AJAX 教程

* [AJAX 教程](http://www.w3school.com.cn/ajax/index.asp)
* [AJAX 简介](http://www.w3school.com.cn/ajax/ajax_intro.asp)

**AJAX**

AJAX = Asynchronous JavaScript and XML（异步的 JavaScript 和 XML）。

AJAX 不是新的编程语言，而是一种使用现有标准的新方法。

AJAX 是与服务器交换数据并更新部分网页的艺术，在不重新加载整个页面的情况下。

[开始学习 AJAX](http://www.w3school.com.cn/ajax/ajax_intro.asp) ！

AJAX 简介

* [AJAX 教程](http://www.w3school.com.cn/ajax/index.asp)
* [AJAX 实例](http://www.w3school.com.cn/ajax/ajax_example.asp)

**AJAX 是一种在无需重新加载整个网页的情况下，能够更新部分网页的技术。**

**您应当具备的基础知识**

在继续学习之前，您需要对下面的知识有基本的了解：

* HTML / XHTML
* CSS
* JavaScript / DOM

如果您希望首先学习这些项目，请在我们的[首页](http://www.w3school.com.cn/index.html)访问这些教程。

**什么是 AJAX ？**

AJAX = 异步 JavaScript 和 XML。

AJAX 是一种用于创建快速动态网页的技术。

通过在后台与服务器进行少量数据交换，AJAX 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。

传统的网页（不使用 AJAX）如果需要更新内容，必需重载整个网页面。

有很多使用 AJAX 的应用程序案例：新浪微博、Google 地图、开心网等等。

**Google Suggest**

在 2005 年，Google 通过其 Google Suggest 使 AJAX 变得流行起来。

Google Suggest 使用 AJAX 创造出动态性极强的 web 界面：当您在谷歌的搜索框输入关键字时，JavaScript 会把这些字符发送到服务器，然后服务器会返回一个搜索建议的列表。

**今天就开始使用 AJAX**

AJAX 基于已有的标准。这些标准已被大多数开发者使用多年。

请阅读下一章，看看 AJAX 是如何工作的！

AJAX 实例

* [AJAX 简介](http://www.w3school.com.cn/ajax/ajax_intro.asp)
* [XHR 创建对象](http://www.w3school.com.cn/ajax/ajax_xmlhttprequest_create.asp)

**为了帮助您理解 AJAX 的工作原理，我们创建了一个小型的 AJAX 应用程序。**

**实例**

AJAX is not a programming language.

It is just a technique for creating better and more interactive web applications.

通过 AJAX 改变内容

**AJAX 实例解释**

上面的 AJAX 应用程序包含一个 div 和一个按钮。

div 部分用于显示来自服务器的信息。当按钮被点击时，它负责调用名为 loadXMLDoc() 的函数：

<html>

<body>

<div id="myDiv"><h3>Let AJAX change this text</h3></div>

<button type="button" onclick="loadXMLDoc()">Change Content</button>

</body>

</html>

接下来，在页面的 head 部分添加一个 <script> 标签。该标签中包含了这个 loadXMLDoc() 函数：

<head>

<script type="text/javascript">

function loadXMLDoc()

{

.... AJAX script goes here ...

}

</script>

</head>

下面的章节会为您讲解 AJAX 的工作原理。

AJAX - 创建 XMLHttpRequest 对象

* [AJAX 实例](http://www.w3school.com.cn/ajax/ajax_example.asp)
* [XHR 请求](http://www.w3school.com.cn/ajax/ajax_xmlhttprequest_send.asp)

**XMLHttpRequest 是 AJAX 的基础。**

**XMLHttpRequest 对象**

所有现代浏览器均支持 XMLHttpRequest 对象（IE5 和 IE6 使用 ActiveXObject）。

XMLHttpRequest 用于在后台与服务器交换数据。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。

**创建 XMLHttpRequest 对象**

所有现代浏览器（IE7+、Firefox、Chrome、Safari 以及 Opera）均内建 XMLHttpRequest 对象。

**创建 XMLHttpRequest 对象的语法：**

variable=new XMLHttpRequest();

**老版本的 Internet Explorer （IE5 和 IE6）使用 ActiveX 对象：**

variable=new ActiveXObject("Microsoft.XMLHTTP");

为了应对所有的现代浏览器，包括 IE5 和 IE6，请检查浏览器是否支持 XMLHttpRequest 对象。如果支持，则创建 XMLHttpRequest 对象。如果不支持，则创建 ActiveXObject ：

var xmlhttp;

if (window.XMLHttpRequest)

