

# AJAX

[http://www.w3school.com.cn/ajax/ajax\\_intro.asp](http://www.w3school.com.cn/ajax/ajax_intro.asp)

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。

```
function success(text) {
    var textarea = document.getElementById('test-response-text');
    textarea.value = text;
}

function fail(code) {
    var textarea = document.getElementById('test-response-text');
    textarea.value = 'Error code: ' + code;
}

var request = new XMLHttpRequest(); // 新建XMLHttpRequest对象

request.onreadystatechange = function () {
    // 当 readyState 改变时，就会触发 onreadystatechange 事件，函数被回调
    if (request.readyState === 4) { // 请求成功完成
        // 判断响应结果：
        if (request.status === 200) {
            // 成功，通过responseText拿到响应的文本：
            return success(request.responseText);
        } else {
            // 失败，根据响应码判断失败原因：
            return fail(request.status);
        }
    } else {
        // HTTP请求还在继续...
    }
}

// 发送请求：
request.open('GET', '/api/categories');
request.send();

alert('请求已发送，请等待响应...');
```

## XMLHttpRequest 对象

用于在后台与服务器交换数据。

## 创建 XMLHttpRequest 对象

```
var xmlhttp = new XMLHttpRequest();
```

### 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");
    }
```

### 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。

参数：

- method: 请求的类型；GET 或 POST。
- url: 服务器上文件的地址。
- async: true (异步) 或 false (同步)。要使用AJAX，必须设置为true。

**send(string)** 将请求发送到服务器。

参数：

- string: 仅用于 POST 请求。需要把 body 部分以字符串或者 FormData 对象传进去。

### GET 与 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

### onreadystatechange 事件

当 readyState 改变时，就会触发 onreadystatechange 事件。

#### readyState

- 0: 未初始化，XMLHttpRequest对象还没有完成初始化
- 1: 正在加载，XMLHttpRequest对象开始发送请求
- 2: 载入完成，XMLHttpRequest对象的请求发送完成
- 3: 正在解析，XMLHttpRequest对象开始读取服务器的响应
- 4: 完成，XMLHttpRequest对象读取服务器响应结束

#### status (HTTP状态码)

- 1xx: 信息响应类，表示接收到请求并且继续处理
- 2xx: 处理成功响应类，表示动作被成功接收、理解和接受
- 3xx: 重定向响应类，为了完成指定的动作，必须接受进一步重定向处理
- 4xx: 客户端错误，客户请求包含语法错误或者是不能正确执行
- 5xx: 服务端错误，服务器不能正确执行一个正确的请求

200 OK //客户端请求成功

301 Moved Permanently //永久移动。今后任何新的请求都应使用新的URI代替

302 Found //临时移动。只是临时被移动。客户端应继续使用原有URI

400 Bad Request //客户端请求有语法错误，不能被服务器所理解

401 Unauthorized //请求未经授权，这个状态代码必须和WWW-Authenticate报头域一起使用

403 Forbidden //服务器收到请求，但是拒绝提供服务

404 Not Found //请求资源不存在，eg: 输入了错误的URL

500 Internal Server Error //服务器发生不可预期的错误

501 Not Implemented //服务器不支持请求的功能，无法完成请求

503 Server Unavailable //服务器当前不能处理客户端的请求，一段时间后可能恢复正常