1. AJAX - 创建 XMLHttpRequest 对象
2. 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。

1. 创建XMLHttpRequest对象 – 具体步骤 – 一般在head

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

具体代码：

var xmlhttp;

if (window.XMLHttpRequest)

{

// IE7+, Firefox, Chrome, Opera, Safari 浏览器执行代码

xmlhttp=new XMLHttpRequest();

}

else

{

// IE6, IE5 浏览器执行代码

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

}

1. AJAX - 向服务器发送请求请求 – XMLHttpRequest对象用于和服务器交换数据
2. 向服务器发送请求

* 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**

1. GET更简单更快，POST请求适用以下情况

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

1. GET请求实例 – GET会形成缓存，可以添加参数

xmlhttp.open("GET","/try/ajax/demo\_get.php",true);

xmlhttp.send();

xmlhttp.open("GET","/try/ajax/demo\_get2.php?fname=Henry&lname=Ford",true);

xmlhttp.send();

1. POST请求

xmlhttp.open("POST","/try/ajax/demo\_post.php",true);

xmlhttp.send();

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","/try/ajax/demo\_post2.php",true);

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

xmlhttp.send("fname=Henry&lname=Ford");

方法：setRequestHeader(header,value)

向请求添加 HTTP 头。

header: 规定头的名称

value: 规定头的值

1. Async – ture 表示异步

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","/try/ajax/ajax\_info.txt",true);

xmlhttp.send();

1. Async – false 表示同步

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","/try/ajax/ajax\_info.txt",false);

xmlhttp.send();

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

1. AJAX - 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。

1. responseText 获得字符串形式的响应数据。

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;

1. responseXML 获得 XML 形式的响应数据。

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：

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;

1. AJAX - onreadystatechange 事件
2. 定义：

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。

1. 三个主要属性：

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：状态 200: "OK" 404: 未找到页面

代码：xmlhttp.**onreadystatechange**=function()

{

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

{

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

}

}

1. AJAX 数据库实例 -- AJAX 可用来与数据库进行动态通信。

* 主要利用PHP与数据库进行通信

1. AJAX XML

* 主要就是open xml文档就行了

实例代码： -- ajax属于异域请求，本地浏览器不支持跨域，所以要把文件放到tomcat工程

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<script>

function loadXMLDoc()

{

var xmlhttp;

if (window.XMLHttpRequest)

{

// IE7+, Firefox, Chrome, Opera, Safari 浏览器执行代码

xmlhttp=new XMLHttpRequest();

}

else

{

// IE6, IE5 浏览器执行代码

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

}

xmlhttp.onreadystatechange=function()

{

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

{

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

}

}

xmlhttp.open("GET","/try/ajax/ajax\_info.txt",true);

xmlhttp.send();

}

</script>

</head>

<body>

<div id="myDiv"><h2>使用 AJAX 修改该文本内容</h2></div>

<button type="button" onclick="loadXMLDoc()">修改内容</button>

</body>

</html>