**通过AJAX发送数据**

**什么是AJAX**

* 读音：['eidʒæks]。
* AJAX即“Asynchronous JavaScript and XML”（异步JavaScript和XML)，AJAX并非缩写词，而是由Jesse James Gaiiett创造的名词，是指一种创建交互式网页应用的网页开发技术。
* 国内翻译常为“阿贾克斯”和阿贾克斯足球队同音。

**AJAX的作用**

* AJAX**不是一种新的编程语言**，而是一种用于创建更好更快以及交互性更强的 Web 应用程序的技术。
* 通过 AJAX，可使用JavaScript的XMLHttpRequest对象来直接与服务器进行通信。通过这个对象， JavaScript 可在不重载页面的情况与Web服务器交换数据。
* AJAX 在浏览器与 Web 服务器之间使用异步数据传输（HTTP 请求），这样就可使网页从服务器请求少量的信息，而不是整个页面。
* AJAX 可使因特网应用程序更小、更快，更友好。
* 从本质上讲，AJAX技术只不过是一种无刷新页面即可从服务器上加载数据的手段。
* 案例

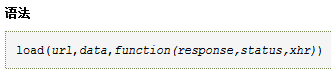


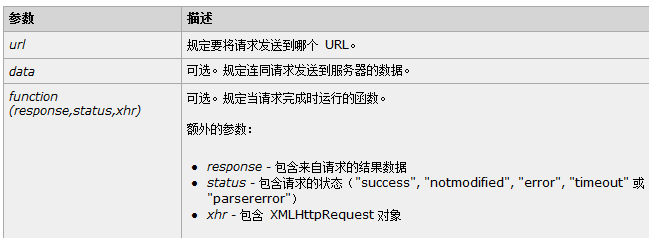
* 案例



**追加HTML**

* load() 方法通过 AJAX 请求从服务器加载数据，并把返回的数据放置到指定的元素中。





* 例6-1

**$(document).ready(function() {**

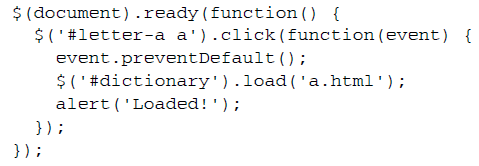
**$('#letter-a a').click(function(event) {**

**event.preventDefault();**

**$('#dictionary').load('a.html');**

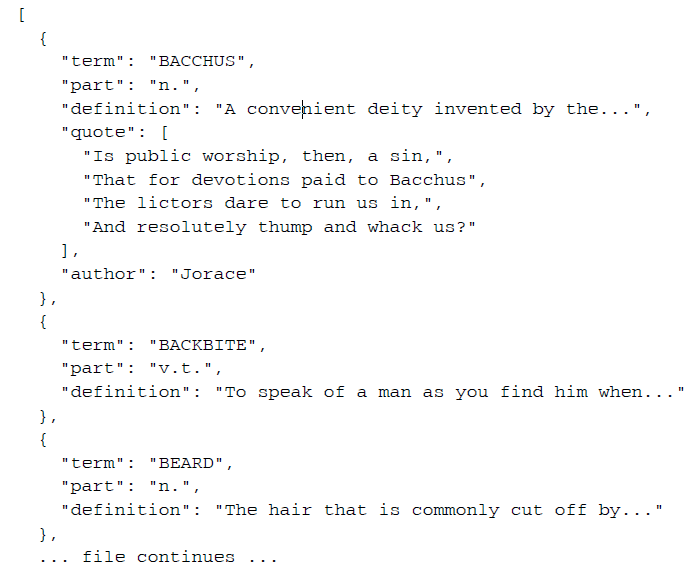
**});**

**});**

* 所有Ajax请求在默认情况下都是**异步**的
* 代码清单6-2

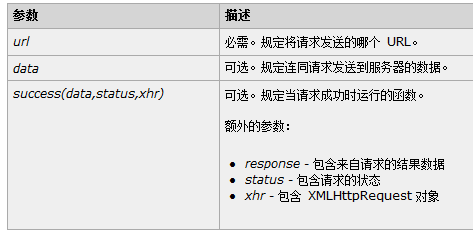
**操作JavaScript对象**

* 通过请求获取充分格式化的HTML虽然很方便，但这也意味着必须在传输文本内容的同时也传输很多HTML标签。有时候，我们希望能够尽量少传输一些数据，然后马上处理这些数据。

