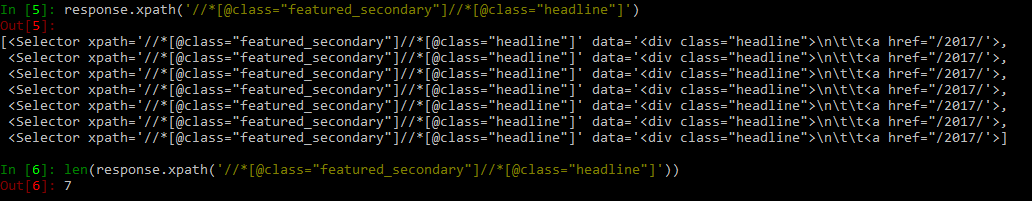
$response.xpath('//\*[@class="headline"]/a/text()').extract()



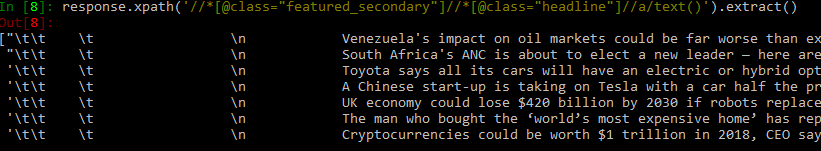
However, thing does not happen that way, therefore we have to find a container that contain the “box” of interest.Next, searching the box of interest with inspection.



We found that class="featured\_secondary" is the first box of interest (there are two boxes of interest for our concern)

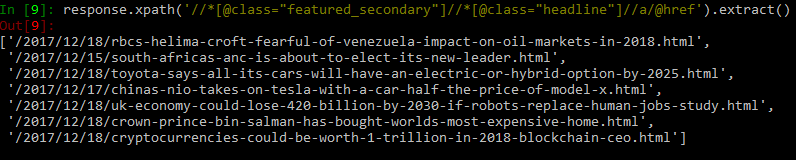


After that, we only obtain the information of class="headline" from class="featured\_secondary", and we check the number of the information whether it is tally with browser. It shows "7" which is correct. Other than that, a for loop will be implement since there is a list (7 elements) containing information.



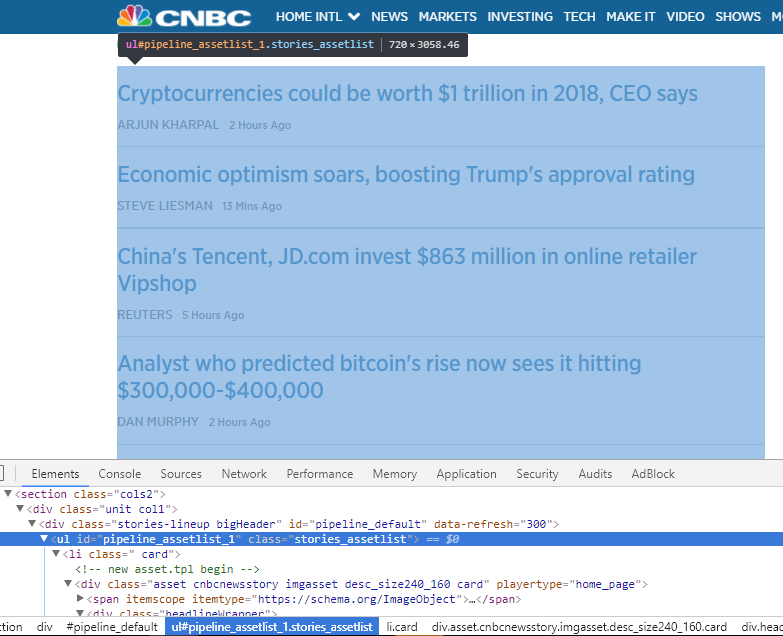
We have no problem extracting the text, there are a lot useless information e.g. “\t\n…”. To remove the them, a .strip() will be appended to the command.

After obtaining the tittle, we have to obtain the corresponding url.

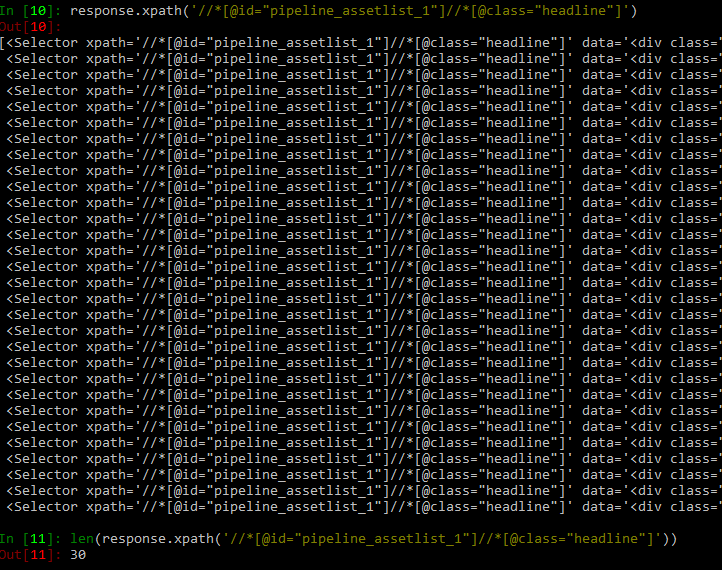


However, it is not a complete url, therefore we have to join them with the domain. Command response.urljoin() is used to tackle this problem.

