# [动态渲染页面爬取](https://cuiqingcai.com/5627.html)

在前一章中，我们了解了Ajax的分析和抓取方式，这其实也是JavaScript动态渲染的页面的一种情形，通过直接分析Ajax，我们仍然可以借助requests或urllib来实现数据爬取。

不过JavaScript动态渲染的页面不止Ajax这一种。比如中国青年网（详见[http://news.youth.cn/gn/](http://news.youth.cn/gn/" \o ")），它的分页部分是由JavaScript生成的，并非原始HTML代码，这其中并不包含Ajax请求。比如ECharts的官方实例（详见[http://echarts.baidu.com/demo.html#bar-negative](http://echarts.baidu.com/demo.html" \l "bar-negative" \o ")），其图形都是经过JavaScript计算之后生成的。再有淘宝这种页面，它即使是Ajax获取的数据，但是其Ajax接口含有很多加密参数，我们难以直接找出其规律，也很难直接分析Ajax来抓取。

为了解决这些问题，我们可以直接使用模拟浏览器运行的方式来实现，这样就可以做到在浏览器中看到是什么样，抓取的源码就是什么样，也就是可见即可爬。这样我们就不用再去管网页内部的JavaScript用了什么算法渲染页面，不用管网页后台的Ajax接口到底有哪些参数。

Python提供了许多模拟浏览器运行的库，如Selenium、Splash、PyV8、Ghost等。本章中，我们就来介绍一下Selenium和Splash的用法。有了它们，就不用再为动态渲染的页面发愁了。