**爬虫0**

到底什么是爬虫呢？

爬虫，从本质上来说，就是利用程序在网上拿到对我们有价值的数据。

爬虫能做很多事，能做商业分析，也能做生活助手，比如：分析北京近两年二手房成交均价是多少？深圳的Python工程师平均薪资是多少？北京哪家餐厅粤菜最好吃？等等。

这是个人利用爬虫所做到的事情，而公司，同样可以利用爬虫来实现巨大的商业价值。比如你所熟悉的搜索引擎——百度和谷歌，它们的核心技术之一也是爬虫，而且是超级爬虫。

以百度为例，你在搜索的时候仔细看，会发现每个搜索结果下面都有一个百度快照。



点击百度快照，你会发现网址的开头有baidu这个词，也就是说这个网页属于百度



这是因为，百度这家公司会源源不断地把千千万万个网站爬取下来，存储在自己的服务器上。你在百度搜索的本质就是在它的服务器上搜索信息，你搜索到的结果是一些超链接，在超链接跳转之后你就可以访问其它网站了。

爬虫还让这些搜索巨头有机会朝着人工智能的未来迈进，因为人工智能的发展离不开海量的数据。而每天使用这些搜索网站的用户都是数以亿计的，产生的数据自然也是难以计量的。

从搜索巨头到人工智能巨头，这是一条波澜壮阔的路。而我们应该看到，事情的源头，却是我们今日所书写的“爬虫”。

现在，我们对爬虫有了初步的印象，知道了爬虫能做什么，那我们接下来来看看，爬虫是如何做到这些事的。

明晰路径

一般来说，传统的拿数据的做法是：通过浏览器上网，手动下载所需要的数据。其实在这背后，浏览器做了很多我们看不见的工作，而只有了解浏览器的工作原理后，才能真正理解爬虫在帮我们做什么。

所以，我们先来看看浏览器的工作原理。

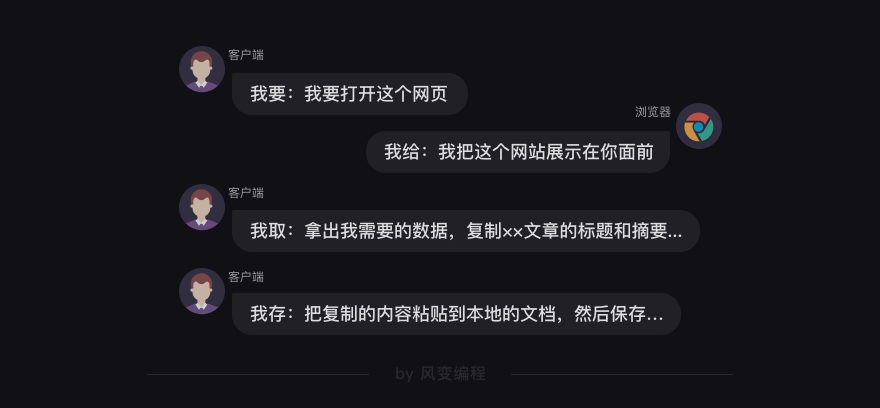
浏览器的工作原理

我们会通过一个网站：人人都是蜘蛛侠 来学习浏览器的工作原理。（这是我们自己搭建的网站，方便教学和实操）



在这个网站上有几篇文章。假设，我们想收藏首页的文章标题和文章摘要，我们可能会复制粘贴到本地文档。

其实这个过程，是一个人和浏览器在交流的过程，请看下图：



实不相瞒，在这个过程中，浏览器的交流对象不只有你，还有【服务器】。我们可以把服务器理解为一个超级电脑，它可以计算和存储大量数据，并且在互联网中互相传输数据。

更完整的交流过程是下图这样的：



首先，我们在浏览器输入网址（也可以叫URL）。然后，浏览器向服务器传达了我们想访问某个网页的需求，这个过程就叫做【请求】。

紧接着，服务器把你想要的网站数据发送给浏览器，这个过程叫做【响应】。

所以浏览器和服务器之间，先请求，后响应，有这么一层关系。



当服务器把数据响应给浏览器之后，浏览器并不会直接把数据丢给你。因为这些数据是用计算机的语言写的，浏览器还要把这些数据翻译成你能看得懂的样子，这是浏览器做的另一项工作【解析数据】。

