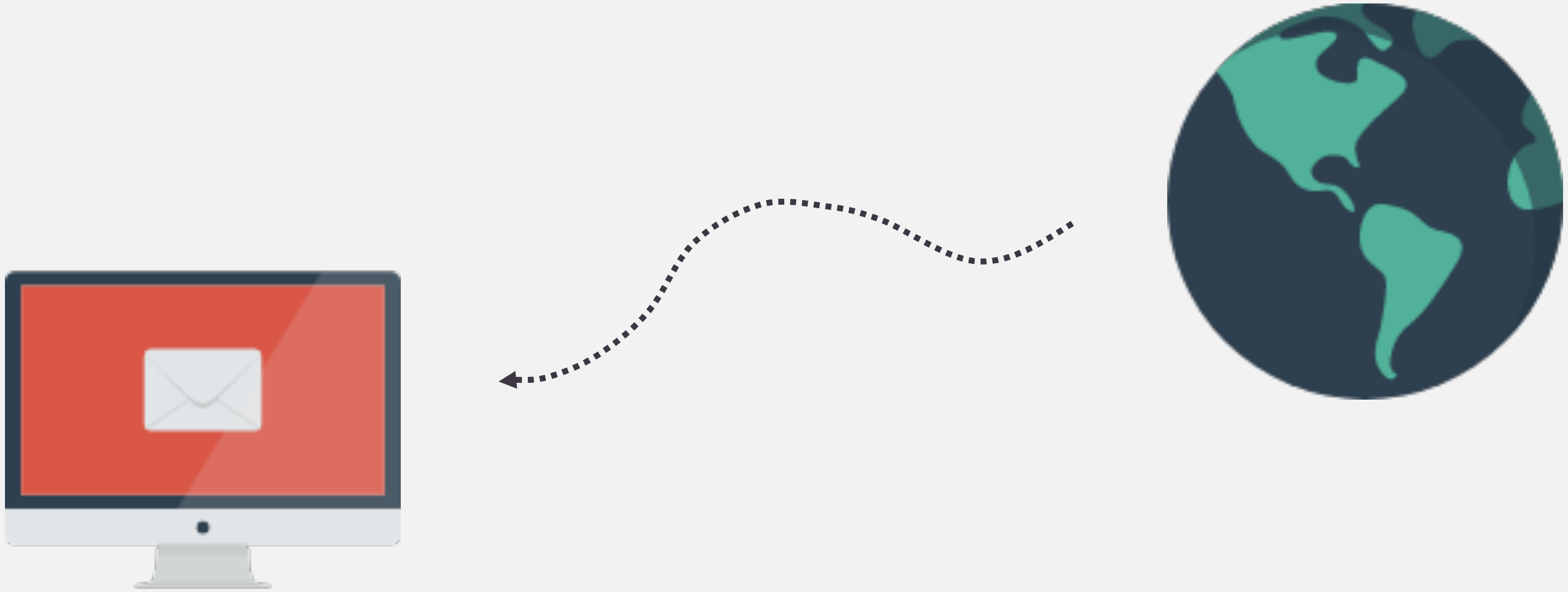
网络爬虫

数据科学引论（Python之道）

Powered by 陈昊鹏

爬虫是什么

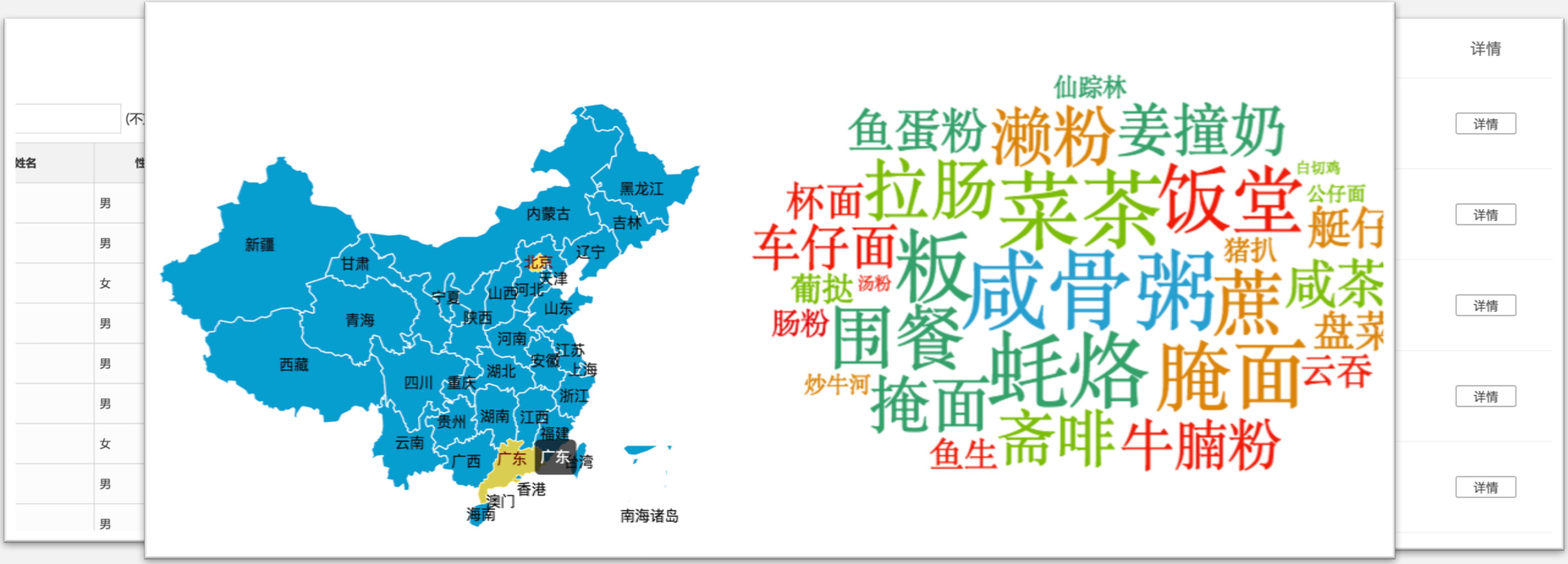
爬虫crawler，即网络爬虫Spider。是去**自动化**获取网络上的内容，是一个能够自动化地访问互联网并将网站内容下载下来的的程序或脚本。



Powered by 陈昊鹏

为什么需要爬虫？

高效自动化地从网络获取收集数据，后续可做数据处理。



(不

姓名	性
	男
	男
	女
	男
	男
	男
	女
	男
	男

详情

详情

详情

详情

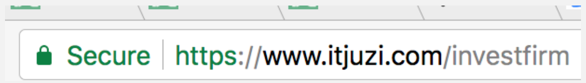
详情

详情

详情

Powered by 陈昊鹏

基本流程



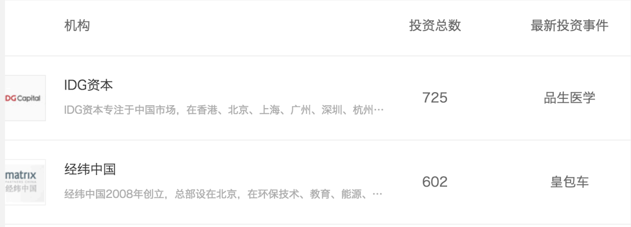
Secure | https://www.itjuzi.com/investfirm

URL

地址

统一资源定位符(网址)

根据url指定需要爬取的网页



	机构	投资总数	最新投资事件
IDG Capital	IDG资本 IDG资本专注于中国市场，在香港、北京、上海、广州、深圳、杭州…	725	品生医学
matrix 经纬中国	经纬中国 经纬中国2008年创立，总部设在北京，在环保技术、教育、能源、…	602	皇包车

