Ajax 基础

Ajax：XMLHttprequest 对象，javascript，XML（可扩展的标记语言），CSS（层叠样式表）,DOM(文档对象模型),技术的整合
在 Ajax 的技术中，最核心的技术就是 XMLHttpRequest 他还 iyige 具有应用程序接口的 javascript 对象，能使用超文本语言传输协议（HTTP）连接一个服务器（微软 1999 年 IE5.0 推出）

Ajax 的优点：

（1） 减轻服务器的负担，Ajax 的原则是“按需求获取数据”最大程度上面减少多于请求和响应对服务器造成的负担。

（2） 可以把一部分以前服务器负担的工作转到客户端，利用客户端限制的资源进行处理，减轻服务器和宽带的负担

（3） 无页面刷新，从而使用户不用再像以前一样在服务器端处理数据时，只能在白屏前焦急的等待，Ajax 使用 XMLHttprequest 对象发送请求并得到服务器的响应，在部需要重新加载整个页面的情况下，就通过 DOM 及时的将更新的内容显示出来

（4） 可以调用 XML 等外部数据，进一步促进页面和数据的分离

（5） 基于标准化的并被广泛支持的技术，不再需要下载插件或者小程序

Ajax： IE 浏览器把 XMLHttpRequest 实例化一个 ActiveX 对象

```
Var_ http_request=new ActiveXObject(“Msxml2.XMLHTTP”);
```

或者

```
Var_ http_request=new ActiveXObject(“Microsoft.XMLHTTP”);
```

对于一些非 IE 浏览器例如：Mozilla,Safari 实例化本里 Javascript 对象，

```
Var http_request=new XMLHttpRequest();
```

而用于不同的浏览器，我们只是需要判断一下用户的浏览器的类型

```
If（Windows.XMLHttpRequest）{//非 IE 浏览器
            http_request=new XMLHttpRequest();
             }else if(windows,ActiveXObject){//IE 浏览器
             Try{
             http_request=new ActiveXObject(“Msxml2.XMLHTTP”);
             }catch(e){
             Try{
             http_request=new ActiveXObject(“Microsoft.XMLHTTp”);
             }catch(e){}
}
}
```

XML 对象常用的方法

1.open(“method”,“url”[,asyncFlag[,”userName”[,”password”]]])

Method:指定请求的类型，get/set
URL:指定请求的位置，可以使用绝对地址或者相对地址，并可以传递查询字符窜
asyncFlag:（可选参数）用于制定请求的方式，同步或者异步默认为 ture(异步)
usernameL（可选参数）指定请求用户名
Password（可选参数）指定请求密码
例如：http_request.open(“GET”,“eal.jsp”，true)

2.send()方法
Send()方法用于向服务器发送请求，如果是异步，立即返回，否则等待响应
Send(conntent)　　(connent 用于指定发送的数据，可以是 DOM 对象，输入流，字符流)，没有参数的话 可以为 null

3.setRequestHeader()方法
setRrequestHeader(“label”,“value”);
请求 setRequest()方法用于为请求的 HTTP 头设置值：
SetRequestHeader(“label”,”value”);　　(label 用于指定 HTTP 头，value 用于指定的 HTTP 头设置值)　　{注意：该方法是在 open()之后才调用的}
如果发送的是 POST 请求的时候，通常设置 COnnent-Type 请求头
http_request.setRequestHeader(“Conent-type”,”application/x-www-form-urlEndoded”)

4.abort()方法
Abort()方法用于停止当异步部请求
Abort()

5.getResponseHeader()
用于字符符串形式返回指定的 HTTP 头信息
getResponseHeader(“headerLabel”)
(例如：;要获取 HTTP 头 Content-type)
getResponseHeader(“Connent-typel”)

6.getAllResponseHeaders()
用于字符符串形式返回完整的的 HTTP 头信息
getAllResponseHeaders()

XMLHttpRequest 属性

Onreadystatechange	每个状态改变的时候都会出发这个时间处理器，通常会调用一个 javascript 函数
readyState	请求的状态有 5 个只可选 0=为初始化 1=正在加载 2=一家在 3 交互中 4 完成
ResponseText	服务器的响应，表示为字符串
ResponseXML	服务器的响应，表示为 XML。这个对象可以解析为一个 DOM 对象
Status	返回服务器 HTTP 的状态码 200=“成功” 202=“请求被接受但是没有成功” 400=“错误的请求” 404=“文件未找到” 500=内部服务器错误”
StatusText	返回 HTTP 状态码对象的文本，如 OK，Not Found 未找到

