[Python爬虫入门：什么是爬虫？](#header-n9)  
 [爬虫特点概要](#header-n12)  
 [爬虫的概念](#header-n25)  
 [爬虫的作用](#header-n39)  
 [爬虫的分类](#header-n53)  
 [根据被爬网闸的数量不同，可以分为：](#header-n54)  
 [根据是否以获取数据为目的，可以分为：](#header-n60)  
 [根据URL地址和对应页面内容是否改变，数据增量爬虫可以分为：](#header-n66)  
 [爬虫流程](#header-n76)  
 [http以及https的概念和区别](#header-n85)  
 [爬虫特别注意的请求头](#header-n102)  
 [爬虫特别注意的响应头](#header-n126)  
 [常见的响应状态码](#header-n132)  
 [http请求的过程](#header-n155)  
 [注意](#header-n160)  
 [最后](#header-n171)

# Python爬虫入门：什么是爬虫？



看到上面的那只蜘蛛没？别误会，今天要教你如何玩上面的蜘蛛。我们正式从0到1轻松学会Python爬虫.......

## 爬虫特点概要

* 知识碎片化

爬虫方向的知识是十分碎片化的，因为我们写爬虫的时候会面对各种各样的网站，每个网站实现的技术都是相似的，但是大多数时候还是有差别的，这就要求我们对不同的网站使用不同的技术手段。爬虫并不像在学习web的时候要实现某一功能只要按照一定的套路就能做出来。

* 学习难度

爬虫的入门相对而言还是要比web简单，但是在后期，爬虫的难度要大于web。难点在于爬虫工程师与运维人员进行对抗，可能你写一个网站的爬虫，结果该网站的运维人员加了反爬的措施，那么作为爬虫工程师就要解决这个反爬。