HTML

网页原内容

网页内容编码

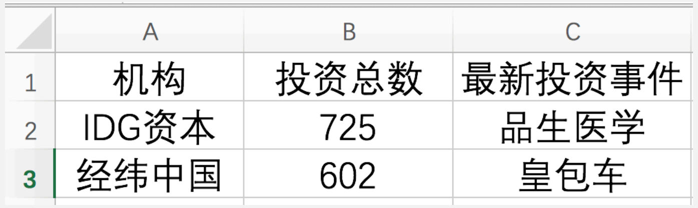
我们要从中提取我们需要的关键信息

TOOL

工具

爬虫工具

访问网页并从html解析信息的具体工具，如Scrapy



	A	B	C
1	机构	投资总数	最新投资事件
2	IDG资本	725	品生医学
3	经纬中国	602	皇包车

FORMAT

格式

统一化格式

得到信息将数据整理成统一格式导出

Powered by 陈昊鹏

Scrapy – URL指定

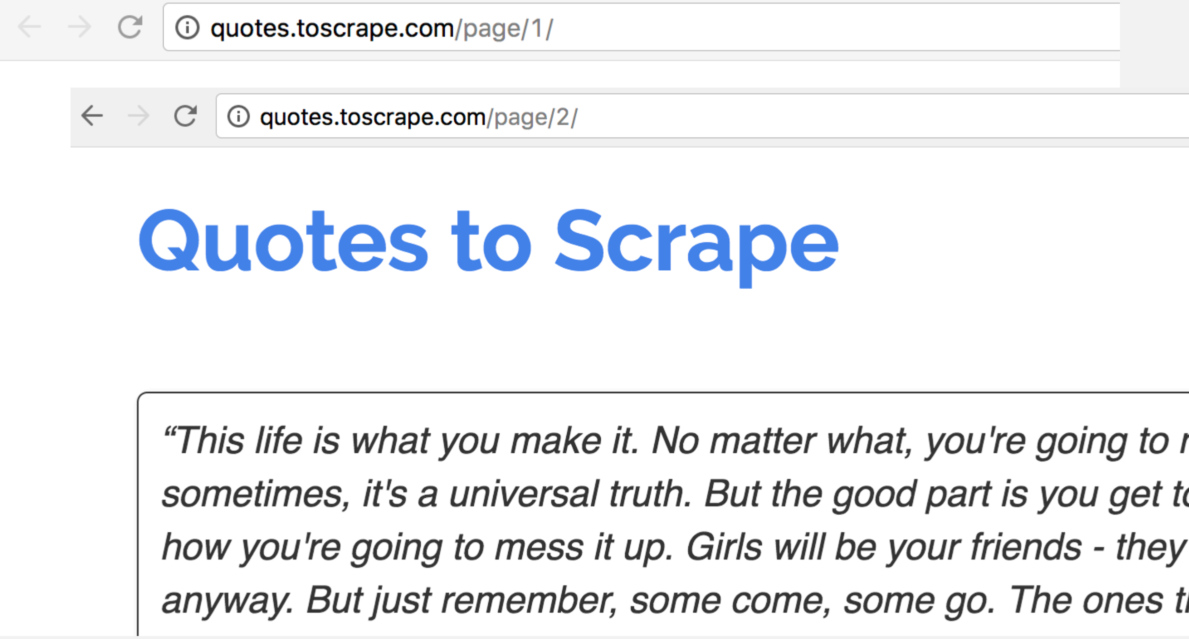
```
    def __init__(self):

        self.file = open('demo1_quotes.json', 'w');

        #设置待爬取网站列表
        self.urls = []
        for i in range(1,3):
            self.urls.append('http://quotes.toscrape.com/page/' + str(i) )

#       初始化效果 效果等同
#         self.urls = [
#             'http://quotes.toscrape.com/page/1/',
#             'http://quotes.toscrape.com/page/2/',
#         ]

        print(self.urls)
```