紧接着，我们就可以在拿到的数据中，挑选出对我们有用的数据，这是【提取数据】。

最后，我们把这些有用的数据保存好，这是【存储数据】。

以上，就是浏览器的工作原理，是人、浏览器、服务器三者之间的交流过程。

可这和爬虫有什么关系呢？

爬虫的工作原理

其实，爬虫可以帮我们代劳这个过程的其中几步，请看下图：



当你决定去某个网页后，首先，爬虫可以模拟浏览器去向服务器发出请求；其次，等服务器响应后，爬虫程序还可以代替浏览器帮我们解析数据；接着，爬虫可以根据我们设定的规则批量提取相关数据，而不需要我们去手动提取；最后，爬虫可以批量地把数据存储到本地。

这就是爬虫做的事。简化上图，就是爬虫的工作原理了：



其实，还可以把最开始的【请求——响应】封装为一个步骤——获取数据。由此，我们得出，爬虫的工作分为四步：



第0步：获取数据。爬虫程序会根据我们提供的网址，向服务器发起请求，然后返回数据。

第1步：解析数据。爬虫程序会把服务器返回的数据解析成我们能读懂的格式。

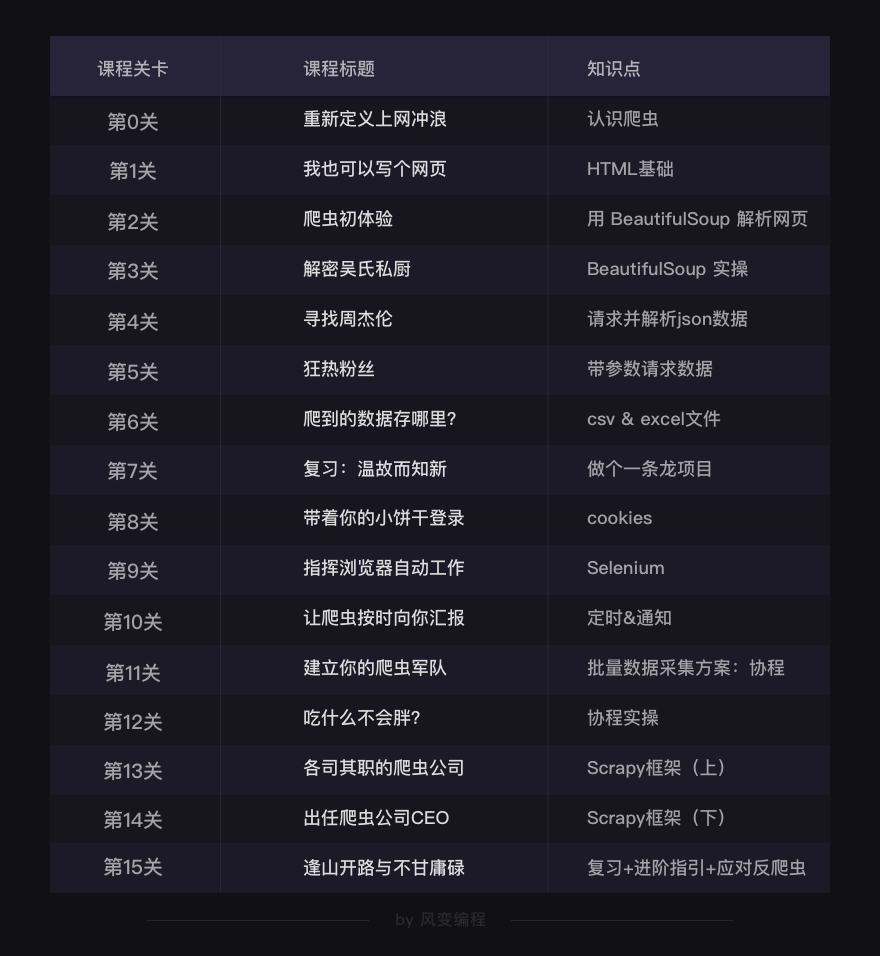
第2步：提取数据。爬虫程序再从中提取出我们需要的数据。

第3步：储存数据。爬虫程序把这些有用的数据保存起来，便于你日后的使用和分析。

这就是爬虫的工作原理啦，无论之后的学习内容怎样变化，其核心都是爬虫原理。

下面，我们快速浏览一下整个关卡的学习大纲。大纲展示了我们将如何逐步学习和掌握爬虫这项技能，了解即可。

爬虫学习大纲



第0关，初识爬虫，你会了解爬虫的工作原理，写出一个简单的爬虫程序，学会爬虫的第0步：获取数据。

第1关，你将会快速入门HTML基础知识，达到读懂和修改HTML文档的水平。有了这些基础，你才能去学习如何解析数据和提取数据。

第2、3、4、5关，我会教你爬虫的第1和第2步：解析数据和提取数据。同时，你还会学到两种不同的发起请求的方式。

在第6关，你将学会存储数据，即把目标数据写入到本地的Excel表格中。到此，你就学会了爬虫完整的四个步骤，掌握了最基本的爬虫技能啦。

第7关，我们一起做一个项目，爬取一个知乎大v的所有文章，并且存到Excel中。以此，我们巩固和复习了0-6关的所有知识。第7关会是一个分水岭，后面关卡的进阶知识都建立在前7关的基础上。

第8关，学会cookies，就可以让浏览器记住你，你们可以更方便地长期保持联系，而不是在一次见面之后就相忘于江湖。

