学习前端的过程中，经常会听人提到HTTP协议，以及听起来挺唬人的Ajax。这里简单作个简单介绍，为进一步学习打下基础，也方便自己随时进行查看。

     
 1.协议和HTTP协议。

     
 说到协议，我们并不是很陌生，日常生活中就存在各种各样的协议。比如说购房合同，工作合同等等。通俗来说，协议按规矩办事。它规定的就是谁该做什么，具体怎么做（格式，次序）等。

     
 计算机中的协议同样如此。网络中的内容之所以能够在全球互通，各种协议功不可没。对于HTTP协议，它规定了在网络中发布、传输和接收html页面的方法。大家都遵循这个协议，就能实现信息的传输。

     
 2.HTTP协议典型的工作流程。

     
 客户端和服务器端没有任何联系——建立连接，客户端发送请求——沿着建立好的连接，服务器端返回响应信息——断开连接。

     
 3.HTTP请求信息的格式。

     
 请求信息分为三部分：请求行、请求头信息和请求主体信息。请求头信息和请求主体信息之间用一个空行分割，不管是否有请求主体信息，这个空行都必须存在。

     
 请求行又分成三部分：请求方法，请求资源的路径，所用协议的版本(目前一般是http1.1，0.9和1.0已经基本不用了)。

     
 请求方法又有以下几种：GET/POST/PUT/DELETE/TRACE/OPTIONS等。

     
 get方式通过地址栏传递参数，post方式是通过请求头信息传递信息的。但是这两种方式传递数据的格式都是相同的。key=value&key=value

     
 4.HTTP响应信息的格式。

     
 响应信息同样分成三部分：响应行，响应头信息，响应主体信息。

     
 响应行：协议版本，状态码，状态描述信息。

     
 5.常见的http状态码：

     
 200 OK，表示一切正常。对GET和POST请求的应答文档跟在后面。

     301 Moved  
Permanently。永久重定向。客户请求的文档在其他地方，新的URL在Location头中给出，浏览器应该自动地访问新的URL。

     302  
Found。临时重定向，新的URL被视为临时性的替代，而不是永久性的。

     304 Not  
Modified。客户端有缓存的文档并发出了一个条件性的请求（一般是提供If-Modified-Since头表示客户只想比指定日期更新的文档）。服务器告诉客户，原来缓存的文档还可以继续使用。

     400 Bad  
Request。请求出现语法错误。

     403  
Forbidden。资源不可用。

     404 Not  
Found。无法找到指定位置的资源。

     405 Method  
Not Allowed。请求方法（GET、POST、HEAD、Delete、PUT、TRACE等）对指定的资源不适用。

     414 Request  
URI Too Long。输入的网站URI太长。

     500  
Internal Server Error。服务器遇到了意料不到的情况，不能完成客户的请求

     501 Not  
Implemented。服务器不支持实现请求所需要的功能。例如，客户发出了一个服务器不支持的PUT请求

     505 HTTP  
Version Not Supported。服务器不支持请求中所指明的HTTP版本。

     
 6.HTTP协议的头信息

（1）Refer信息是一个url，可以知道是从哪个页面跳转到当前页面的，也就是用户来到当前页面的上一页是哪里。通过这个特性结合服务器配置，可以用来防盗链。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Ajax\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