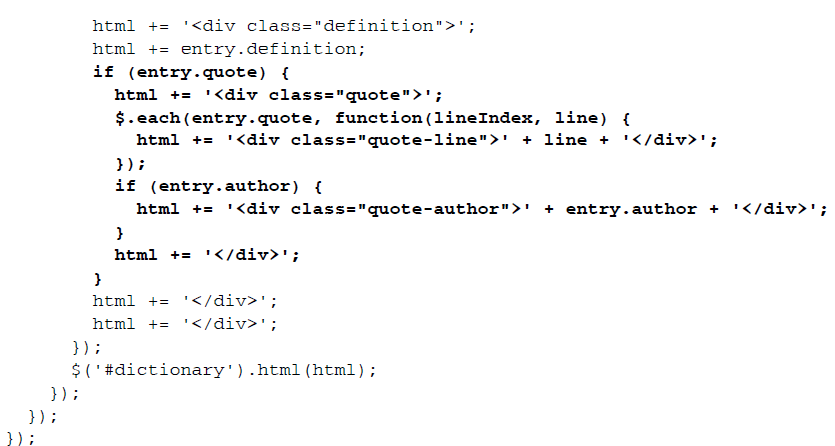
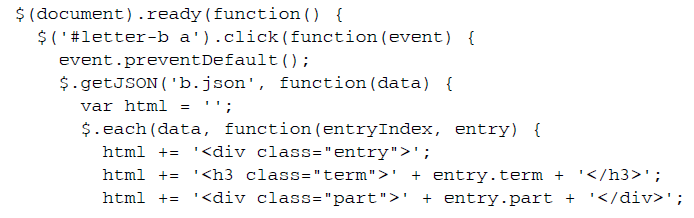


**操作JavaScript对象**

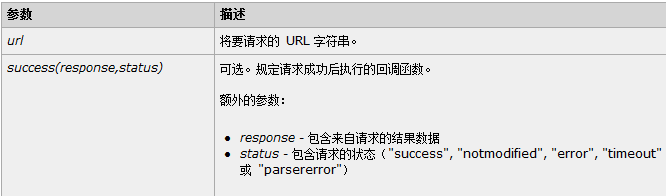
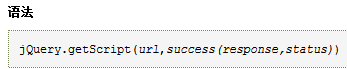
* 1取得JSON
* jQuery.getJSON(*url*,*data*,*success(data,status,xhr)*)
* 通过 HTTP GET 请求载入 JSON 数据

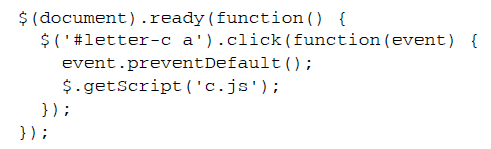


* 2$.each()
* 该函数不操作jQuery对象，它以数组或映射作为第一个参数，以回调函数作为第二个参数
* 代码6-6



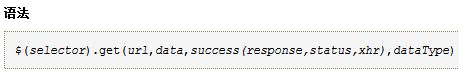
* 3执行脚本
* 有时候，在页面初次加载时就取得所需全部Javascript也是没有必要的。具体需要取得哪个脚本，要视用户的操作而定。虽然可以在需要时动态地引入<script>标签，但更优雅的方式是通过Jquery直接加载.js文件
* getScript() 方法通过 HTTP GET 请求载入并执行 JavaScript 文件。