{// code for IE7+, Firefox, Chrome, Opera, Safari

xmlhttp=new XMLHttpRequest();

}

else

{// code for IE6, IE5

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

}

在下一章中，您将学习发送服务器请求的知识。

AJAX - 向服务器发送请求

* [XHR 创建对象](http://www.w3school.com.cn/ajax/ajax_xmlhttprequest_create.asp)
* [XHR 响应](http://www.w3school.com.cn/ajax/ajax_xmlhttprequest_response.asp)

**XMLHttpRequest 对象用于和服务器交换数据。**

**向服务器发送请求**

如需将请求发送到服务器，我们使用 XMLHttpRequest 对象的 open() 和 send() 方法：

xmlhttp.open("GET","test1.txt",true);

xmlhttp.send();

|  |  |
| --- | --- |
| **方法** | **描述** |
| open(*method*,*url*,*async*) | 规定请求的类型、URL 以及是否异步处理请求。   * *method*：请求的类型；GET 或 POST * *url*：文件在服务器上的位置 * *async*：true（异步）或 false（同步） |
| send(*string*) | 将请求发送到服务器。   * *string*：仅用于 POST 请求 |

**GET 还是 POST？**

与 POST 相比，GET 更简单也更快，并且在大部分情况下都能用。

然而，在以下情况中，请使用 POST 请求：

* 无法使用缓存文件（更新服务器上的文件或数据库）
* 向服务器发送大量数据（POST 没有数据量限制）
* 发送包含未知字符的用户输入时，POST 比 GET 更稳定也更可靠

**GET 请求**

一个简单的 GET 请求：

xmlhttp.open("GET","demo\_get.asp",true);

xmlhttp.send();

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=ajax_get)

在上面的例子中，您可能得到的是缓存的结果。

为了避免这种情况，请向 URL 添加一个唯一的 ID：

xmlhttp.open("GET","demo\_get.asp?t=" + Math.random(),true);

xmlhttp.send();

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=ajax_get_unique)

如果您希望通过 GET 方法发送信息，请向 URL 添加信息：

xmlhttp.open("GET","demo\_get2.asp?fname=Bill&lname=Gates",true);

xmlhttp.send();

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=ajax_get2)

**POST 请求**

一个简单 POST 请求：

xmlhttp.open("POST","demo\_post.asp",true);

xmlhttp.send();

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=ajax_post)

如果需要像 HTML 表单那样 POST 数据，请使用 setRequestHeader() 来添加 HTTP 头。然后在 send() 方法中规定您希望发送的数据：

xmlhttp.open("POST","ajax\_test.asp",true);

xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");

xmlhttp.send("fname=Bill&lname=Gates");

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=ajax_post2)

|  |  |
| --- | --- |
| **方法** | **描述** |
| setRequestHeader(*header*,*value*) | 向请求添加 HTTP 头。   * *header*: 规定头的名称 * *value*: 规定头的值 |

**url - 服务器上的文件**

open() 方法的 *url* 参数是服务器上文件的地址：

xmlhttp.open("GET","ajax\_test.asp",true);

该文件可以是任何类型的文件，比如 .txt 和 .xml，或者服务器脚本文件，比如 .asp 和 .php （在传回响应之前，能够在服务器上执行任务）。

**异步 - True 或 False？**

AJAX 指的是异步 JavaScript 和 XML（Asynchronous JavaScript and XML）。

XMLHttpRequest 对象如果要用于 AJAX 的话，其 open() 方法的 async 参数必须设置为 true：

xmlhttp.open("GET","ajax\_test.asp",true);

对于 web 开发人员来说，发送异步请求是一个巨大的进步。很多在服务器执行的任务都相当费时。AJAX 出现之前，这可能会引起应用程序挂起或停止。

通过 AJAX，JavaScript 无需等待服务器的响应，而是：

* 在等待服务器响应时执行其他脚本
* 当响应就绪后对响应进行处理

**Async = true**

当使用 async=true 时，请规定在响应处于 onreadystatechange 事件中的就绪状态时执行的函数：

xmlhttp.onreadystatechange=function()

{

if (xmlhttp.readyState==4 && xmlhttp.status==200)

{

document.getElementById("myDiv").innerHTML=xmlhttp.responseText;

}

}

xmlhttp.open("GET","test1.txt",true);

xmlhttp.send();

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=ajax_async_true)

