**1.分分钟扒一个网页下来**

怎样扒网页呢？其实就是根据URL来获取它的网页信息，虽然我们在浏览器中看到的是一幅幅优美的画面，但是其实是由浏览器解释才呈现出来的，实质它 是一段HTML代码，加 JS、CSS，如果把网页比作一个人，那么HTML便是他的骨架，JS便是他的肌肉，CSS便是它的衣服。所以最重要的部分是存在于HTML中的，下面我 们就写个例子来扒一个网页下来。

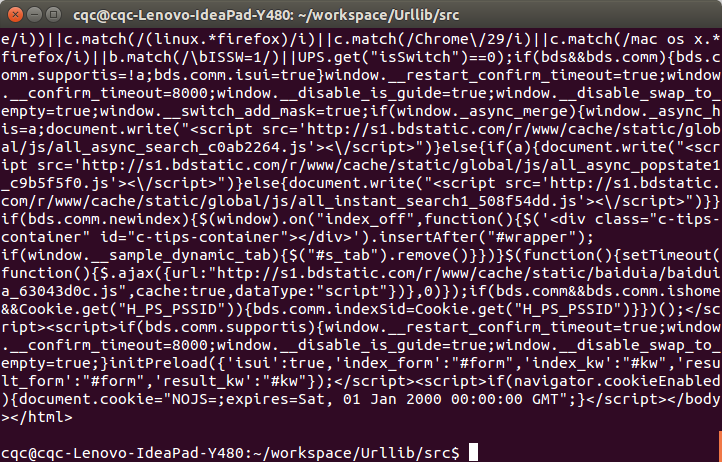
Python

|  |  |
| --- | --- |
| 1  2  3  4 | import urllib2    response = urllib2.urlopen("http://www.baidu.com")  print response.read() |

是的你没看错，真正的程序就两行，把它保存成 demo.py，进入该文件的目录，执行如下命令查看运行结果，感受一下。

Python

|  |  |
| --- | --- |
| 1 | python demo.py |



看，这个网页的[源码](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=8&fv=17&jk=5f209a25363e8ac5&k=%D4%B4%C2%EB&k0=%D4%B4%C2%EB&kdi0=0&luki=5&n=10&p=baidu&q=35091180_cpr&rb=0&rs=1&seller_id=1&sid=c58a3e36259a205f&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1959022&u=http%3A%2F%2Fcuiqingcai%2Ecom%2F947%2Ehtml&urlid=0" \t "_blank)已经被我们扒下来了，是不是很酸爽？

**2.分析扒网页的方法**

那么我们来分析这两行代码，第一行

Python

|  |  |
| --- | --- |
| 1 | response = urllib2.urlopen("http://www.baidu.com") |

首先我们调用的是urllib2库里面的urlopen方法，传入一个URL，这个网址是百度首页，协议是HTTP协议，当然你也可以把HTTP换做FTP,FILE,HTTPS 等等，只是代表了一种[访问控制](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=8&fv=17&jk=5f209a25363e8ac5&k=%B7%C3%CE%CA%BF%D8%D6%C6&k0=%B7%C3%CE%CA%BF%D8%D6%C6&kdi0=0&luki=1&n=10&p=baidu&q=35091180_cpr&rb=0&rs=1&seller_id=1&sid=c58a3e36259a205f&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1959022&u=http%3A%2F%2Fcuiqingcai%2Ecom%2F947%2Ehtml&urlid=0" \t "_blank)协议，urlopen一般接受三个参数，它的参数如下：

Python

|  |  |
| --- | --- |
| 1 | urlopen(url, data, timeout) |

第一个参数url即为URL，第二个参数data是访问URL时要传送的数据，第三个timeout是设置超时时间。

第二三个参数是可以不传送的，data默认为空None，timeout默认为 socket.\_GLOBAL\_DEFAULT\_TIMEOUT

第一个参数URL是必须要传送的，在这个例子里面我们传送了百度的URL，执行urlopen方法之后，返回一个response对象，返回信息便保存在这里面。

Python

|  |  |
| --- | --- |
| 1 | print response.read() |

response对象有一个read方法，可以返回获取到的网页内容。

如果不加read直接打印会是什么？答案如下：

Python

|  |  |
| --- | --- |
| 1 | <addinfourl at 139728495260376 whose fp = <socket.\_fileobject object at 0x7f1513fb3ad0>> |

直接打印出了该对象的描述，所以记得一定要加read方法，否则它不出来内容可就不怪我咯！

**3.构造Requset**

其实上面的urlopen参数可以传入一个request请求,它其实就是一个Request类的实例，构造时需要传入Url,Data等等的内容。比如上面的两行代码，我们可以这么改写

Python

|  |  |
| --- | --- |
| 1  2  3  4  5 | import urllib2    request = urllib2.Request("http://www.baidu.com")  response = urllib2.urlopen(request)  print response.read() |