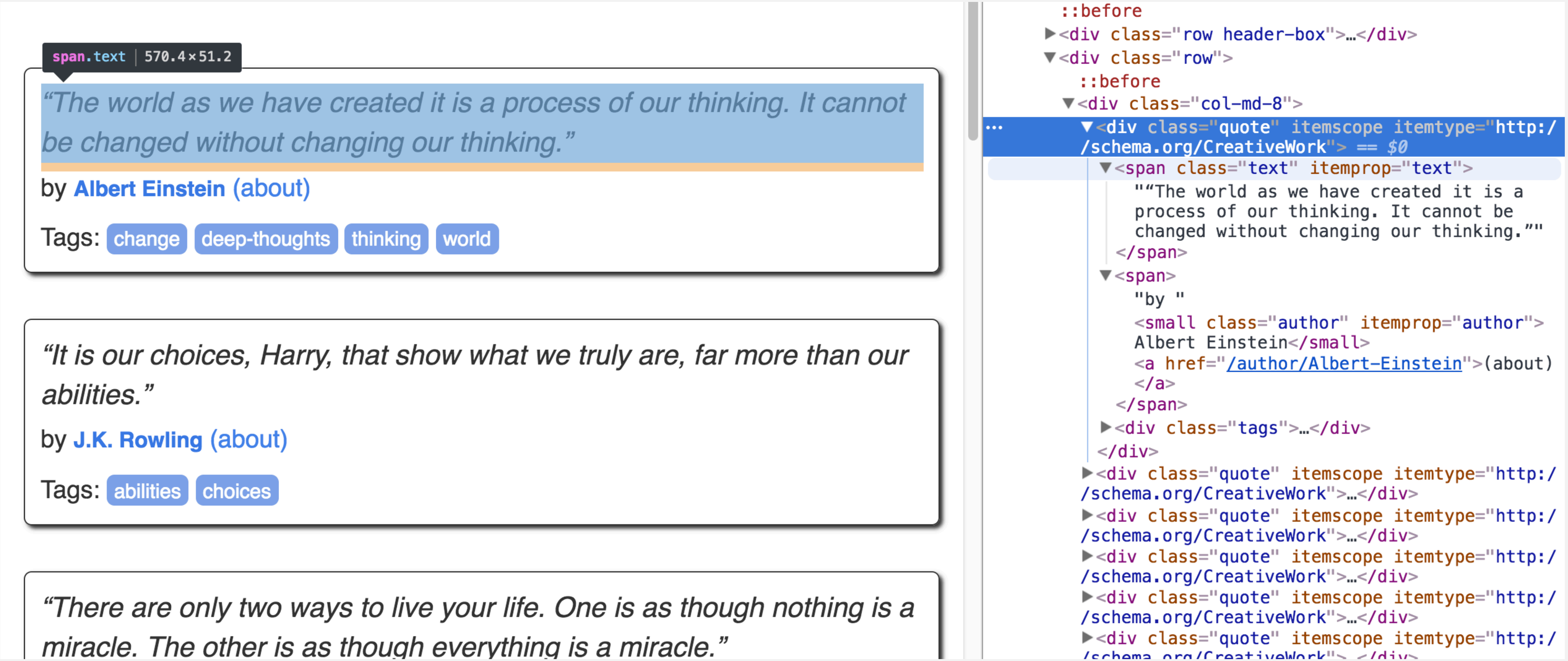
quotes.toscrape.com/page/1/

quotes.toscrape.com/page/2/

Quotes to Scrape

“This life is what you make it. No matter what, you're going to r
sometimes, it's a universal truth. But the good part is you get to
how you're going to mess it up. Girls will be your friends - they
anyway. But just remember, some come, some go. The ones t

Scrapy – HTML分析



span.text | 570.4×51.2

“The world as we have created it is a process of our thinking. It cannot be changed without changing our thinking.”

by **Albert Einstein** (about)

Tags: change deep-thoughts thinking world

“It is our choices, Harry, that show what we truly are, far more than our abilities.”

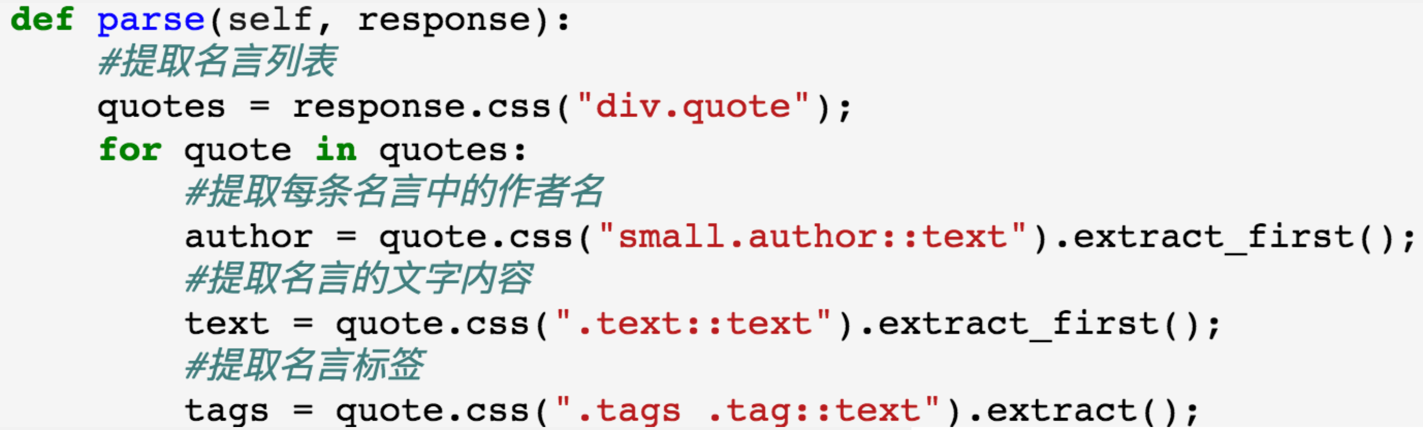
by **J.K. Rowling** (about)