第9关，学习控制浏览器，来应对爬虫中一些更复杂的情况。第10关，你的爬虫会变得更自动化，爬虫程序不但可以定时工作，还可以把爬取结果传递给你。

接下来的4关，你将学会更高效更强大的爬虫方法，让爬虫技能升级。

第15关毕业总结，就到了告别的时刻了。这时你也学成出师，可以用爬虫知识去做自己想做的事情了，让爬虫为你消灭重复劳动，高效获取信息，创造出更多价值。

以上，就是爬虫学习大纲。

好，从宏观上理解爬虫后，接下来，我们将近距离地体验一下爬虫，也就是——写代码。

体验爬虫

这一部分的任务就是学会爬虫的第0步：获取数据。

我们将会利用一个强大的库——requests来获取数据。

在学习系统里，已经帮你预装好requests库。如果你希望在电脑上安装，方法是：在Mac电脑里打开终端软件（terminal），输入pip3 install requests，然后点击enter即可；Windows电脑里叫命令提示符（cmd），输入pip install requests 即可。

requests库可以帮我们下载网页源代码、文本、图片，甚至是音频。其实，“下载”本质上是向服务器发送请求并得到响应。

先来看requests.get()方法。

requests.get()

requests.get()的具体用法如下，请仔细阅读注释噢：

import requests

#引入requests库

res = requests.get('URL')

#requests.get是在调用requests库中的get()方法，它向服务器发送了一个请求，括号里的参数是你需要的数据所在的网址，然后服务器对请求作出了响应。

#我们把这个响应返回的结果赋值在变量res上。

第1行代码，是引用requests库。

第3行代码，requests.get()发送了请求，然后得到了服务器的响应。服务器返回的结果是个Response对象，现在存储到了我们定义的变量res中。

如果用图片展示，那就是这样的：



现在，我们试着用requests.get()来下载一个小说——《三国演义》：



小说的URL（网址）是：[https://localprod.pandateacher.com/python-manuscript/crawler-html/sanguo.md](https://localprod.pandateacher.com/python-manuscript/crawler-html/sanguo.md" \t "_blank)

