# Ajax简介

1. Ajax（Asynchronous JavaScript and XML）表示异步的JavaScript和XML。它并不是一门单一的技术，而是系列Web技术所形成的结合体，不管使用何种技术开发Web，都可以使用Ajax。
2. Ajax是一门局部刷新技术。通过使用Ajax可以在不刷新页面的前提下，从后台加载数据更新到前台。
3. Ajax的使用必须有服务器的存在。因为Ajax技术是在不刷新页面的前提下，从服务端得到数据。

# 实现Ajax

在JQuery中通过ajax( )、get( )、post( )可以方便的实现Ajax。

## ajax( )方法

通过ajax( )方法可以方便的实现Ajax。

### 使用语法

ajax( )方法的语法如下：

**$.ajax(JS对象) ;**

如：



当单击按钮时将发送ajax请求。所以ajax请求的触发是绑定在一个事件上的。

可以看出，ajax( )方法的参数是一个JS对象，该对象包含了各项配置及回调信息。其中该对象的属性及其属性值的取值可以如下：

|  |  |  |
| --- | --- | --- |
| 属性名 | 值类型 | 说明 |
| async | Boolean | 设置本次ajax请求是否是异步请求，默认为true。如果设置为false，表示同步请求，同步请求将锁住浏览器，用户其他操作必须等待请求完成才可以执行。（一般默认为true） |
| beforeSend | Function | 发送请求前可修改XMLHttpRequest对象的函数，如添加自定义HTTP头。XMLHttpRequest对象时唯一的参数。如果该函数返回false，可以取消本次Ajax请求。 |
| cache | Boolean | 是否设置缓冲，默认为true。当dataType属性值为script时，默认为false，为false时将不会从浏览器缓存中加载请求信息。 |
| complete | Function | 请求完成后回调函数（请求成功或失败均调用）。该函数由两个参数：XMLHttpRequest对象和一个描述成功请求类型的字符串。 |
| **contentType** | **String** | **发送信息至服务器时内容的编码类型。默认为application/x-www-form-urlencoded** |
| **data** | **Object**  **String** | **发送到服务端的数据。将自动转化为请求字符串格式，必须为key/value格式。GET请求中将附在URL后。查看processData选项说明以禁止此自动转换。如果为数据，JQuery将自动为不同值对应同一个名称。如{foo:[ “bar1”,”bar2”]}转换为’&foo=ba1r&foo=bar2’** |
| dataFilter | Function | 给Ajax返回的原始数据进行预处理的函数。提供data和type两个参数：data是Ajax返回的原始数据，type是调用$.ajax( )时提供的dataType参数。函数返回的值将由JQuery进一步处理。 |
| **dataType** | **String** | **预期本次Ajax请求返回的数据类型。如果不指定，JQuery自动根据HTTP包含的MIME信息返回responseXML或responseText，并作为回调函数参数传递，可用值：**  **xml：XML文档，可用JQuery处理**  **html：返回纯文本HTML信息，包含的script标签会在插入DOM时执行**  **script：返回纯JavaScript代码。不会自动缓存结果。除非设置了cache参数。如果请求地址是远程的（跨域请求），所有的POST请求将转换为GET请求**  **text：返回纯文本字符串** |
| error | Function | 请求失败时调用的函数。该函数包含3个参数：XMLHttpRequest对象、错误信息（可选）、捕获的错误对象。如果发生了错误，错误信息（第二个参数）处了得到null之外，还可能是timeout、error、notmodified和parseerror |
| global | Boolean | 是否触发全局Ajax事件，默认为true。设置为false将不会触发全局Ajax事件，如ajaxStart或ajaxStop可用于控制不同的Ajax事件 |
| ifModified | Boolean | 是否仅在服务端数据改变时获取新数据，默认值为false。使用HTTP包含的Last-Modified头信息进行判断 |
| jsonp | String | 在一个jsonp请求中重写回调函数的名字。这个值用来替代在”callback=?”这种GET或POST请求中URL参数里的callback部分，如{jsonp:’onJsonPLoad’}会导致将’onJsonPLoad=?”传给服务器 |
| password | String | 用于相应HTTP访问认证请求的密码 |
| processData | Boolean | 是否将发送的数据转换为对象（技术上讲并非字符串）已配合默认内容类型application/x-www-form-urlencoded。默认为true，如果要发送DOM树信息或其他不希望转换的信息，请设置为false |
| scriptCharset | String | 只有当dataType属性为jsonp或script,并且type属性为GET时才会用于强制修改charset。通常在本地和远程的内容编码不同时使用。 |
| **success** | **Function** | **请求成功后执行的回调函数。函数的参数是由服务器返回、并根据dataType属性值进行处理后的数据，描述状态的字符串。** |
| timeout | Number | 设置请求超时时间（毫秒）。此设置将覆盖全局。 |
| type | String | 设置请求方式，如POST或GET，默认为GET。其他HTTP请求方式，如PUT和DELETE也可以使用，但仅部分浏览器支持。 |
| **url** | **String** | **发送请求的地址，默认为当前地址。** |
| username | String | 用于响应HTTP访问认证请求的用户名 |
| xhr | Function | 需要返回一个XMLHttpRequest对象，默认在IE下是ActionObject，其他情况下是XMLHttpRequest。用于重写或者提供一个增强的XMLHttpRequest对象 |