**代码6-7**

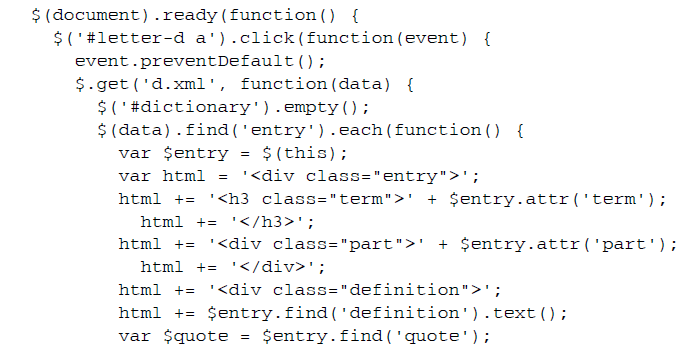
**加载XML文档**

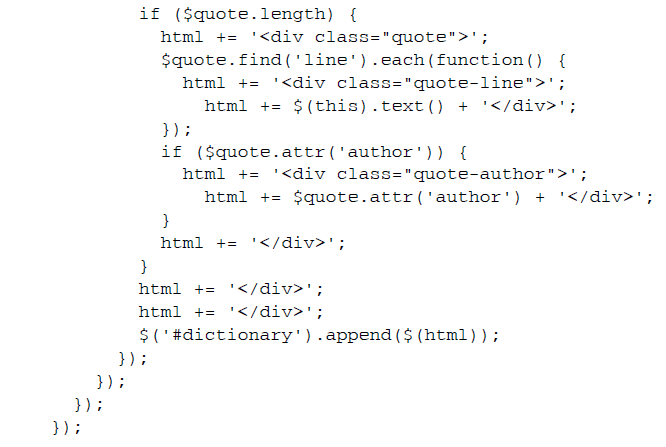
* 我们可以以XML文档的形式替换我们前面使用的JSON。
* get() 方法通过远程 HTTP GET 请求载入信息。





* 例6-9



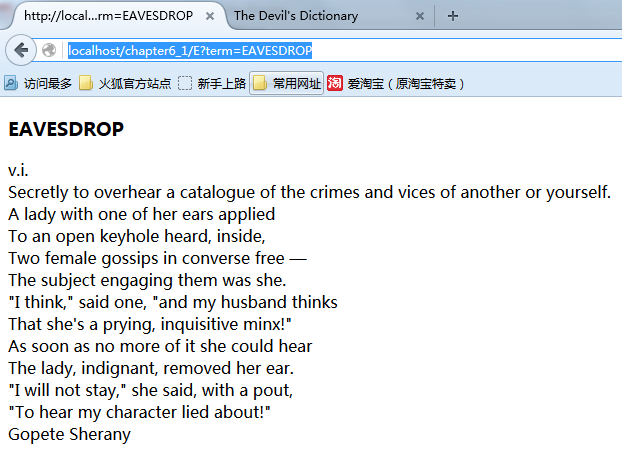


**选择数据格式**

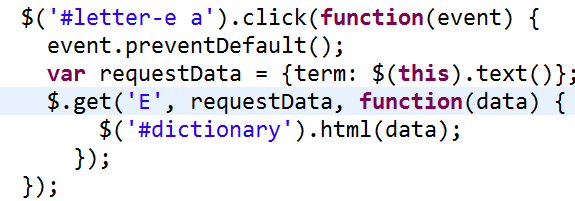
* 以上4种外部数据格式各有优缺点，在应用程序中应灵活使用。
* HTML片段实现起来只需很小的工作量，使用简单，但数据结构不一定在其他应用程序中得到重用。
* JSON文件简洁，容易阅读，方便重用。但是JSON文件中的错误可能会导致页面上的脚本静默地中止运行。
* JavaScript文件能够提供极大的灵活性，但它却不是一种真正的数据存储机制。这种文件针对特定的语言，不能通过它们将同样的信息提供给不同的系统。
* XML是一种通用的数据格式，但是Javascript开发人员更钟爱JSON，XML的文件体积相对较大，解析和操作它们的速度要慢一些。