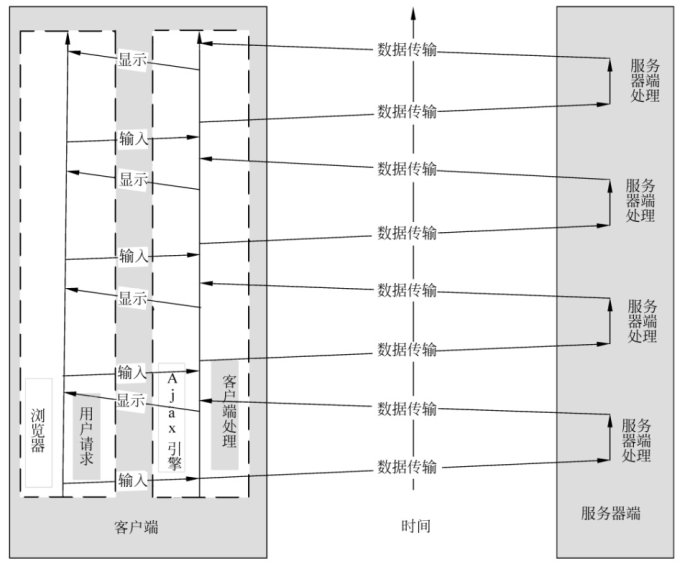
     
 1.Ajax是个极其热门的概念，大家也都谈论的很欢，那么它到底是个什么东西？

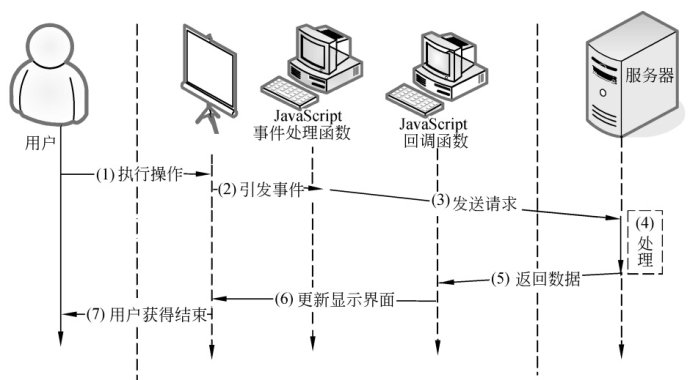
     
 一般的资料上是这样写的：Ajax是几个英文单词的首字母缩写，说得具体一些，就是Asynchronous  
Javascript And  
XML”（异步JavaScript和XML）。说实话这样并不容易让一个对此完全没概念的人有什么收获，查了资料跟没查资料差不多，我还是彻彻底底的不知道这玩意儿有什么神奇之处。

     
 经过一番折腾，我对它有了一点儿粗浅的认识，暂时先把现在的理解写在这里，如果以后发现不正确或者不合适，再进行修改。我想可以这样解释：Ajax能够在不刷新整个页面的情况下，跟后端的服务器进行通信，从而获得当前页面请求的某些内容，然后对页面进行部分更新。因为一张网页的大部分地方在加载之后是不需要更新的，这种方式减少了很多没有意义的数据传输，可以提高效率，也能提升用户体验。一句话概括，Ajax能完成的就是一件事，进行更有效的前后端数据交互。

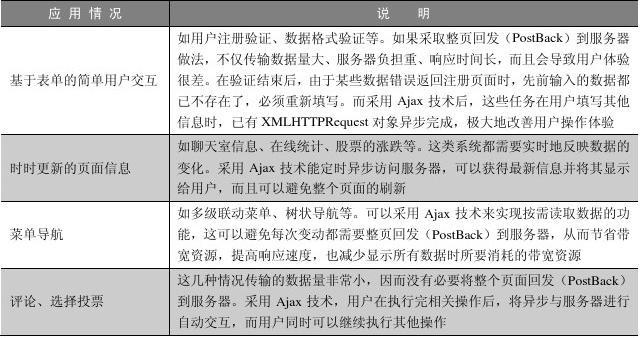
     
 从具体的实现上来说，现代的浏览器通过一个JavaScript的内置对象，也就是出现频率很高的XMLHttpRequest对象来实现这种效果。如果我们要获取系统的时间，就要使用Date对象，如果要进行数据交互，就得使用XHR对象，就是这样。

     
 下面两幅图表示有了Ajax之后的请求流程：

[](http://photo.blog.sina.com.cn/showpic.html#blogid=12ce411a90102w0ef&url=http://album.sina.com.cn/pic/005vDmpHzy6W4OOTxLU46)

[](http://photo.blog.sina.com.cn/showpic.html#blogid=12ce411a90102w0ef&url=http://album.sina.com.cn/pic/005vDmpHzy6W4OPjbHU26)

     
 这是Ajax的典型应用场景：

[](http://photo.blog.sina.com.cn/showpic.html#blogid=12ce411a90102w0ef&url=http://album.sina.com.cn/pic/005vDmpHzy6W4OQNX5Zd7)