运行结果是完全一样的，只不过中间多了一个request对象，推荐大家这么写，因为在构建请求时还需要加入好多内容，通过构建一个request，[服务器](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=8&fv=17&jk=5f209a25363e8ac5&k=%B7%FE%CE%F1%C6%F7&k0=%B7%FE%CE%F1%C6%F7&kdi0=0&luki=6&n=10&p=baidu&q=35091180_cpr&rb=0&rs=1&seller_id=1&sid=c58a3e36259a205f&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1959022&u=http%3A%2F%2Fcuiqingcai%2Ecom%2F947%2Ehtml&urlid=0" \t "_blank)响应请求得到应答，这样显得逻辑上清晰明确。

**4.POST和GET数据传送**

上面的程序演示了最基本的网页抓取，不过，现在大多数网站都是动态网页，需要你动态地传递参数给它，它做出对应的响应。所以，在访问时，我们需要传递数据给它。最常见的情况是什么？对了，就是登录注册的时候呀。

把数据用户名和密码传送到一个URL，然后你得到服务器处理之后的响应，这个该怎么办？下面让我来为小伙伴们揭晓吧！

数据传送分为POST和GET两种方式，两种方式有什么区别呢？

最重要的区别是GET方式是直接以链接形式访问，链接中包含了所有的参数，当然如果包含了密码的话是一种不安全的选择，不过你可以直观地看到自己提交了什么内容。POST则不会在网址上显示所有的参数，不过如果你想直接查看提交了什么就不太方便了，大家可以酌情选择。

**POST方式：**

上面我们说了data参数是干嘛的？对了，它就是用在这里的，我们传送的数据就是这个参数data，下面[演示](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=8&fv=17&jk=5f209a25363e8ac5&k=%D1%DD%CA%BE&k0=%D1%DD%CA%BE&kdi0=0&luki=2&n=10&p=baidu&q=35091180_cpr&rb=0&rs=1&seller_id=1&sid=c58a3e36259a205f&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1959022&u=http%3A%2F%2Fcuiqingcai%2Ecom%2F947%2Ehtml&urlid=0" \t "_blank)一下POST方式。

Python

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | import urllib  import urllib2    values = {"username":"1016903103@qq.com","password":"XXXX"}  data = urllib.urlencode(values)  url = "https://passport.csdn.net/account/login?from=http://my.csdn.net/my/mycsdn"  request = urllib2.Request(url, data)  response = urllib2.urlopen(request)  print response.read() |

我们引入了urllib库，现在我们模拟登陆CSDN，当然上述代码可能登陆不进去，因为还要做一些设置头部header的工作，或者还有一些参数 没有设置全，还没有提及到在此就不写上去了，在此只是说明登录的原理。我们需要定义一个字典，名字为values，参数我设置了username和 password，下面利用urllib的urlencode方法将字典编码，命名为data，构建request时传入两个参数，url和data，运 行程序，即可实现登陆，返回的便是登陆后呈现的页面内容。当然你可以自己搭建一个[服务器](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=8&fv=17&jk=5f209a25363e8ac5&k=%B7%FE%CE%F1%C6%F7&k0=%B7%FE%CE%F1%C6%F7&kdi0=0&luki=6&n=10&p=baidu&q=35091180_cpr&rb=0&rs=1&seller_id=1&sid=c58a3e36259a205f&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1959022&u=http%3A%2F%2Fcuiqingcai%2Ecom%2F947%2Ehtml&urlid=0)来测试一下。

注意上面字典的定义方式还有一种，下面的写法是等价的

Python

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | import urllib  import urllib2    values = {}  values['username'] = "1016903103@qq.com"  values['password'] = "XXXX"  data = urllib.urlencode(values)  url = "http://passport.csdn.net/account/login?from=http://my.csdn.net/my/mycsdn"  request = urllib2.Request(url,data)  response = urllib2.urlopen(request)  print response.read() |

以上方法便实现了POST方式的传送

**GET方式：**

至于GET方式我们可以直接把参数写到网址上面，直接构建一个带参数的URL出来即可。

Python

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | import urllib  import urllib2    values={}  values['username'] = "1016903103@qq.com"  values['password']="XXXX"  data = urllib.urlencode(values)  url = "http://passport.csdn.net/account/login"  geturl = url + "?"+data  request = urllib2.Request(geturl)  response = urllib2.urlopen(request)  print response.read() |

你可以print geturl，打印输出一下url，发现其实就是原来的url加？然后加编码后的参数

Python

|  |  |
| --- | --- |
| 1 | http://passport.csdn.net/account/login?username=1016903103%40qq.com&password=XXXX |

和我们平常GET访问方式一模一样，这样就实现了数据的GET方式传送。

本节[讲解](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=8&fv=17&jk=5f209a25363e8ac5&k=%BD%B2%BD%E2&k0=%BD%B2%BD%E2&kdi0=0&luki=3&n=10&p=baidu&q=35091180_cpr&rb=0&rs=1&seller_id=1&sid=c58a3e36259a205f&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1959022&u=http%3A%2F%2Fcuiqingcai%2Ecom%2F947%2Ehtml&urlid=0)了一些基本使用，可以抓取到一些基本的网页信息，小伙伴们加油！