您将在稍后的章节学习更多有关 onreadystatechange 的内容。

**Async = false**

如需使用 async=false，请将 open() 方法中的第三个参数改为 false：

xmlhttp.open("GET","test1.txt",false);

我们不推荐使用 async=false，但是对于一些小型的请求，也是可以的。

请记住，JavaScript 会等到服务器响应就绪才继续执行。如果服务器繁忙或缓慢，应用程序会挂起或停止。

**注释：**当您使用 async=false 时，请不要编写 onreadystatechange 函数 - 把代码放到 send() 语句后面即可：

xmlhttp.open("GET","test1.txt",false);

xmlhttp.send();

document.getElementById("myDiv").innerHTML=xmlhttp.responseText;

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=ajax_async_false)

AJAX - 服务器响应

* [XHR 请求](http://www.w3school.com.cn/ajax/ajax_xmlhttprequest_send.asp)
* [XHR readyState](http://www.w3school.com.cn/ajax/ajax_xmlhttprequest_onreadystatechange.asp)

**服务器响应**

如需获得来自服务器的响应，请使用 XMLHttpRequest 对象的 responseText 或 responseXML 属性。

|  |  |
| --- | --- |
| **属性** | **描述** |
| responseText | 获得字符串形式的响应数据。 |
| responseXML | 获得 XML 形式的响应数据。 |

**responseText 属性**

如果来自服务器的响应并非 XML，请使用 responseText 属性。

responseText 属性返回字符串形式的响应，因此您可以这样使用：

document.getElementById("myDiv").innerHTML=xmlhttp.responseText;

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=ajax_async_true)

**responseXML 属性**

如果来自服务器的响应是 XML，而且需要作为 XML 对象进行解析，请使用 responseXML 属性：

请求 [books.xml](http://www.w3school.com.cn/example/xmle/books.xml) 文件，并解析响应：

xmlDoc=xmlhttp.responseXML;

txt="";

x=xmlDoc.getElementsByTagName("ARTIST");

for (i=0;i<x.length;i++)

{

txt=txt + x[i].childNodes[0].nodeValue + "<br />";

}

document.getElementById("myDiv").innerHTML=txt;

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=ajax_responsexml)

AJAX - onreadystatechange 事件

* [XHR 响应](http://www.w3school.com.cn/ajax/ajax_xmlhttprequest_response.asp)
* [AJAX ASP/PHP](http://www.w3school.com.cn/ajax/ajax_asp_php.asp)

**onreadystatechange 事件**

当请求被发送到服务器时，我们需要执行一些基于响应的任务。

每当 readyState 改变时，就会触发 onreadystatechange 事件。

readyState 属性存有 XMLHttpRequest 的状态信息。

下面是 XMLHttpRequest 对象的三个重要的属性：

|  |  |
| --- | --- |
| **属性** | **描述** |
| onreadystatechange | 存储函数（或函数名），每当 readyState 属性改变时，就会调用该函数。 |
| readyState | 存有 XMLHttpRequest 的状态。从 0 到 4 发生变化。   * 0: 请求未初始化 * 1: 服务器连接已建立 * 2: 请求已接收 * 3: 请求处理中 * 4: 请求已完成，且响应已就绪 |
| status | 200: "OK"  404: 未找到页面 |

在 onreadystatechange 事件中，我们规定当服务器响应已做好被处理的准备时所执行的任务。

当 readyState 等于 4 且状态为 200 时，表示响应已就绪：

xmlhttp.onreadystatechange=function()

{

if (xmlhttp.readyState==4 && xmlhttp.status==200)

{

document.getElementById("myDiv").innerHTML=xmlhttp.responseText;

}

}

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=ajax_async_true)

**注释：**onreadystatechange 事件被触发 5 次（0 - 4），对应着 readyState 的每个变化。

**使用 Callback 函数**

callback 函数是一种以参数形式传递给另一个函数的函数。

如果您的网站上存在多个 AJAX 任务，那么您应该为创建 XMLHttpRequest 对象编写一个**标准**的函数，并为每个 AJAX 任务调用该函数。

该函数调用应该包含 URL 以及发生 onreadystatechange 事件时执行的任务（每次调用可能不尽相同）：

function myFunction()

{

loadXMLDoc("ajax\_info.txt",function()

{

if (xmlhttp.readyState==4 && xmlhttp.status==200)

{

document.getElementById("myDiv").innerHTML=xmlhttp.responseText;

}

});

}

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=ajax_callback)