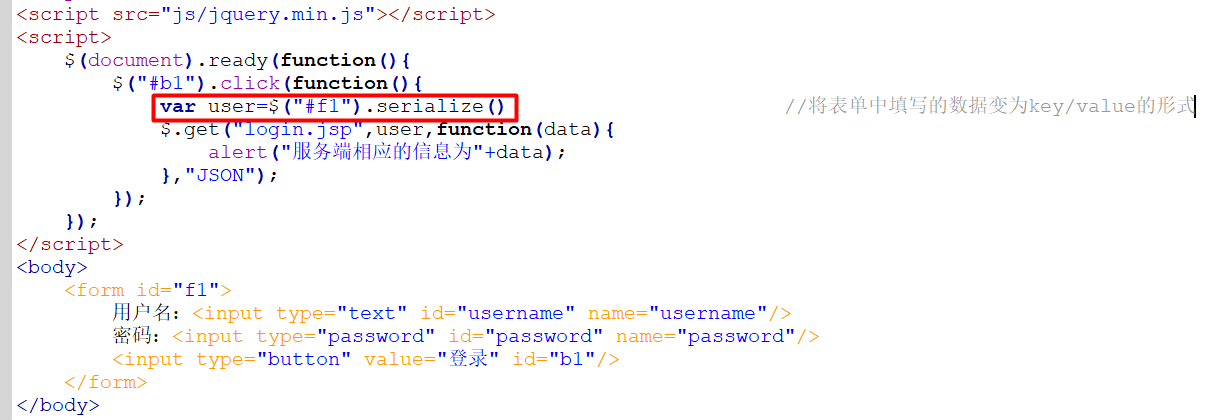
在这些常用属性中，常用的属性有：url、type、data、contentType、success。

### 序列化字符串

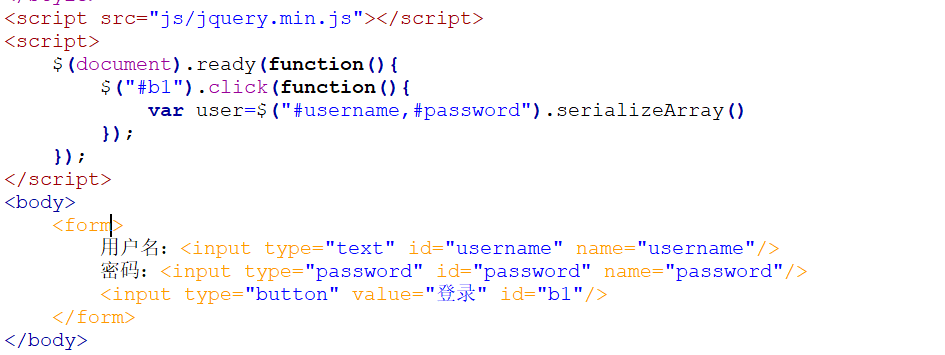
在使用Ajax时往往需要发送数据给后台，但发送ajax请求时，数据格式必须是key/value或者JSON的形式。如果这些数据由用户自己拼凑，则非常繁琐。为此JQuery中提供了serialize( )方法和serializeArray( )方法，可以轻松的将数据转换为key/value的格式或JSON格式。

如下：

转换为key/value格式。如下：



转换为JSON的格式。如下：



注意：转换为JSON格式时不能对表单进行操作，而是对表单中的表单元素进行操作。

### 设置Ajax选项

1）如果一个页面中存在多个ajax请求，则每一次编写一个ajax请求都需要设置很多参数，从而使得操作繁琐。为此，JQuery中定义了ajaxSetup( )方法，该方法可以将每个ajax请求中相同的参数设置为通用选项。从而减轻频繁设置选项的繁琐。

2）ajaxSetup( )方法的参数与ajax( )的参数相同，是一个JS对象，该对象的属性列表与ajax()中参数相同，通过ajaxSetup( )可以实现全局共享，从而在具体的ajax( )中只需要设置个性化参数即可。



### 请求状态跟踪

JQuery在XMLHttpRequest对象定义的readyState属性基础上，对异步交互中服务器响应的状态进行了封装，提供了6个响应事件，以便于对整个ajax请求响应过程的追踪。

|  |  |  |
| --- | --- | --- |
| 事件 | 说明 | 优先级 |
| ajaxStart( ) | Ajax请求开始时进行响应 | 1 |
| ajaxSend( ) | Ajax请求发送前进行响应 | 2 |
| ajaxComplete( ) | Ajax请求完成时（不管成败）进行响应 | 4 |
| ajaxSuccess( ) | Ajax请求成功时进行响应 | 3 |
| ajaxStop( ) | Ajax请求结束时进行响应 | 5 |
| ajaxError( ) | Ajax请求发生错误时进行响应 | 3 |