     
 2.形象说明用不用Ajax的差别。

     
 传统的方式对用户来说是同步的，也就是浏览器请求数据，服务器返回数据，在请求到返回的过程中，页面上什么内容也没有，这样的用户体验是很不好的，如果请求到返回的事件很长，那么电脑屏幕将会是长时间的空白。用了Ajax之后，内置的XMLHttpRequest对象会承担起来发送请求和接受数据的责任，请求发送之后我们仍然可以对网页进行查看以及操作，等到服务器返回了请求的内容，再把它们添加到页面上即可。这样就没有了中间等待的间隔。

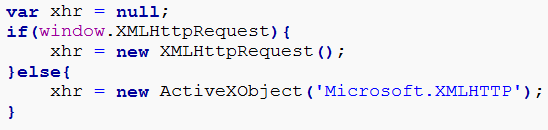
     
 举个生活中的例子，比如说我正在宿舍看电影，这时候突然想喝一瓶饮料。传统的方式是我迅速跑到超市买了一瓶再跑回宿舍，在这中间的过程就不能看电影了。这就是所谓的同步方式带来的问题。但是用了类似Ajax的方法之后，我打电话叫了外卖，会有别人把饮料给我送到宿舍中，在这个过程中我仍然可以看电影，等饮料来了再喝就好了。

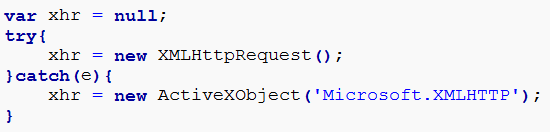
     
 3.通过上面的描述，我们可以看出来，从用户的角度来看，Ajax就是一种效果，就是不刷新页面但可以获取数据。其实在没有XMLHttpRequest对象，我们就能实现这样的效果。比如利用下面的方式：

     
 (1).在不需要提交任何信息的时候，前端可以通过a标签的href或者动态创建img标签，或者加载外部css和script等方法，让页面请求到某个具体的后台页面。而在服务器端，可以通过http协议的204状态码来实现不让浏览器跳转的效果。

     
 (2).如果需要提交信息，又不想通过地址栏传参的get方式，可以在本页面加入一个iframe，设置表单target的属性值等于iframe的name属性值。这样就能利用post方式提交而不让页面跳转。当然，我们可以对iframe的样式进行设置，让它在视觉上看不出来。然后在服务器处理之后，根据处理结果，用后台语言通过script对标签操作页面的样式，给用户返回结果。

     
 4.兼容获取XMLHttpRequest对象的方法，两种方式都可以。一种是判断语句，一种是用异常处理，能够捕获错误的try-catch语句。

[](http://photo.blog.sina.com.cn/showpic.html#blogid=12ce411a90102w0ef&url=http://album.sina.com.cn/pic/005vDmpHzy6Wb5Bj0Arab)

[](http://photo.blog.sina.com.cn/showpic.html#blogid=12ce411a90102w0ef&url=http://album.sina.com.cn/pic/005vDmpHzy6Wb5MpL2Oc2)

     
 5.XMLHttpRequest对象的属性。

     
 (1)responseText。从HTTP协议的角度来看，它代表的是服务器响应的主体信息，body信息。从我们用户的角度看，就是在浏览器的页面上能看到的信息，这个属性的值肯定是字符串类型的。

     
 (2)status是服务器返回的状态码。就是上面提到的http协议的状态码。

     
 (3)statusText，状态码对应的文字描述，也是http协议中的内容。

     
 (4)onreadystatechange，一个事件句柄，用来绑定一个函数。具体来说，它绑定的是当XMLHttpRequest对象的状态发生变化的时候激发的函数。

     
 (5)readyState，这是XMLHttpRequest对象自身的状态码，有以下几种：

0（未初始化）还没有调用send()方法,此阶段确认XMLHttpRequest对象是否创建，值为0表示对象已经存在。

1（载入），此阶段对XMLHttpRequest对象进行初始化，已经调用  
send()方法，值为1表示正在向服务端发送请求。

2表示载入完成，（send()方法执行完毕），已经收到全部的响应内容。但是此时收到的并不是我们能直接看懂的内容，所以还不能在客户端调用。

3正在解析响应内容。

4响应内容解析完成。这里要特别注意，完成并不一定代表成功得到了想要得到的内容。

     
(6).responseXML，对于大量的格式化文档，可以用XML来传输或交换，由后台程序把数据封装在XML文档中，JS接收XML对象并解析其中的内容。

     
 6.XMLHttpRequest对象的方法。

     
 (1).open('请求方式','url',同步/异步)，默认为true，表示异步。同步和异步听起来有些高大上，其实可以这样形容，前面的没执行完，后面的就不能执行，这就是同步，它们排着队，前面的会阻塞后面的。而异步就是前面的没做完也不影响后面的执行，等到前面的执行完了把结果动态的拿过来就好了，通过及时“插队”的形式实现了异步。