**向服务器传递数据**

* 此前，我们的例子都是从Web服务器上取得静态的数据文件。然而，Ajax的价值只有当服务器能够基于浏览器的输入动态形成数据时才能得到充分体现。在这种情况下，jQuery同样也能为 我们提供帮助；前面介绍的所有方法在经过修改之后，都可以实现双向的数据传送。
* 为了示范客户端（使用JavaScript）与服务器（在我们这个例子中使用JSP）之间的通信，我们要编写一个基于每次请求只向浏览器发送一个字典词条的脚本。词条的选择取决于从浏览器发送到服务器的参数。
* 见E.java



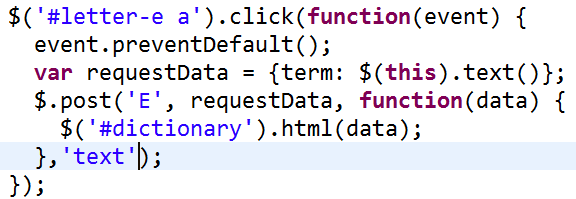
* http://localhost/chapter6\_1/E?term=EAVESDROP
* 代码清单6-10



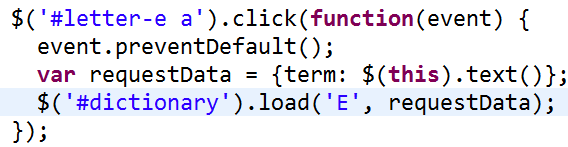
**执行POST请求**

* 使用POST方法与使用GET方法的HTTP请求几乎是一样的。从视觉上来看，它们之间一个最大的区别就是GET请求把参数放在作为URL一部分的查询字符串中，而POST请求则不是。但是，在Ajax请求中，即使是这种区别对一般用户而言也是不可见的。
* 因此只需改变调用的jQuery函数，就可以在GET和POST之间进行转换.

**代码清单6-11**

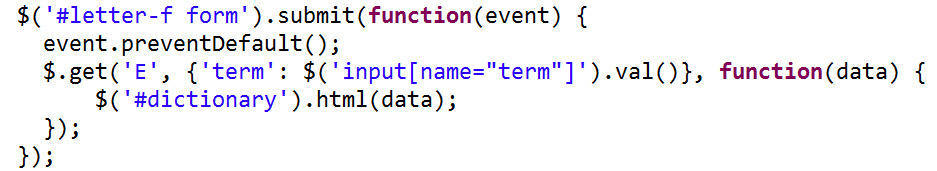


* 虽然参数相同，但这里的请求是通过POST方法发送的。而通过使用.load()方法还可以进一步简化这些代码，因为.load()方法在接收到包含数据的对象参数时，会默认使用POST方法发送请求，

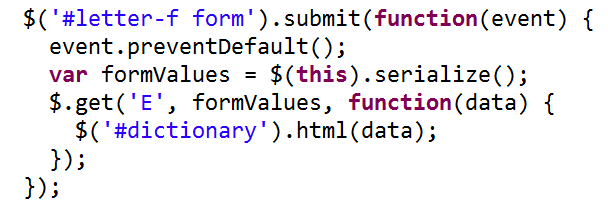
**代码清单6-12**

**序列化表单**

* 向服务器发送数据经常会涉及用户填写表单。常规的表单提交机制会在整个浏览器窗口中加载响应，而使用jQuery的Ajax工具箱则能够异步地提交表单，并将响应放到当前页面中。
* 代码清单6-13



* 虽然以上代码能够实现预期的效果，但通过名称属性逐个搜索输入字段并将字段的值添加到对象中总是有点麻烦。特别是随着表单变得更复杂，这种方法也会明显变得缺乏扩展性。
* jQuery为这种常用的操作提供了一种简化方式**——.serialize()**方法。
* 代码清单6-14