There are two boxes of interest mentioned above, we going to do the same to the second box of interest.



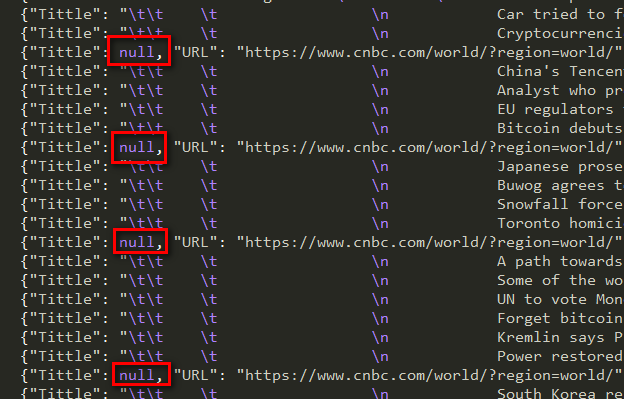
Therefore, id="pipeline\_assetlist\_1" is used to highlight the second box of interest and only scrap those class="headline" within the box of interest.



We write the same commands with the first box of interest.



However, some null tittles are recorded, and they will cause error as .strip() can not process null statement.



To tackle this problem, a for loop is implemented and it will strip only if the tittle is not null.



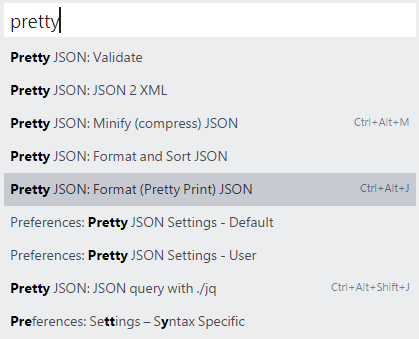
After describing the behaviour of the spider, it is ready to run and command below is used to execute the code in command prompt.

$scrapy crawl news\_scraping -o item.json

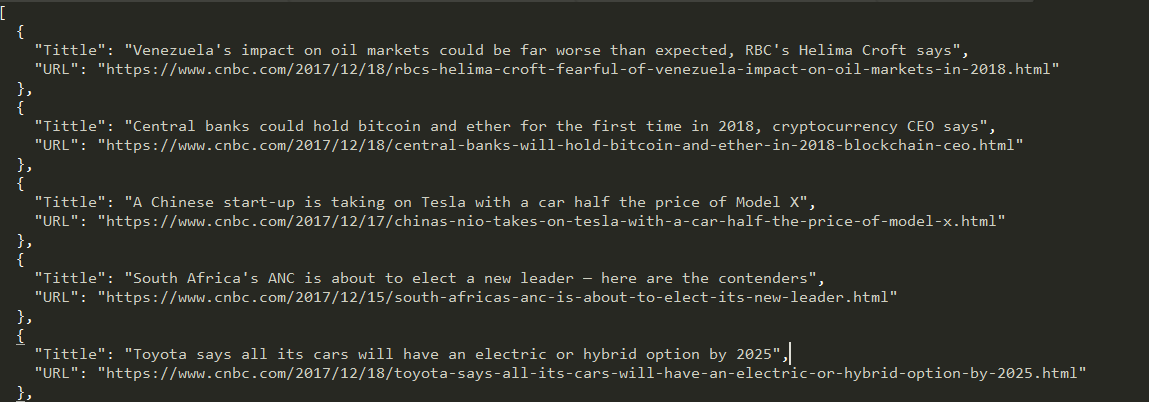
The pre-processed output is shown in figure below:



Pretty JSON with sublime text is used to make the json file more readable.

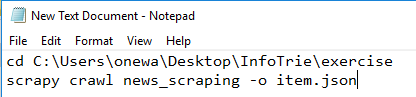


Final output after Pretty JSON:



# 2.1 Step 2 : Schedule Task Periodically

After describing the behaviour of the spider, we assign the computer to run the spider with Window Task Scheduler. Before using that, we have to declare the commands in batch file.



First line is to change the directory to the scrapy project in order to run with scrapy framework.

Second line is the command for strap and output to item.json.

Save the file with .bat extension, timer.bat is declared for this project.

Steps to schedule task in window system :Task Scheduler > Create Basic Task> Choose the .bat file > Schedule Timing (12pm every day, it can change according to its need).

# 3.0 Project Report

Figure below is the report after one execution, the execution only takes about one second.

