**Ajax：Asynchronous（异步） JavaScript and XML,一种用于创建动态网页的技术**

(属于前端，js可以实现)

**在不重新加载整个页面的情况下，与服务器交换数据并更新部分网页的技术**

**XMLHttpRequest**

**是一个 JavaScript对象**

**是处理所有服务器通信的对象**

**XMLHttpRequest 惟一的目的是让您发送请求和接收响应**

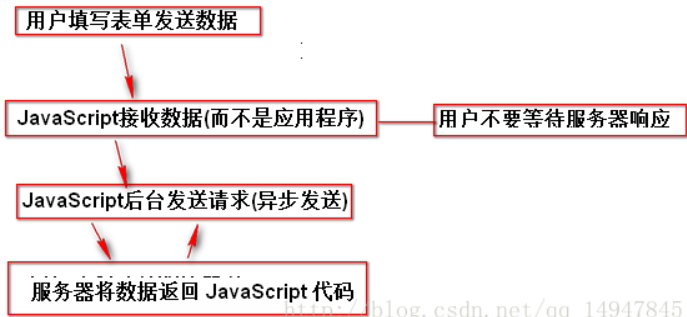
**其他一切都是 JavaScript、CSS 或页面中其他代码的工作**

通过 XMLHttpRequest 对象与服务器进行对话的是JavaScript 技术（其它则是应用程序）

与服务器上的Web 应用程序打交道的是 JavaScript 技术，而不是应用程序的HTML表单

Ajax基本上就是把JavaScript 技术和 XMLHttpRequest对象放在Web表单和服务器之间

if(window.XMLHttpRequest){  
  request=new XMLHttpRequest(); //除IE浏览器  
}else{  
  request=new ActiveXObject("Microsoft.XMLHTTP");

****

**基本流程**

**1.从 Web 表单中获取需要的数据  
2、建立要连接的 URL。**

var url = "/scripts/getZipCode.php?c~~~  
**3、打开到服务器的连接。**

xmlHttp.open("GET", url, true);//true表明异步连接，false等待服务器响应 **4、设置服务器在完成后要运行的函数。**

xmlHttp.onreadystatechange =函数名;

onreadystatechange每当 HTTP 就绪状态改变时它都会调用该方法

通过http状态码判断运行是否出错

xmlhttp.onreadystatechange=function()

{

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

{

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

}

}  
**5、发送请求**

xmlHttp.send(null);//不需要发送数据则null

可以使用 send() 发送数据，但也能通过 URL 本身发送数据

**Ajax 应用程序五种就绪状态：**

**0：请求没有建立**（在调用 open() 之前） **1：请求已经建立但还没有发出**（调用 send() 之前）。  **2：请求已经发出正在处理之中（**这里通常可以从响应得到内容头部） **3：请求已经处理**，响应中通常有部分数据可用，但是服务器还没有完成响应 **4：响应已完成**，可以访问服务器响应并使用它

**返回的数据保存在 XMLHttpRequest 对象的 responseText 属性中**

不放到方法或函数体中的 JavaScript 代码称为静态 JavaScript（直接写在script中）

其中代码在页面显示给用户之前运行

escape() 方法，它用于转义不能用明文正确发送的任何字符。比如，电话号码中的空格将被转换成字符 %20，从而能够在 URL 中传递这些字符

**DOM节点**

**文档节点表示整个 HTML 文档（页面本身）**

**元素节点表示 HTML 元素，如 a 或 img。**

**属性节点表示 HTML 元素的属性，如 href（a 元素）或 src（img 元素）。**

**文本节点表示 HTML 文档中的文本，如 “Click on the link below for a complete set list”。这是出现在 p、a 或 h2 这些元素中的文字**