* 学习特点

学习爬虫并不像学习web，学习web有一个完整的项目可以练手，因为爬虫的特点，也导致学习爬虫是以某网站为对象的，可以理解为一个技术点一个案例。

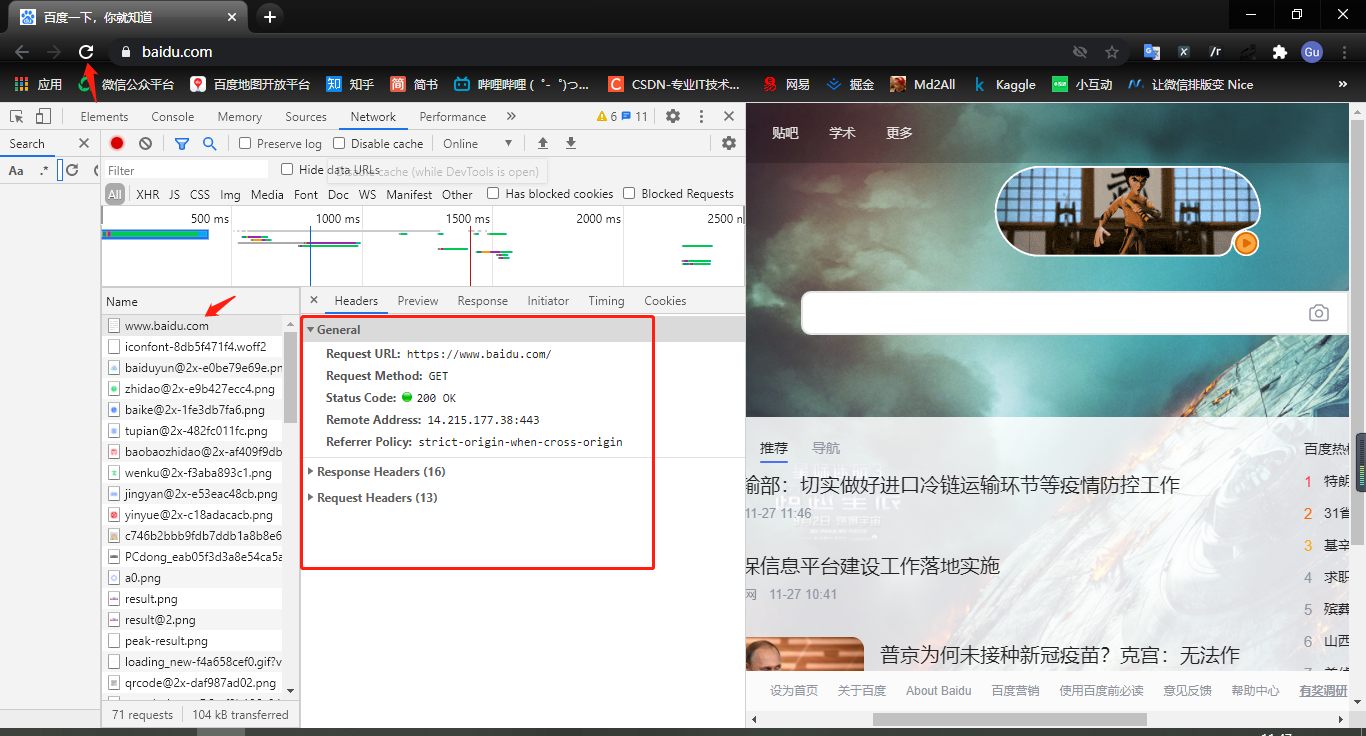
## 爬虫的概念

模拟浏览器，发送请求，获取响应

网络爬虫（又被称为网页蜘蛛、网页机器人）就是模拟客户端（主要是指浏览器）发送请求，接收请求响应，按照一定规则、自动抓取互联网信息的程序。

* 原则上，只要是浏览器能做的事情，爬虫都能做
* 爬虫也只能获取浏览器所展示出来的数据

在浏览器中输入百度网址，打开开发者工具，点击network，点击刷新，即可进行抓包。



**了解爬虫概念**

## 爬虫的作用

爬虫在互联网中的作用

* 数据采集
* 软件测试
* 12306抢票
* 网站投票
* 网络安全

## 爬虫的分类

### 根据被爬网闸的数量不同，可以分为：

* 通用爬虫，如搜索引擎
* 聚焦爬虫，如12306抢票，或者专门抓取某一网站的某一类数据

### 根据是否以获取数据为目的，可以分为：

* 功能性爬虫，给你喜欢的明星，投票点赞
* 数据增量式爬虫，比如招聘信息

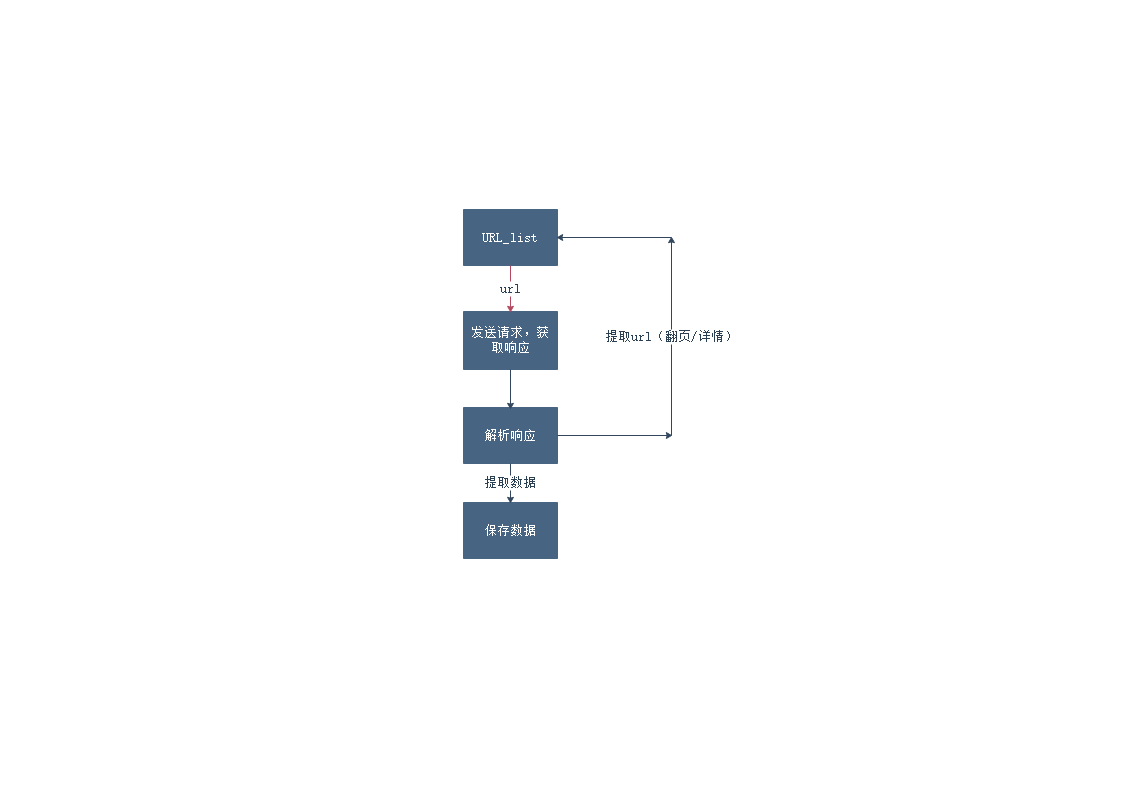
### 根据URL地址和对应页面内容是否改变，数据增量爬虫可以分为：

* 基于URL地址变化，内容变化的增量式爬虫
* URL地址不变，内容变化的数据增量式爬虫



**了解爬虫分类**

## 爬虫流程



1、获取一个URL

2、向URL发送请求，并获取响应（http协议）