**为Ajax 请求提供不同的内容**

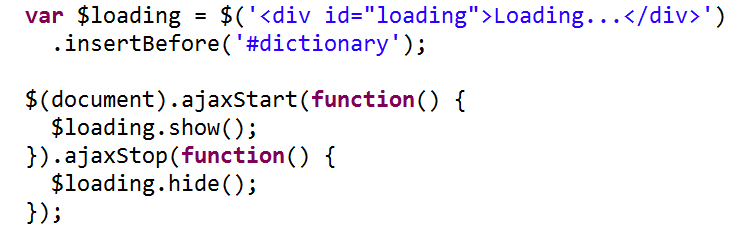
* 在返回HTML数据的情况下，我们知道如果只让浏览器自己打开页面而不使用JavaScript，那么没有样式的文档片段会很难看。为了给没有JavaScript用户提供比这里更好的体验，可以有条件的加载包含<html>、<head>和<body>以及其他所有内容的完整的页面。为些，就要利用jQuery随同Ajax请求一起发送的请求头部。在服务器端代码中，我们需要检查**X-Requested- With**头部。如果存在这个头部而且它的值为**XMLHttpRequest**，那么就会只发送文档片段；否则，就发送完整的文档。在jsp中，使用下面代码可得到对应头信息
* **request.getHeader("x-requested-with");**
* 关注请求
* ，jQuery提供了一组函数，通过它们能够为各种与Ajax相关的事件注册回调函数。
* **.ajaxStart()**和**.ajaxStop()**方法就是这些“观察员”函数中的两个例子，可以把它们添加给任何jQuery对象。当Ajax请求开始且尚未进行其他传输时，会触发.ajaxStart()的回调函数。相反，当最后一次活动请求终止时，则会执行通过.ajaxStop()注册的回调函数。所有这些“观察员”都是全局性的，因为无论创建它们的代码位于何处，当Ajax通信发生时都需要调用它们。而且这些方法都与.ready()方法一样，只能由$(document)调用。
* 页面中用作反馈的HTML可以包含适当的Loading…（加载中）信息

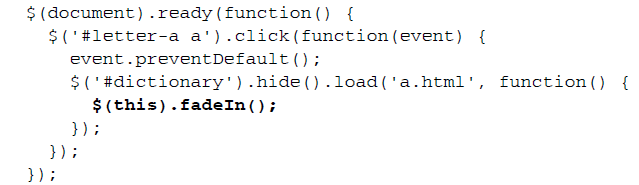
**<div id="loading">**

**Loading...**

**</div>**

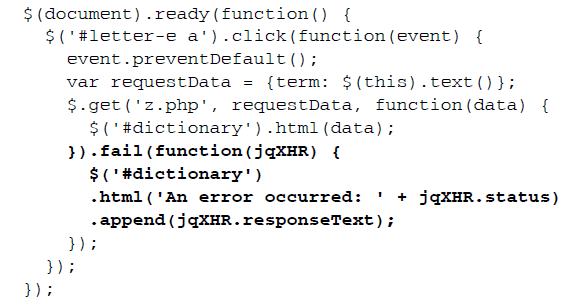
* 代码清单6-15



* 代码清单6-16
* 通过使加载的内容淡入视图而不是突然出现，可以从视觉上加入一些增强的效果。

**错误处理**

* 除了使用全局的.ajaxError()方法，我们还可以利用jQuery的延迟对象系统。可以给.load()之外的Ajax方法连缀**.done()、.always()和.fail()**方法，并通过它们添加相应的回调函数即可。比如，可以在代码清单6-16的基础上，把URL改为一个不存在的地址，然后测试.fail()方法，如代码清单6-17所示。现在，单击字母E开头的任何链接都会产生一个错误，



**Ajax 和事件**

* 假设我们想让字典词条名来控制后面解释的可见性，即单击词条名可以显示或隐藏相应的解释。使用到目前为止介绍的技术，实现这一点应该很简单

**$(document).ready(function() {**

**$('h3.term').click(function() {**

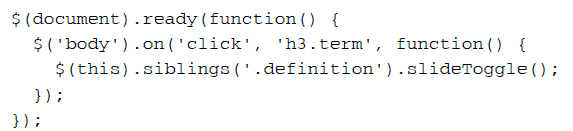
**$(this).siblings('.definition').slideToggle();**