那么根据所学，代码应该是这样的：

import requests

#引入requests库

res = requests.get('https://localprod.pandateacher.com/python-manuscript/crawler-html/sanguo.md')

#发送请求，并把响应结果赋值在变量res上

Response对象的常用属性

Python是一门面向对象编程的语言，而在爬虫中，理解数据是什么对象是非常、特别、以及极其重要的一件事。因为只有知道了数据是什么对象，我们才知道对象有什么属性和方法可供我们操作。

所以，我们现在来打印看看刚刚用requests.get()获取到的数据是什么，请点击运行下面的程序：

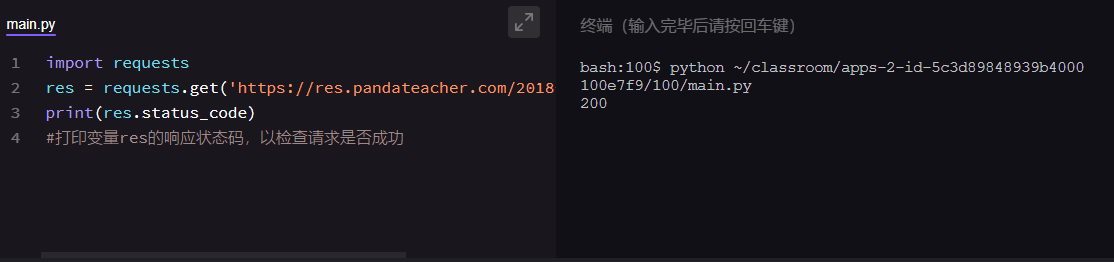
终端显示：

这代表着：res是一个对象，属于requests.models.Response类。好，既然已经知道res是一个Response对象了，我们也就可以去了解它的相应属性和方法了。

我们主要讲Response对象常用的四个属性：



首先是response.status\_code，直接看用法。在看完代码后，请点击运行。



第3行代码是在打印res的响应状态码，它可以用来检查我们的requests请求是否得到了成功的响应。我们看到终端结果显示了200，这个数字代表服务器同意了请求，并返回了数据给我们。

除了200，我们还可能收到其他的状态码。下面有一个表格，供你参考不同的状态码代表什么，但不需要记住它们，在遇到问题的时候查询就好。



response.status\_code是一个很常用的属性，在我们之后的爬虫代码中也将多次出现。



接着的属性是response.content，它能把Response对象的内容以二进制数据的形式返回，适用于图片、音频、视频的下载，看个例子你就懂了。

假如我们想下载这张图片，它的URL是：[https://res.pandateacher.com/2018-12-18-10-43-07.png](https://res.pandateacher.com/2018-12-18-10-43-07.png" \t "_blank)



那么代码可以这样写：

import requests

res = requests.get('https://res.pandateacher.com/2018-12-18-10-43-07.png')

#发出请求，并把返回的结果放在变量res中

pic=res.content

#把Reponse对象的内容以二进制数据的形式返回

photo = open('ppt.jpg','wb')

#新建了一个文件ppt.jpg，这里的文件没加路径，它会被保存在程序运行的当前目录下。

#图片内容需要以二进制wb读写。你在学习open()函数时接触过它。

photo.write(pic)

#获取pic的二进制内容

photo.close()

#关闭文件

这样，我们的图片就下载成功啦~你也可以在本地运行这个程序。

讲完了response.content，继续看response.text，这个属性可以把Response对象的内容以字符串的形式返回，适用于文字、网页源代码的下载。

举个例子，我们还是来下载小说《三国演义》的第一章。

代码如下：

import requests

#引用requests库

res = requests.get('https://localprod.pandateacher.com/python-manuscript/crawler-html/sanguo.md')