3、如果从响应中提取URL，则继续发送请求获取响应

4、如果从响应中获取数据，则数据进行保存

**掌握爬虫流程**

## http以及https的概念和区别

在爬虫流程的第二步，向URL发送请求，那么就要依赖于HTTP/HTTPS协议。

HTTPS比HTTP更安全，但是性能更低

* HTTP:超文本传输协议，默认端口为80
* 。超文本：是指超过文本，不限于文本，可以传输图片、视频、音频等数据
* 。传输协议：是指使用公共约定的固定格式来传递转换成字符串的超文本内容
* HTTPS:HTTP+SSL（安全套接字），即带有安全套接字层的超文本传输协议，默认端口443
* 。SSL对传输内容（超文本，也就是请求头和响应体）进行加密
* 可以打开一个浏览器访问URL，右键检查，点击network，选择一个URL，查看HTTP协议的形式。

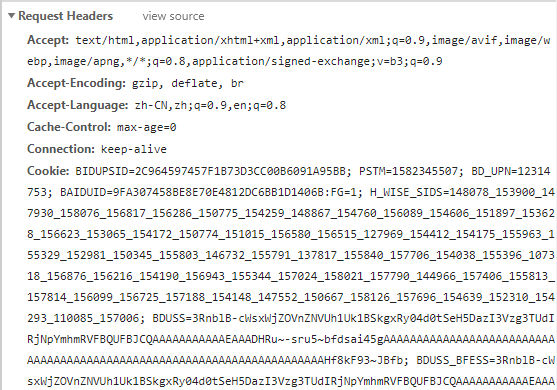
**掌握http及https的概念和默认端口**

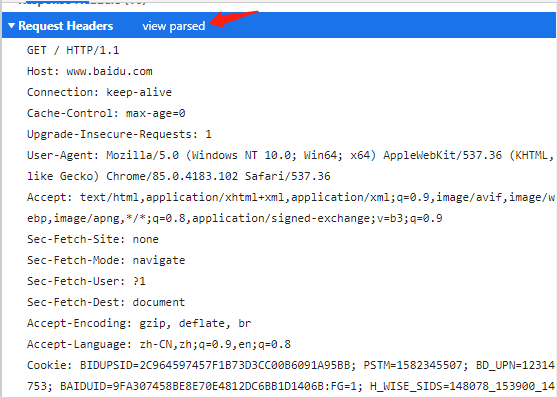
## 爬虫特别注意的请求头



http请求形式如上图所示，爬虫要特别关注以下几个请求头字段

* Content-Type
* Host
* Connection
* Upgrade-Insecure-Requests(升级为https请求)
* **User-Agent**（用户代理）
* **Referer**
* **Cookie**（保持用户状态）
* Authorization（认证信息）

