# [Ajax学习笔记](http://www.cnblogs.com/lzhlearn/p/5788903.html) ——汪如祥

**1-1 简介**

1.AJAX的全称—Asynchronous Javascript and XML异步的Javascript和XML （读法：阿贾克斯 --  A杰克s）

2.AjAX不是编程语言，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

3.传统的网页，要想更新内容或提交一个表单，就要重新载入页面；使用AjAX技术的页面，通过后台跟服务器进行少量的数据交换，网页就可以实现异步局部更新；

**1-2 Ajax三步骤**

1、运用HTML和CSS实现页面，表达信息；

2、运用XMLHttpRequest和web服务器进行数据的异步交换；

3、运用JavaScript操作DOM，实现动态局部刷新；

**2-1 同步和异步**

1.同步：用户填写所有信息后，提交给服务器，等待服务器的回应（检验数据），是一次性的。信息错误又要重新填写！

2.异步：当用户填写完一条信息后，该信息会自动向服务器提交，然后服务器响应客户端，在此过程中，用户依然在填写表格的信息，即向服务器请求多次，节省了用户的时间，提高了用户的体验。

3.XMLhttpRequest对象来实现这一功能，也需要javascript来操作DOM实现局部的信息更新。不需要重新加载整个页面

**2-2 创建对象**

XMLhttprequest对象实现异步，由于IE6以前的不支持该对象，故一下代码进行兼容

var request;

if(window.XMLHttpRequest){

request = new XMLHttpRequest();//IE7+Firefox,Chrome,Opera,Safari...

}else{

request = new ActiveXObject("Microsoft.XMLHTTP");//IE6,IE5

}

**2-3 http请求**

1.HTTP是计算机通过网络进行通信的规则，使浏览器请求信息和服务

2.HTTP是一种无状态的协议(不建立持久的连接，服务端不保留连接的相关信息,浏览器发出请求和服务器返回响应是一个没有记忆的过程)

3.一个完整的【HTTP请求】过程有7个步骤：

1>建立TCP连接

2>Web浏览器向Web服务器发送请求命令

3>Web浏览器发送请求头信息

4>Web服务器应答

5>Web服务器发送应答头信息

6>Web服务器向浏览器发送数据

7>Web服务器关闭TCP连接

**GET:**一般用于信息的获取，使用URL传递参数，对发送信息的数量也有限制，一般在2000字符！默认方式，一般用于查询、获取操作,不是很安全，任何人可见，信息都显示在URL中

**POST：**一般用于修改服务器上的资源，对所发送的数量无限制。一般用于发送表单数据，新建，修改，删除等操作，要安全一些，不在URL中显示，对其他人不显示。

1.HTTP请求的方法或动作，post或get

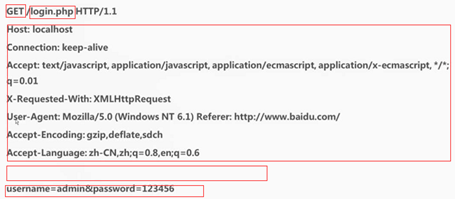
2.正在请求的URL

3.请求头，包含一些客户端环境信息，身份验证信息等

4.请求体(请求正文)，包含要发送的一些字符串信息,表单信息等等

//请求头和请求体之间有一个空行，表面请求头已经结束

如下例子



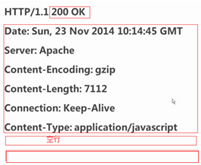
**【HTTP响应】一般由3部分组成：**

①一个数字或文字组成的状态码，用来显示请求是成功还是失败

②响应头，和请求头一样包含许多有用信息，如服务器类型、日期时间、内容类型和长度等

③响应体，即响应正文//响应头和响应体之间有空行

下例中没有正文



**【HTTP状态码】**

1XX：信息类，表示收到Web浏览器请求，正在进一步处理中

2XX:成功，表示用户请求被正确接收

3XX:重定向，表示请求没有成功，客户必须采取进一步动作

4XX:客户端错误，表示客户端提交的请求有错误，例如：404 NOT Found,意味着请求中所引用的文档不存在

5XX:服务器错误，表示服务器不能完成对请求的处理，如：500

**2-4 xmlhttprequest发送请求**

XMLHttpRequest发送请求：

两个方法

open(method,url,async)

method：规定HTTP发送请求的方式是get还是post,不区分大小写，一般来说用大写

url：请求地址(相对地址或绝对地址)

async:同步/异步(false/true)，默认是异步也就是true，可以不用填写

send(string):发送到服务器（该参数可以填或者不填-----get方法不填或填null，post:一般要填）

request.open("POST","create.php",true);

request.setRequestHeader("Content-type","application/x-www-form-urlencoded ")//设置HTTP头信息--一定要写在open()和send()之间

request.send("name=xxxx&set=xxx");

# 

# **2-5 Xmlhttprequest如何取得响应**

通过readystate属性知道服务器是否响应，只要4完成即可

readyState属性的变化代表服务器响应的变化

0：请求未初始化，open还没有调用

1：服务器连接已建立，open已经调用了

2：请求已接收，也就是接收到头信息了

3：请求处理中，也就是接收到了响应主体

4：请求已完成，且响应已就绪，也就是响应完成了

[IMG_258](http://www.cnblogs.com/lzhlearn/p/javascript:void(0);)

var request = new XMLHttpRequest() //建立XHR对象

request.open("GET","get.php",true); //用get方法异步打开get.php

request.send(); //发送请求头信息

request.onreadystatechange=function(){  
//readyState===4响应完成；status===200，请求成功

if(request.readState===4&&request.status===200){

//做一些事情 request.responseText;

}

}

[IMG_259](http://www.cnblogs.com/lzhlearn/p/javascript:void(0);)

通过onreadystatechange事件 ，对readyState属性进行监听即对服务器的响应进行监听，

**建立异步请求的过程4个步骤：**

a:new一个XHR对象

b:调用open方法

c:send一些数据

d:对过程进行监听，来知道服务器是不是正确地做出了响应，接着可以做一些事情

（监听readyState,响应成功可以做一些事情，比如获取服务器响应的内容在页面上做一些呈现）

**常用方法：**

XMLHTTPRequest取得响应

responseText：获得字符串形式的响应数据

responseXML：获得XML形式的响应数据

status和statusText：以数字和文本形式返回HTTP代码

getAllResponseHeader()：获取所有响应表头

getResponseHeader()：获取响应中的某个字段的值