Tags: abilities choices

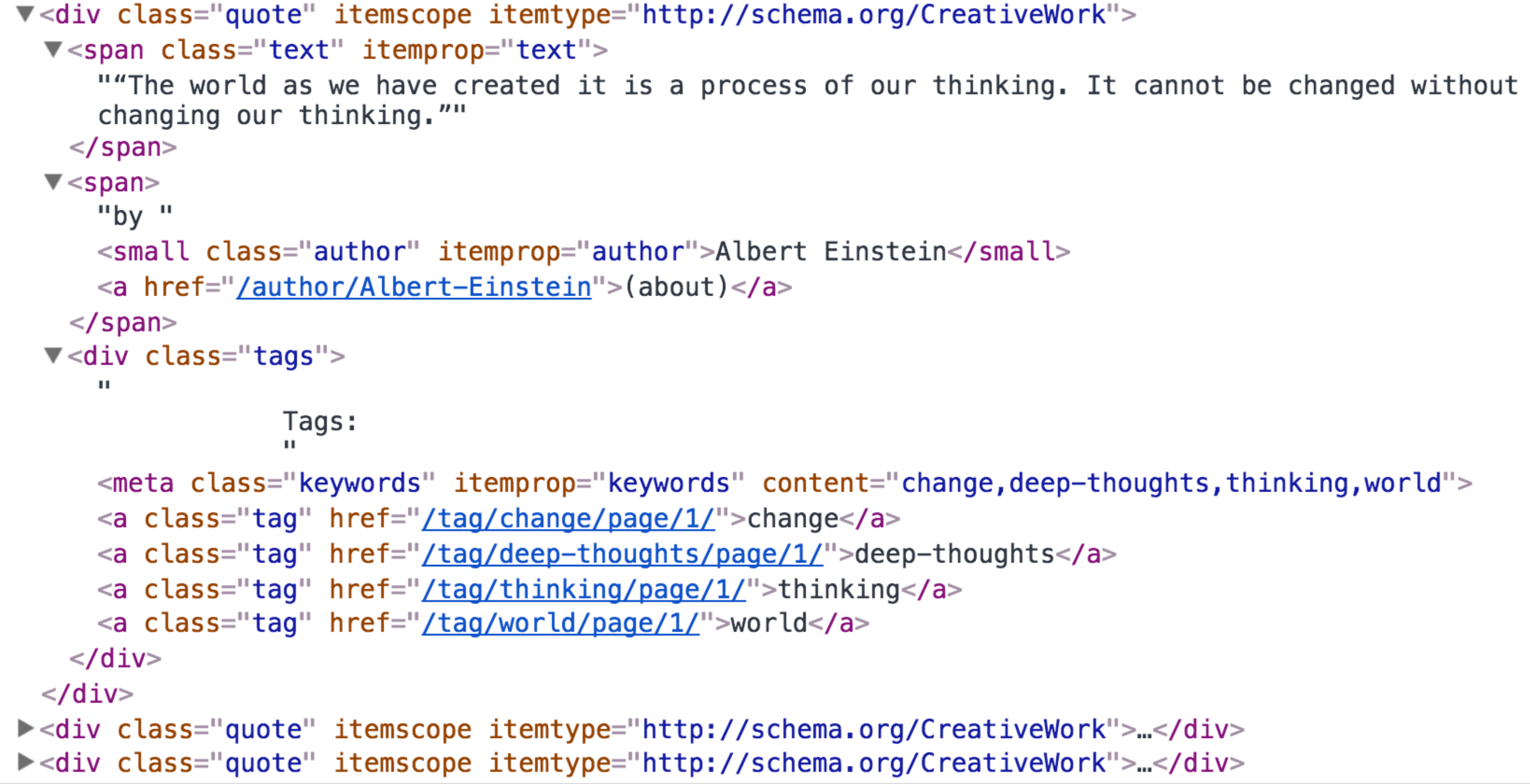
“There are only two ways to live your life. One is as though nothing is a miracle. The other is as though everything is a miracle.”

```
    ::before
  ▶<div class="row header-box">…</div>
  ▼<div class="row">
     ::before
    ▼<div class="col-md-8">
…    ▼<div class="quote" itemscope itemtype="http:/
     /schema.org/CreativeWork"> == $0
       ▼<span class="text" itemprop="text">
          "“The world as we have created it is a
          process of our thinking. It cannot be
          changed without changing our thinking.”"
        </span>
       ▼<span>
          "by "
          <small class="author" itemprop="author">
          Albert Einstein</small>
          <a href="/author/Albert-Einstein">(about)
          </a>
        </span>
       ▶<div class="tags">…</div>
      </div>
     ▶<div class="quote" itemscope itemtype="http:/
     /schema.org/CreativeWork">…</div>
     ▶<div class="quote" itemscope itemtype="http:/
     /schema.org/CreativeWork">…</div>
     ▶<div class="quote" itemscope itemtype="http:/
     /schema.org/CreativeWork">…</div>
     ▶<div class="quote" itemscope itemtype="http:/
     /schema.org/CreativeWork">…</div>
     ▶<div class="quote" itemscope itemtype="http:/
     /schema.org/CreativeWork">…</div>
```