#下载《三国演义》第一回，我们得到一个对象，它被命名为res

novel=res.text

#把Response对象的内容以字符串的形式返回

print(novel[:800])

#现在，可以打印小说了，但考虑到整章太长，只输出800字看看就好。在关于列表的知识那里，你学过[:800]的用法。

之后，我们就可以用通过读写文件把小说保存到本地了。这是Python基础语法知识，你应该已经学会了。下面直接给出做法，你也可以在自己的本地电脑上做尝试练习。

import requests

#引用requests库

res = requests.get('https://localprod.pandateacher.com/python-manuscript/crawler-html/sanguo.md')

#下载《三国演义》第一回，我们得到一个对象，它被命名为res

novel=res.text

#把Response对象的内容以字符串的形式返回

k = open('《三国演义》.txt','a+')

#创建一个名为《三国演义》的txt文档，指针放在文件末尾，追加内容

k.write(novel)

#写进文件中

k.close()

#关闭文档

接下来，我们看最后一个属性：response.encoding，它能帮我们定义Response对象的编码。

我们仍然以三国演义的小说来做示范，请看下面的代码（第5行为新增代码），然后点击运行：

import requests

#引用requests库

res = requests.get('https://localprod.pandateacher.com/python-manuscript/crawler-html/sanguo.md')

#下载《三国演义》第一回，我们得到一个对象，它被命名为res

res.encoding='gbk'

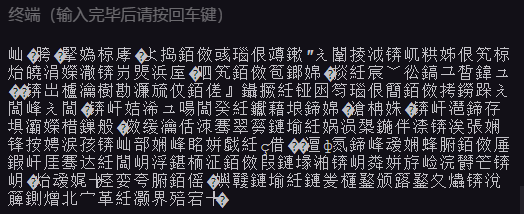
#定义Response对象的编码为gbk

novel=res.text

#把Response对象的内容以字符串的形式返回

print(novel[:800])

#打印小说的前800个字



诶，为什么会出现一段乱码呢？

事情是这样的：首先，目标数据本身有它的编码类型，这个《三国演义》URL中的数据类型是'utf-8'（因为这个网页是老师写的，所以我知道）。而用requests.get()发送请求后，我们得到一个Response对象，其中，requests模块会对数据的编码类型做出自己的判断。

但是，第5行的代码不管原来的判断是什么，直接定义了Response对象的编码类型是'gbk'。这样一来，跟数据本身的编码'utf-8'就不一致了，所以打印出来，就是一堆乱码。

import requests

#引用requests库

res = requests.get('https://localprod.pandateacher.com/python-manuscript/crawler-html/sanguo.md')

#下载《三国演义》第一回，我们得到一个对象，它被命名为res

res.encoding='utf-8'

#定义Reponse对象的编码为utf-8。

novel=res.text

#把Response对象的内容以字符串的形式返回

print(novel[:800])

#打印小说的前800个字。

这只是个示范，是为了让大家理解res.encoding的意义，也就是它能定义Response对象的编码类型。

那在真实的情况中，我们该在什么时候用res.encoding呢？

首先，目标数据本身是什么编码是未知的。用requests.get()发送请求后，我们会取得一个Response对象，其中，requests库会对数据的编码类型做出自己的判断。但是！这个判断有可能准确，也可能不准确。

如果它判断准确的话，我们打印出来的response.text的内容就是正常的、没有乱码的，那就用不到res.encoding；如果判断不准确，就会出现一堆乱码，那我们就可以去查看目标数据的编码，然后再用res.encoding把编码定义成和目标数据一致的类型即可。

总的来说，就是遇上文本的乱码问题，才考虑用res.encoding。

好，到这里，requests.get()方法和Response对象常见的四个属性就讲完了。

如果用一张图来总结，那就是这样的：



可以看到，爬虫的第0步：获取数据，本质就是通过URL去向服务器发出请求，服务器再把相关内容封装成一个Response对象返回给我们，这是通过requests.get()实现的，而我们获取到的Response对象下有四个常用的属性。

