

Web scrapping Craigslist(Auto)

Radhey Shyam

CLnew york>staten island>for sale>cars+trucks

postaccount

cars & trucks - by owner

allownerdealer

☐ search titles only

☐ has image

☐ posted today

MILES FROM ZIP

miles

from zip

PRICE

min

max

MAKE AND MODEL

make / model

MODEL YEAR

min

max

ODOMETER

min

max

+ condition

+ cylinders

+ drive

+ fuel

+ paint color

+ size

+ title status

+ transmission

+ type

reset

update search

search cars & trucks - by owner


save search

gallery

<<< prev1 to 100 of 2410next >>


newest

\$5200




☆ May 23 2008 BMW 535xi For Sale @ Ace Auto World \$5200

\$1499




☆ May 23 2000 FORD TAURUS 3,0 FOR SALE \$1499 (STATEN ISLAND)

\$12500




☆ May 23 TOYOTA 2008 RAV4 - VERY LOW MILLAGES \$12500 (STATEN ISLAND)

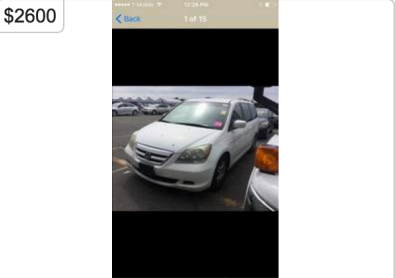
\$1999



\$5000



\$2600



javaforosx.dmg

[01]Intro_Python_Par...ipynb

MongoChef.dmg

Panda_HW_solution .ipynb

_solution.ipynb

Show All

mongodb_demo > demo > spiders > demo_spider.py

Project Files Problems demo_spider.py scrapy.cfg items.py settings.py pipelines.py craig_long.txt

mongodb_demo (~/.classnotes/mongodb/Demo) demo spiders init_.py items.py pipelines.py settings.py craig copy.txt craig.txt craig_long.txt craig_short.txt craig_short_xls.xlsx scrapy.cfg

price 6 matches

```
def parse(self, response):
    #ids = response.xpath('//*[class="row"]/@data-pid').extract()
    # title= response.xpath('// *[@ id = "titletextonly"]').extract()
    # price= response.xpath('//*[class = "price"]').extract()
    urls = []
    for i in range(26):
        tn = '?s=' + str(100*i)
        urls.append(response.urljoin(tn))
    for url in urls:
        #ids = response.xpath('//*[class="row"]/@data-pid').extract()
        yield Request(url, callback=self.parse_main_page)

def parse_main_page(self, response):
    ids = response.xpath('//*[class="row"]/@data-pid').extract()
    #time=xpath
    for id in ids:
        link = 'https://newyork.craigslist.org/stn/cto/' + str(id) + '.html'
        yield Request(link, callback=self.parse_detail_page)

def parse_detail_page(self, response):
    price = response.xpath('//*[class = "price"]/text()').extract()[0]
    title = response.xpath('//*[@ id = "titletextonly"]/text()').extract()[0]
    # body = response.xpath('//*[@ id = "postingbody"]').extract()
    post_time=response.xpath('//*[@id = "pagecontainer"]/section/section/div[2]/p[2]/time/text()').extract()[0]
    # update_time=response.xpath('//*[@id="pagecontainer"]/section/section/div[2]/p[3]/time').extract()
    body = response.xpath('//*[@id = "postingbody"]//text()').extract()
    body = reduce(lambda x,y: str(x).strip() + ' ' + str(y).strip(), body)
    #//*[@id="postingbody"]
    item = DemoItem()
    item['price'] = str(price)
    item['title'] = str(title)
    item['post_time']=str(post_time)
    item['body'] = body
    # item['update_time']=update_time

    yield item
```

Debug: demo demo demo

Debugger Console

```
price : $1500 ,
'title': "'02 SPRINTER'"
2016-05-22 01:27:39 [scrapy] DEBUG: Crawled (200) <GET https://newyork.craigslist.org/stn/cto/5570265157.html> (referer: https://newyork.craigslist.org/search/stn/cto?s=1700)
2016-05-22 01:27:40 [scrapy] DEBUG: Scraped from <200 https://newyork.craigslist.org/stn/cto/5570265157.html>
{'body': "Selling my 2011 black gt mustang . 30 thousand miles on it and car is kept in mint condition ...I need money for school and I'm sad that I have to let go of the ca
```

Error running demo: Coverage is not importable in this environment. Please install coverage.py to selected interpreter or enable 'Use bundled coverage' in Settings | Coverage (5/17/16, 4:09 PM) 18:36 LF UTF-8 1

Motivation

- Price distribution of the cars
- Peak day and time of posting the ads.
- Most common model years
- Most common words/models on craigslist

Conclusion

- Price range \$3500-\$4000 is the most common
- Friday is the most common day to put day, followed by Wednesday and Saturday
- Friday at 9-10 am , Saturday 8-10 am
- Frequency of word in title are Ford, Nissan, miles, civic etc.
- Year of the car (more than 10 years old)