**});**

**});**

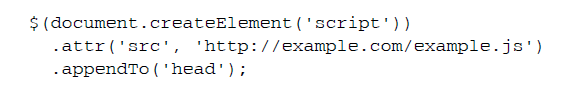
虽然看起来一切正常，但在现有代码基础上单击不会有什么结果。因为在添加click处理程序的时候，词条还没有被添加到文档中。而且即使已经把click处理程序添加到词条元素，只要一单击其他字母，这些处理程序仍然会丢失绑定。这是通过Ajax生成页面内容时的一个常见问题。对此，一种常见的解决方案就是在页面内容更新时重新绑定处理程序。但这样做会相当繁琐，因为哪怕页面的DOM结构有一点点变化，都会调用绑定处理程序的代码。另外一种值得推荐的做法是第3章介绍的**事件委托**。

* 代码清单6-19



**安全限制**

* 尽管构建动态的Web应用程序非常实用，但XMLHttpRequest（jQuery的Ajax实现背后的底层浏览器技术）常常会受到严格限制。为了防止各种跨站点脚本攻击，一般情况下从提供原始页面的服务器之外点请求文档是不可能的。
* 但是，从第三方来源中加载数据往往是很有必要的。因而，也有许多方式可以绕过上述安全限制，即能够实现通过Ajax请求取得其他站点的数据。
* 其中一种方法是通过服务器加载远程数据，然后在客户请求时提供给浏览器。
* 如果想不通过服务器的参与加载远程地址中的数据，加载外来JavaScript文件的一种流行方法是根据请求注入<script>标签，由于jQuery能帮我们插入新的DOM元素，因此向文档中注入<script>标签非常简单



* 另一种方法是使用<iframe>这个HTML标签来加载远程数据。可以为<iframe>元素指定任何URL作为其获取数据的来源，包括与提供页面的服务器不匹配的URL。

**使用JSONP加载远程数据**

* 使用<script>标签从远程获取JavaScript文件的思路，可以变通为从其他服务器取得JSON文件。不过，这样需要对服务器上的JSON文件稍加修改。在实现这一技术的众多解决方案中，jQuery直接支持的是JSONP（JSON with Padding，填充式JSON）。JSONP的格式是把标准JSON文件包装在一对圆括号中，圆括号又前置一个任意字符串。这个字符串，即所谓的P（Padding，填充），由请求数据的客户端来决定。
* 用JSP在服务器端实现对JSONP的支持非常简单：

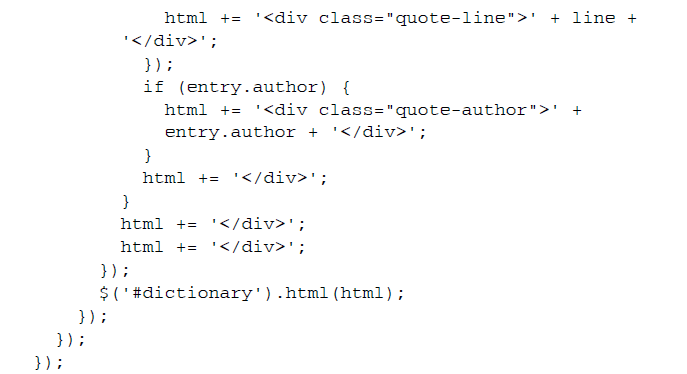
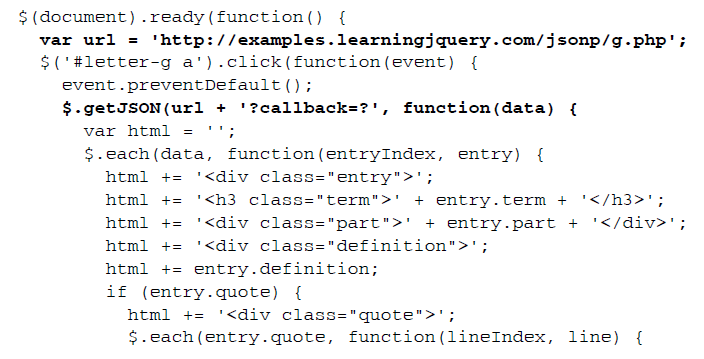
**<%**