例如：



### Ajax载入文件

通过ajax( )能够在不刷新页面的基础上将某一个网页的内容载入到该页的某个标签中。类似于JSP中的包含指令。

## get( )方法

get( )方法用于通过GET提交方式发出ajax请求。语法如下

**$.get( url , [data] , [callback] , [type] )**

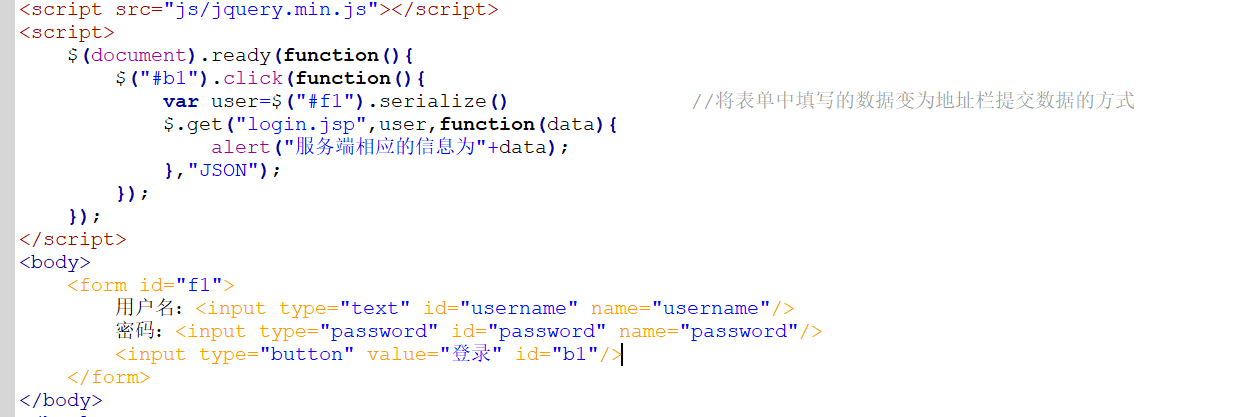
url表示要请求的地址，[

data为向请求地址传递的参数，为”a=1&b=1”的格式

callback表示请求成功之后调用的函数，该函数可以有一个参数，参数值为服务端响应的信息；（可选）

type表示服务端响应信息返回的内容格式，如XML、HTML、Script、JSON和Text，或者\_default

实例如下：



## post( )方法

post( )方法用于通过POST提交方式来发出ajax请求。语法如下：

**$.post( url , [data] , [callback] , [type] )**

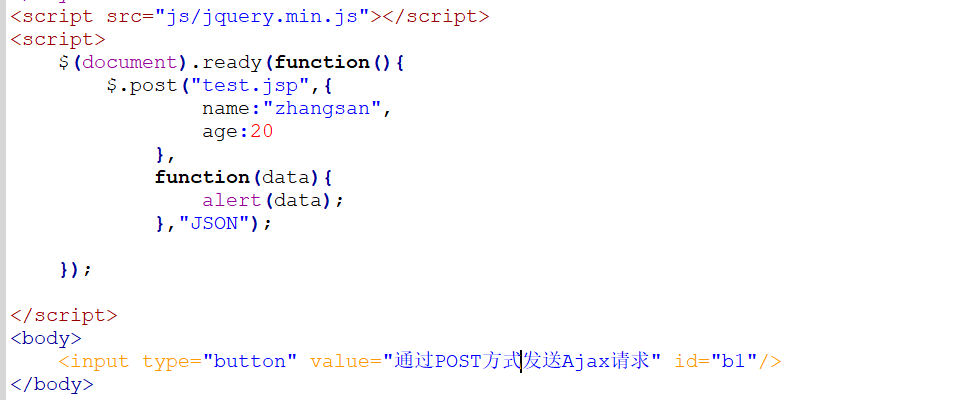
url表示要请求的URL地址；

data为向请求地址传递的数据，格式为JS对象结构的键/值列表；（可选）

callback表示请求成功之后调用的函数，该函数可以有一个参数，参数值为服务端响应的信息；（可选）

type表示服务端响应信息返回的内容格式，如XML、HTML、Script、JSON和Text，或者\_default

示例如下：



# GET和POST请求方式的对比

1. 发送的数据量

GET只能发送有限数量的数据，因为数据是在URL中发送的。

POST可以发送大量的数据，因为数据是从正文发送的。

1. 安全性

GET发送的数据不安全。因为是从URL中发送的。

POST发送数据是安全的，因为数据未在URL中公开。

1. 使用用途

GET一般用于获取数据

POST一般用于更新数据

1. 保存书签

GET查询的结果可以保存为书签

POST查询结果无法保存为书签