Scrapy – 解析网页



```
def parse(self, response):
    #提取名言列表
    quotes = response.css("div.quote");
    for quote in quotes:
        #提取每条名言中的作者名
        author = quote.css("small.author::text").extract_first();
        #提取名言的文字内容
        text = quote.css(".text::text").extract_first();
        #提取名言标签
        tags = quote.css(".tags .tag::text").extract();
```

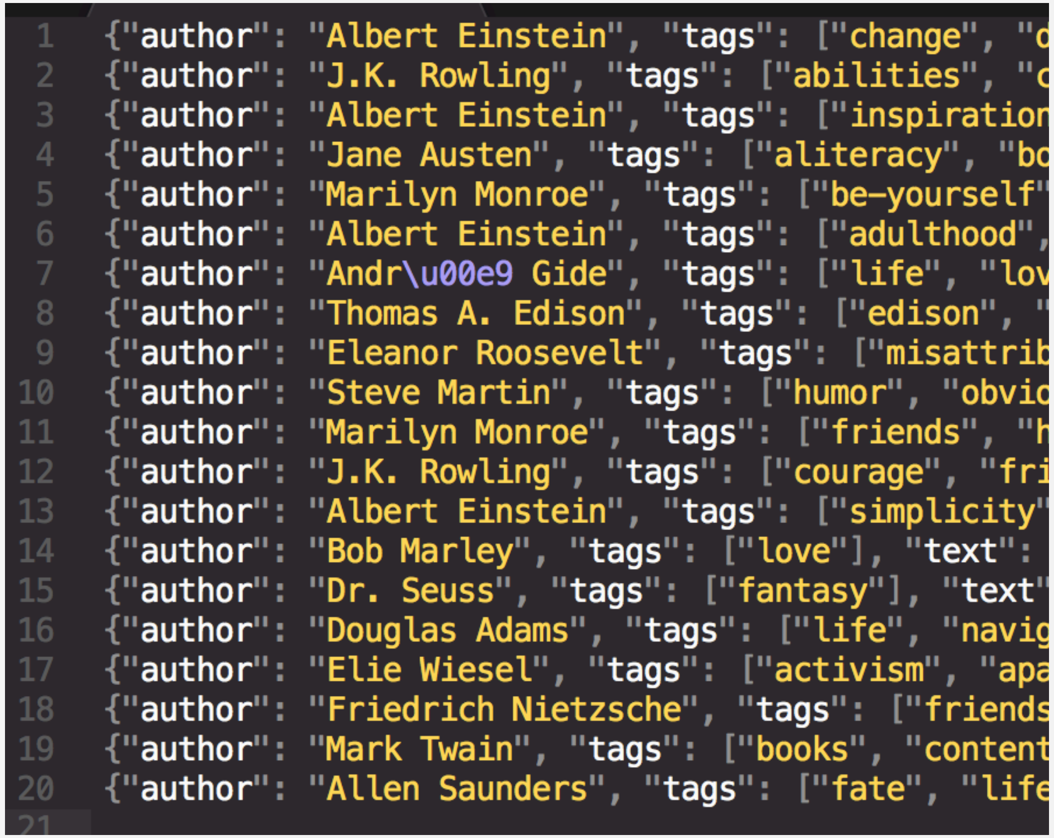


```
▼<div class="quote" itemscope itemtype="http://schema.org/CreativeWork">
  ▼<span class="text" itemprop="text">
     "“The world as we have created it is a process of our thinking. It cannot be changed without
     changing our thinking.”"
   </span>
  ▼<span>
     "by "
     <small class="author" itemprop="author">Albert Einstein</small>
     <a href="/author/Albert-Einstein">(about)</a>
   </span>
  ▼<div class="tags">
     "
                 Tags:
                 "
     <meta class="keywords" itemprop="keywords" content="change,deep-thoughts,thinking,world">
     <a class="tag" href="/tag/change/page/1/">change</a>
     <a class="tag" href="/tag/deep-thoughts/page/1/">deep-thoughts</a>
     <a class="tag" href="/tag/thinking/page/1/">thinking</a>
     <a class="tag" href="/tag/world/page/1/">world</a>
   </div>
 </div>
▶<div class="quote" itemscope itemtype="http://schema.org/CreativeWork">…</div>
▶<div class="quote" itemscope itemtype="http://schema.org/CreativeWork">…</div>
```