同学们要注意这种从URL到Response这种操作对象的转换关系。

好啦，你已经知道怎么用requests库来获取数据了，爬虫的第0步你就搞定了，此处应该有掌声。

说完了和爬虫相关的知识，在本关卡的最后，还想跟你聊聊和爬虫有关的道德准则。

爬虫伦理

就像是两个人在来来往往的相处中，会考虑对方的感受；在互联网的世界中，我们也要考虑一下服务器对爬虫的感受是怎样的。

我们说过，服务器其实就是一个超级电脑，拥有这个服务器的公司，对爬虫其实也有明确的态度。

通常情况下，服务器不太会在意小爬虫，但是，服务器会拒绝频率很高的大型爬虫和恶意爬虫，因为这会给服务器带来极大的压力或伤害。

不过，服务器在通常情况下，对搜索引擎是欢迎的态度（刚刚讲过，谷歌和百度的核心技术之一就是爬虫）。当然，这是有条件的，而这些条件会写在Robots协议。

Robots协议是互联网爬虫的一项公认的道德规范，它的全称是“网络爬虫排除标准”（Robots exclusion protocol），这个协议用来告诉爬虫，哪些页面是可以抓取的，哪些不可以。

如何查看网站的robots协议呢，很简单，在网站的域名后加上/robots.txt就可以了。

我们截取了一部分淘宝的robots协议 （ <http://www.taobao.com/robots.txt>）。在截取的部分，可以看到淘宝对百度和谷歌这两个爬虫的访问规定，以及对其它爬虫的规定。

User-agent: Baiduspider #百度爬虫

Allow: /article #允许访问 /article.htm

Allow: /oshtml #允许访问 /oshtml.htm

Allow: /ershou #允许访问 /ershou.htm

Allow: /$ #允许访问根目录，即淘宝主页

Disallow: /product/ #禁止访问/product/

Disallow: / #禁止访问除 Allow 规定页面之外的其他所有页面

​

User-Agent: Googlebot #谷歌爬虫

Allow: /article

Allow: /oshtml

Allow: /product #允许访问/product/

Allow: /spu

Allow: /dianpu

Allow: /oversea

Allow: /list

Allow: /ershou

Allow: /$

Disallow: / #禁止访问除 Allow 规定页面之外的其他所有页面

​

…… # 文件太长，省略了对其它爬虫的规定，想看全文的话，点击上面的链接

​

User-Agent: \* #其他爬虫

Disallow: / #禁止访问所有页面

协议里最常出现的英文是Allow和Disallow，Allow代表可以被访问，Disallow代表禁止被访问。而且有趣的是，淘宝限制了百度对产品页面的爬虫，却允许谷歌访问。

所以，当你在百度搜索“淘宝网”时，会看到下图的这两行小字。



因为百度很好地遵守了淘宝网的robots.txt协议，自然，你在百度中也查不到淘宝网的具体商品信息了。

虽然Robots协议只是一个道德规范，和爬虫相关的法律也还在建立和完善之中，但我们在爬取网络中的信息时，也应该有意识地去遵守这个协议。

网站的服务器被爬虫爬得多了，也会受到较大的压力，因此，各大网站也会做一些反爬虫的措施。不过呢，有反爬虫，也就有相应的反反爬虫，这些我们会在后面关卡中会详细说。

爬虫就像是核技术，人们可以利用它去做有用的事，也能利用它去搞破坏。

恶意消耗别人的服务器资源，是一件不道德的事，恶意爬取一些不被允许的数据，还可能会引起严重的法律后果。

工具在你手中，如何利用它是你的选择。当你在爬取网站数据的时候，别忘了先看看网站的Robots协议是否允许你去爬取。

同时，限制好爬虫的速度，对提供数据的服务器心存感谢，避免给它造成太大压力，维持良好的互联网秩序，也是我们该做的事。

以上，就是本关的学习内容了。