     
 (2).send(null,参数)，参数的格式是键值对&键值对。

比如这样：obj.send('username=' un '&email=' eml);

     
 (3).getResponseHeader()：获取响应的某个头信息，括号中写入到底要获得哪一个头信息，比如('Content-Type')用来获取返回的主体内容的长度。

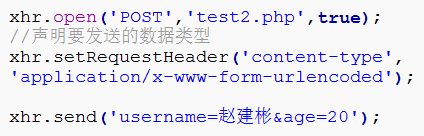
     
 (4).getAllResponseHeaders()：获取响应的所有头信息。

     
 (5).Abort,忽略，不要再进行下去了，到此为止。

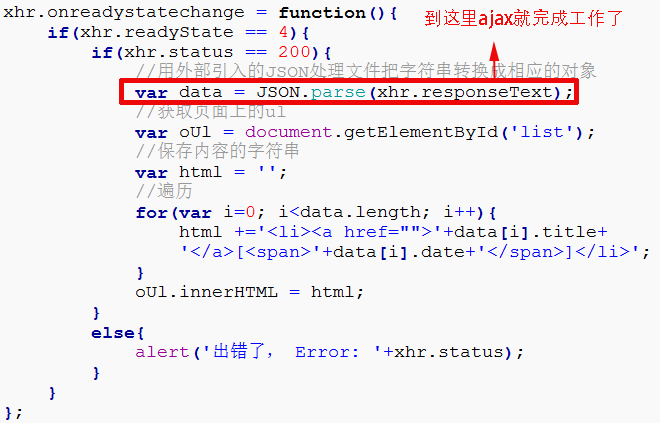
     
 (6).setRequestHeader(key,value),这个用来设置请求头信息。

     
 12.get方式提交数据，直接把要发送的内容跟在请求地址的后面即可，同时还要注意两个问题：缓存和中文乱码。浏览器的缓存会导致后端更新的数据不能及时的反馈到前端，所以要在地址最后加上一个时间戳或者是随机数。而解决中文乱码，可以在发送的时候用encodeRUI函数对汉字进行编码，这个函数的作用就是把字符串作为URI进行编码，具体的示例如下图。[HTTP协议和Ajax基础](http://photo.blog.sina.com.cn/showpic.html#blogid=12ce411a90102w0ef&url=http://album.sina.com.cn/pic/005vDmpHzy6WcGBAYgRf5)

     
 7.post方式没有缓存问题，也没有中文乱码问题。它用来提交数据，提交数据是不会产生缓存的。把要提交的数据放到send方法的括号中作为参数传递过去。具体的拼接方式仍然不变，不过一定要注意一点，post提交数据的时候，必须在请求头信息中声明Content-Type，告诉后端你要发送的大致是什么内容，方便服务器进行相应的解析。同时，由于设置了内容类型，也就没有中文乱码问题了。下面是示例：

  [](http://photo.blog.sina.com.cn/showpic.html#blogid=12ce411a90102w0ef&url=http://album.sina.com.cn/pic/005vDmpHzy6WcHDmIgE18)  
  
   
     
 Content-Type的类型不只一种，要按照需要进行设置。

     
 8.下面是一段从后端获取新闻列表和日期并且添加到页面ul中的代码。ajax的任务到获取完返回数据就完成了，剩下的是页面的dom操作。用到的JSON方法是通过引入外部的js文件做的，因为在低版本ie中不支持原生的JSON方法。

[](http://photo.blog.sina.com.cn/showpic.html#blogid=12ce411a90102w0ef&url=http://album.sina.com.cn/pic/005vDmpHzy6WcN5H6lk1a)

     
 9.如果把上面整个过程中相应的变量用参数代替，然后在响应成功和失败的时候分别执行一个回调函数，再配合定时器定时执行相应的DOM操作，就是ajax最基本的应用了。