**out.print(request.getParameter(“callback”)**

**+’(‘+data+’)’);**

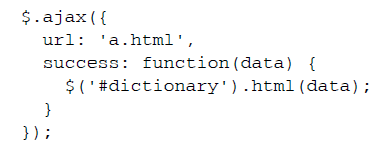
**%>**

* data是一个包含JSON文件字符串表示的变量。调用这段脚本时，从客户端请求中取得的callback查询字符串参数，会被添加到包含JSON数据文件的圆括号前面。
* 修改代码清单6-6中的JSON示例，以便调用这个远程数据源。$.getJSON()函数利用了一个特殊的占位符?来实现这一点，参见代码清单6-20。



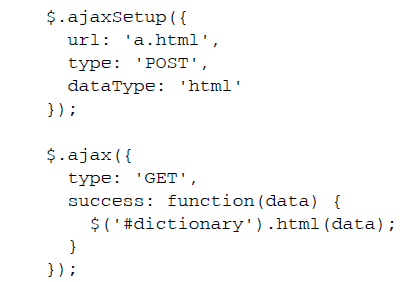
**其他工具**

* 低级Ajax方法
* 前面已经介绍了一些用于启动Ajax通信的方法。但在内部，jQuery会把这些方法都转变为$.ajax()全局函数的一种变体。
* 例如使用$('#dictionary').load('a.html')加载HTML片段。同样的操作如果使用$.ajax()来实现



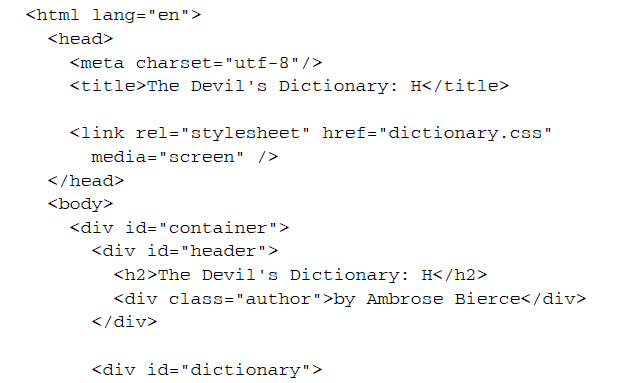
* 这里，$.ajax()接受了一个包含30余项设置（settings）的对象作为参数（或者一个URL字符串作为第一个参数，一个对象作为第二个参数），提供了极大的灵活性。使用低级的$.ajax()函数时，可以获得下列特殊的好处。
* 避免浏览器缓存来自服务器的响应。非常适合服务器动态生成数据的情况。
* 抑制正常情况下所有Ajax交互都可以触发的全局处理程序（例如通过$.ajaxStart()注册的处理程序）。
* 在远程主机需要认证的情况下，可以提供用户名和密码。

**修改默认选项**

* 使用$.ajaxSetup()函数可以修改调用Ajax方法时每个选项的默认值。这个函数与$.ajax()接受相同的选项对象参数，之后的所有Ajax请求都将使用传递给该函数的选项——除非明确覆盖
* 代码清单6-22

**部分加载HTML页面**

* 本章讨论的第一种，也是最简单的一种Ajax技术，就是取得并将HTML片段插入到当前页面中。不过，有时候服务器提供的页面中虽然包含我们需要的部分，但该部分之外的HTML却不是我们所需要的。当遇到这种服务器不能提供适当的数据格式的情况时，也可以在客户端求助于jQuery
* 如果在本章第一个例子中，我们需要的字典解释包含在如下所示的完整的HTML页面（h.html）中户端求助于jQuery。



* 要去掉页面中多余的内容，可以利用.load()的一些新特性——在指定要加载文档的URL时，也可以提供一个jQuery选择符表达式。如果指定了这个表达式，.load()方法就会利用它查找加载文档的匹配部分。最终，只有匹配的部分才会被插入到页面中
* 代码清单6-24

