一、HTTP 简介

1.URL

URL 的全称为Universal Resource Locator ，即统一资源定位符。通过这样的一个链接，我们便可以从互联网上查找到我们需要的资源。在编写爬虫的过程中主要也是通过URL对我们需要的内容进行访问，从而爬取相应的信息。

2.请求方法

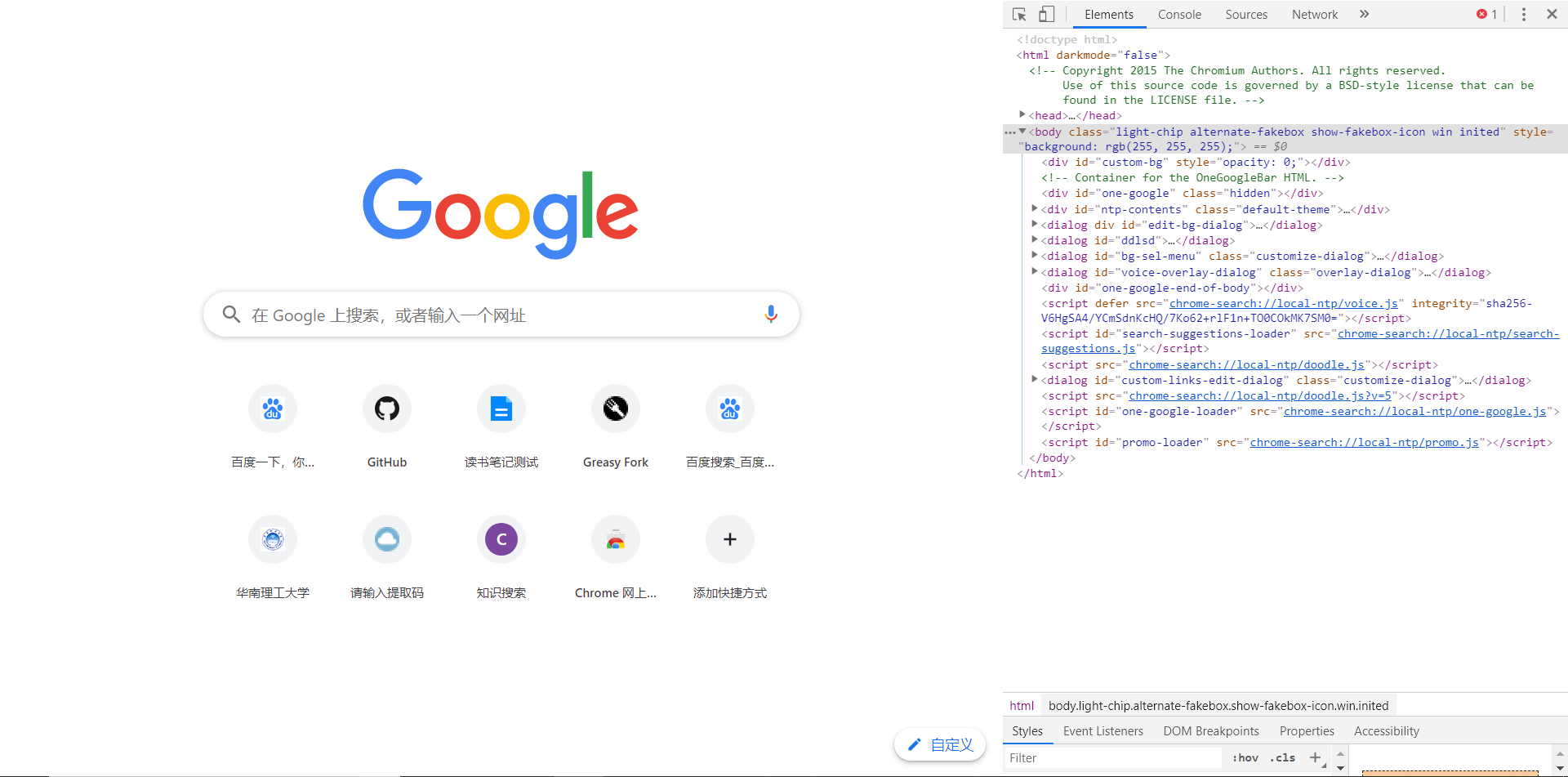
常见的是get和post方法。两者的区别是，get请求中的参数在URL里面可以看到，需要的话可以直接提取出来。但是post请求就有点区别，数据都在URL中是不可见的。

1. 网页的组成

网页组成分为三大部分——HTML , CSS 和JavaScript。

HTML全称Hyper Text Markup Language，即超文本标记语言。超文本标记语言定义了多种数据类型的元素内容，如脚本数据和样式表的数据，和众多类型的属性值，包括ID、名称、URI、数字、长度单位、语言、媒体描述符、颜色、字符编码、日期和时间等。所有这些数据类型都是专业的字符数据。

如下是谷歌浏览器初始界面网页的源代码。爬虫就是在网页源代码里面得到需要的信息。



CSS ，全称叫作Cascading Style Sheets ，即层叠样式表。它是一种用来表现HTML（标准通用标记语言的一个应用）或XML（标准通用标记语言的一个子集）等文件样式的计算机语言。CSS不仅可以静态地修饰网页，还可以配合各种脚本语言动态地对网页各元素进行格式化。JavaScript，简称JS，是一种脚本语言，实现了一种实时、动态、交互的页面功能。上述三部分相互结合才产生了最终我们看见的网页。