Scrapy – 格式化导出

```
#parse方法会在每个request收到response之后调用
def parse(self, response):
    #提取名言列表
    quotes = response.css("div.quote");
    for quote in quotes:
        #提取每条名言中的作者名
        author = quote.css("small.author::text").extract_first()
        #提取名言的文字内容
        text = quote.css(".text::text").extract_first();
        #提取名言标签
        tags = quote.css(".tags .tag::text").extract();
        #构建字典对象
        item = {"author":author, "text": text, "tags":tags };
        #将字典转换成json字符串
        line = json.dumps(dict(item))
        #将每个条目写入文件
        self.file.write(line + "\n")

    #及时将内容写入文件，否则可能会出现少许延迟
    self.file.flush()
    os.fsync(self.file)
    #输出当前解析完成的网页网址，可以当做爬取进度来看待,与程序逻辑无关
    print("over: " + response.url)
```



```
{"author": "Albert Einstein", "tags": ["change", "d
{"author": "J.K. Rowling", "tags": ["abilities", "c
{"author": "Albert Einstein", "tags": ["inspiration
{"author": "Jane Austen", "tags": ["aliteracy", "bo
{"author": "Marilyn Monroe", "tags": ["be-yourself"
{"author": "Albert Einstein", "tags": ["adulthood",
{"author": "Andr\u00e9 Gide", "tags": ["life", "lov
{"author": "Thomas A. Edison", "tags": ["edison", "
{"author": "Eleanor Roosevelt", "tags": ["misattrib
{"author": "Steve Martin", "tags": ["humor", "obvio
{"author": "Marilyn Monroe", "tags": ["friends", "h
{"author": "J.K. Rowling", "tags": ["courage", "fri
{"author": "Albert Einstein", "tags": ["simplicity"
{"author": "Bob Marley", "tags": ["love"], "text":
{"author": "Dr. Seuss", "tags": ["fantasy"], "text"
{"author": "Douglas Adams", "tags": ["life", "navig
{"author": "Elie Wiesel", "tags": ["activism", "apa
{"author": "Friedrich Nietzsche", "tags": ["friends
{"author": "Mark Twain", "tags": ["books", "content
{"author": "Allen Saunders", "tags": ["fate", "life
```

常见问题

IP被封杀

重构

网页更新

Powered by 陈昊鹏

Thanks for Watching

Powered by 陈昊鹏