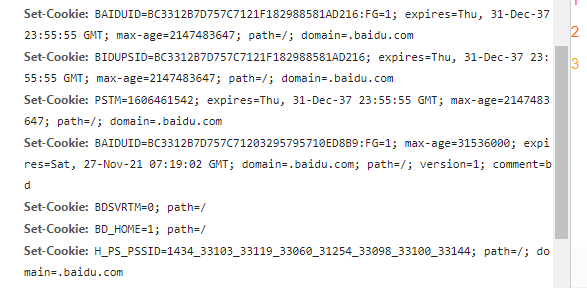
例如，使用浏览器访问百度进行抓包  




当我点击view source的时候，就会出现另外一种格式的请求头，这个是原始的版本，如果没有点击view source的请求头格式是经过浏览器优化的。

## 爬虫特别注意的响应头

* set-cookie



cookie是基于服务端生成的，在客户端头信息中，在第一次把请求发送到服务端，服务端生成cookie，存放到客户端，下次发送请求时会带上cookie。

## 常见的响应状态码

* 200：成功
* 302：跳转，新的URL在响应中的Location头中给出
* 303：浏览器对于post响应进行重定向至新的URL
* 307：浏览器对于get响应进行重定向至新的URL
* 403：资源不可用，服务器理解客户端的请求，但拒绝处理它（没有权限）
* 404：找不到页面
* 500：服务器内部错误
* 503：服务器由于维护或者负载过重未能应答。在响应中可能会携带Retry-After响应头，有可能是因为爬虫频繁访问URL，使服务器忽视爬虫的请求，最终返回503状态码

**所有的状态码都不可信，一切要以抓包得到的响应中获取的数据为准**

network中抓包得到的源码才是判断依据。element中的源码是渲染之后的源码，不能作为判断标准。

**了解常见的响应状态码**

## http请求的过程

1、浏览器在拿到域名对应的IP之后，先向地址栏中的URL发起请求，并获取响应。

2、在返回响应内容（HTML）中，会带有CSS、JS、图片等URL地址，以及Ajax代码，浏览器按照响应内容中的顺序依次发送其他请求，并获取响应。

3、浏览器每获取一个响应就对展示出的结果进行添加（加载），JS、CSS等内容会修改页面内容，JS也可以重新发送请求，获取响应。

4、从获取第一个响应并在浏览器中展示，直到最终获取全部响应，并在展示结果中添加内容或修改，这个过程叫做浏览器的**渲染**。

## 注意

在爬虫中，爬虫只会请求URL地址，对应的拿到URL地址对应的响应（该响应可以是HTML、CSS 、JS或是是图片、视频等等）。

浏览器渲染出来的页面和爬虫请求抓取的页面很多时候是不一样的，原因是爬虫不具有渲染功能。

* **浏览器最终展示的结果是由多次请求响应共同渲染的结果**
* **爬虫只对一个URL地址发起请求并得到响应**

**理解浏览器展示的结果可以是多次请求响应共同渲染的结果，而爬虫是一次请求对应一个响应。**

## 最后

**路漫漫其修远兮，吾将上下而求索！**

我是**啃书君**，一个专注于学习的人，**你懂得越多，你不懂得越多。**

更多精彩内容我们下期再见！

作者简介：我是一名大三学生，热爱编程与学习，希望可以将我的经验与学习的成果分享给大家。

联系方式：18613034807（微信同号）