三、requests库

1.在requests库中每一个请求方法都有一个对应的API，比如GET请求就可以使用get()方法，POST请求就可以使用post()方法，并且将需要提交的数据传递给data参数即可。

而其他的请求类型，都有各自对应的方法：

response = requests.get(‘https://httpbin.org/get’)

response = requests.post(‘http://gttpbin.org/post’,data={‘key’:'value’})

- post请求四种传送正文方式：

　　- 请求正文是application/x-www-form-urlencoded

　　- 请求正文是multipart/form-data

　　- 请求正文是raw

　　- 请求正文是binary

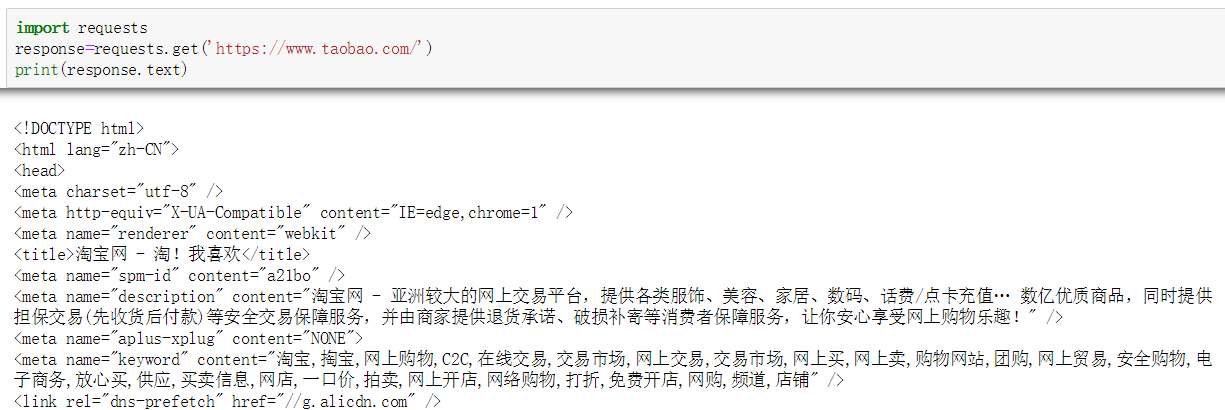
response = requests.put(‘http://httpbin.org/put’,data={‘key’:'value’})

response = requests.delete(‘http://httpbin.org/delete’)

response = requests.head('http://httpbin.org/get‘）

response = requests.options(‘http://httpbin.org/get’)

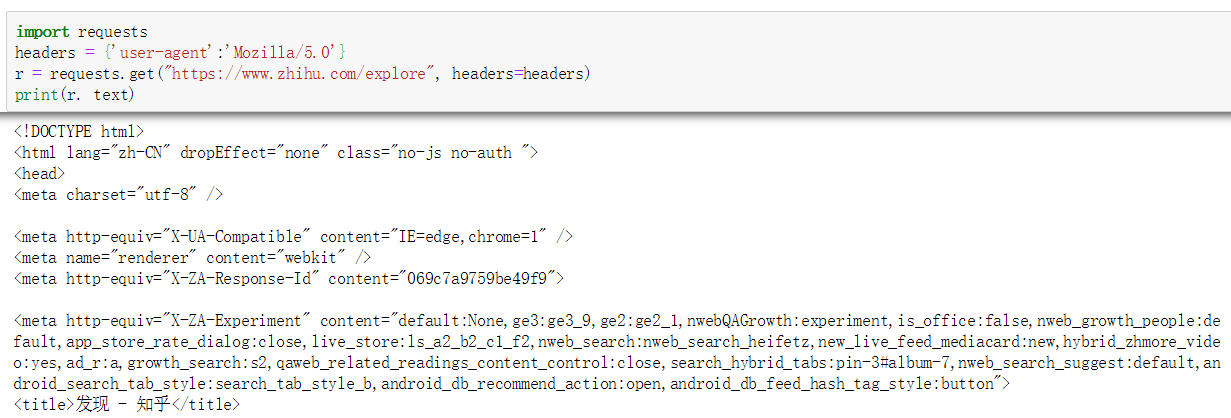
比如请求淘宝首页的信息。



如果我们需要访问首页的其他的内容，可以对请求的链接进行修改。

2.自定义Headers

如果想自定义请求的Headers，同样的将字典数据传递给headers参数，比如爬取知乎的信息。



3.数据类型

response = requests.get("https://www.zhihu.com/explore", headers=headers)

如果需要获得原始的二进制数据，使用content属性即可 response.content

如果我们访问之后获得的是json格式的，那么可以使用json()方法，直接获取转换成字典格式的数据 response.json()

四、Beauti ful Soup库简介

Beauti ful Soup 是Python 的一个HTML 或XML 的解析库，可以用它来方便地从网

页中提取数据。

1.调用beautifulsoup

from bs4 import BeautifulSoup

2.具体使用方法

soup=BeautifulSoup(html,'html.parser')

用 BeautifulSoup 可以很方便地获取文档中的标签：

soup.title、soup.head、soup.a、soup.p

soup.head.string可以返回头部结点里面具体的内容

soup.find\_all('div',attrs={'class':''})

就可以查找到网页中，所有“div”中class为“wz\_content”的部分，