异步加载AJAX：Asynchronous Javascript And XML，实现网页异步更新，不重新加载整个网页的情况下，对网页的某部分进行更新。

逆向工程：俗称“抓包”。通过分析动态加载的URL来获取异步加载的网页信息。

例：程序crawler\_test/crawer\_test12.py

import requests  
from lxml import etree  
import pymongo  
from mongodb\_test import mongo\_host  
from mongodb\_test import mongo\_port  
from crawler\_test1 import headers  
  
client = pymongo.MongoClient(mongo\_host, mongo\_port)  
mydb = client['mydb']  
timeline = mydb['timeline']  
  
  
def get\_time\_info(url, page):  
 user\_id = url.split('/')  
 user\_id = user\_id[4]  
 if url.find('page='):  
 page += 1  
 html = requests.get(url, headers=headers)  
 selector = etree.HTML(html.text)  
 infos = selector.xpath('//ul[@class="note-list"]/li')  
 for info in infos:  
 dd = info.xpath('div/div/div/span/@data-datetime')[0]  
 type = info.xpath('div/div/div/span/@data-type')[0]  
 timeline.insert\_one({'date': dd, 'type': type})  
  
 # 获取max\_id信息  
 id\_infos = selector.xpath('//ul[@class="note-list"]/li/@id')  
 if len(id\_infos) > 1:  
 feed\_id = id\_infos[-1]  
 max\_id = int(feed\_id.split('-')[1]) - 1  
 next\_url = 'http://www.jianshu.com/users/{0}/timeline?max\_id={1}&page={2}'.format(user\_id, max\_id, page)  
 get\_time\_info(next\_url, page) # 获取下一页信息  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 get\_time\_info('https://www.jianshu.com/users/9104ebf5e177/timeline', 1)

异步加载：

例：程序crawler\_test/crawer\_test13.py

import requests  
from lxml import etree  
import pymongo  
from mongodb\_test import mongo\_host  
from mongodb\_test import mongo\_port  
from crawler\_test1 import headers  
import re  
import json  
from multiprocessing import Pool  
  
client = pymongo.MongoClient(mongo\_host, mongo\_port)  
mydb = client['mydb']  
sevenday = mydb['sevenday']  
  
  
def get\_url(url):  
 html = requests.get(url, headers=headers)  
 selector = etree.HTML(html.text)  
 infos = selector.xpath('//ul[@class="note-list"]/li')  
 for info in infos:  
 article\_url\_part = info.xpath('div/a/@href')[0]  
 get\_info(article\_url\_part)  
  
  
def get\_info(url):  
 url\_header = 'https://www.jianshu.com/'  
 article\_url = url\_header + url  
 html = requests.get(article\_url, headers=headers)  
 selector = etree.HTML(html.text)  
 author = selector.xpath('//span[@class="name"]/a/text()')[0]  
 article = selector.xpath('//h1[@class="title"]/text()')[0]  
 date = selector.xpath('//span[@class="publish-time"]/text()')[0]  
 word = selector.xpath('//span[@class="wordage"]/text()')[0]  
 view = re.findall('"views\_count":(.\*?),', html.text, re.S)[0]  
 comment = re.findall('"comments\_count":(.\*?),', html.text, re.S)[0]  
 like = re.findall('"likes\_count":(.\*?),', html.text, re.S)[0]  
 id1 = re.findall('{"id":(.\*?),', html.text, re.S)[0]  
 gain\_url = url\_header + "notes/{0}/rewards?count=20".format(id1)  
 wb\_data = requests.get(gain\_url, headers=headers)  
 json\_data = json.loads(wb\_data.text)  
 gain = json\_data['rewards\_count'] # 获取打赏数据  
  
 include\_list = [] # 专题信息  
 include\_urls = [(url\_header + "notes/{0}/included\_collections?page={1}").format(id1, str(i)) for i in range(1, 10)]  
 for include\_url in include\_urls:  
 html = requests.get(include\_url, headers=headers)  
 json\_data = json.loads(html.text)  
 includes = json\_data['collections']  
 if len(includes) == 0:  
 pass  
 else:  
 for include in includes:  
 include\_title = include['title']  
 include\_list.append(include\_title)  
  
 info = {  
 'author': author,  
 'article': article,  
 'date': date,  
 'word': word,  
 'view': view,  
 'comment': comment,  
 'like': like,  
 'gain': gain,  
 'include': include\_list  
 }  
 sevenday.insert\_one(info)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 urls = ['https://www.jianshu.com/trending/weekly?page={0}'.format(str(i)) for i in range(1, 6)]  
 pool = Pool(processes=4)  
 for url in urls:  
 pool.apply\_async(get\_url, args=(url,))  
 pool.close()  
 pool.join()