发送请求和处理响应

在 Ajax 中 Ajaxf 发送请求的方式有来那两种：POSH. ,GET d 但是步骤都是一下的 4 步：

(1) 初始化 XMLHttprequest（为了提高程序的兼容性，需要创建一个跨浏览器的 XMLHttpRequest 对象，并判断 XMLHttpRequest 对象的实例是否成功，如果不成功，给予提示）

```
Http_request=false;
If（Windows.XMLHttpRequest）{//非 IE 浏览器
            http_request=new XMLHttpRequest();
             }else if(windows,ActiveXObject){//IE 浏览器
             Try{
             http_request=new ActiveXObject(“Msxml2.XMLHTTP”);
             }catch(e){
             Try{
             http_request=new ActiveXObject(“Microsoft.XMLHTTp”);
             }catch(e){
             }
}
}
If(!http_request){
Alert(“不能创建一个 XMLHttpRequest 实力对象”)
```

Retuen false;
}
(2) 为 XMLHttpRequest 对象指定一个回调函数，用于对返回的结果进行处理：
http_request.onreadystatechange=getResult();（使用要 XMLHttpRequest 对象的 onreadystatechange 属性的指定回调函数的时候，不能指定要传递的参数，如果要传递参数的话）用以下方式：
http_request.onreadystatechange=function(){getResult(param)}

(3) 创建一个服务器的连接，在创建的时候需要指定一个发送的方式，创建的时候，需要至死难过一个发送的请求方式（GET，POST），以及设置是否采用异步

例如：异步发送 GET 请求;

http_request.open('GET',url,true);

例如：异步发送 POST 请求

http_request.open('POST',url,true);

open()方法中的 url 参数可以是一个 JSP 页面。也可以是一个 Servlet 的映射位置（也就是说可以是一个 JSP 页面，也可以是一个 servlet）

(z 指定一个 url 参数的时候，最好将一个时间戳追加到该 url 的后面，这样可以放置因刘看齐的缓存结果而导致不能能实时得到最新的结果)比如指定一个 url 的参数代码：

String url="deal.jsp?nocache+"+new Date.getTime();

(4) 向服务器发送请求，利用 XMLHttpRequest 对象的 send()方法可以实现向服务器发送请求，该方法需要传递一个参数，如果是 GET，该参数为 null,如果是 POST 方法请求，可以制定发送一个参数。

向服务器发送一个 GET 请求

http_request.send(null);

向服务器发送一个 POST 请求

```
Var param="user="+form1,user.value+
          "&pwd"+"form1"+
          "&email="+form1.eamail.value+
          "&question="+form1.question.value+
          "&answer="+form1.answer.value+
          "&city="+form1.city.value;(组合参数)
http_request.send(param);
```

需要注意的是，在发送 POST 请求前，还需要设置正确的的请求头：

http_request.setRequestHeader("Content-Type","application/x-www-form-urlencodes")

添加在"http_request.send(param);"

处理服务器响应

在不同的条件下面，对于不同的请求也可能不同的响应，例如网络不通畅的情况

下面就会返回一些错误的结果，所以要根据不通的状态，采取不同的方式
XMLHttpRequest 对象提供了两个用来访问服务器响应的属性;
1:responseText 属性，返回字符窜响应
2.responseXML 属性，返回的 XML 响应

(1)处理字符窜响应:

```
Function getResult(){
If(http_request.readyState==4){//强求判断状态，4=完成
Alert(http_request.responseText);//请求成功，开始处理响应
}else{
Alert(“您所请求的页面有错误”)；//返回错误页面
}
}
}
```

如果需要将响应的结果相识到页面制定的位子，也可以现在页面的何时位置添加一个 或者<spen>标签，设置改变前的 id 属性，如：div_result,然后调用回调函数中的应用：

```
Document.getElementByID(“div_result”).innerHTML=http_request.responseText;
```

(2)处理 XML 响应

如果在服务器短需要生成一个特别复杂的响应，那么就需要应用 XML 响应，应用 XMLHttpRequest 对象的 responeseXML 属性可以生成一个 xml 文档，而且当前浏览器已经提供了很好的解析 xml 文档对象的方法。
在回调函数中遍历保存留言信息的 xml 文档，并显示在页面;

```
<script language=”javascript”>
Function getResult(){
        If(http_request.readyState==4){//强求判断状态，4=完成
            If(http_request.status==200){
                Var xmldoc=http_request.responeXML;
                Var msgs=””;
                For(i=0;i<xmdoc.getElementsByTagname(“board”).length;i++){
                Var board=xmldoc.getElementsByTagNmae(“board”),item(i);
Msgs=msgs+board.getAttribute(“name”)+”
的留言：”+board.getElementByTagName(‘msg’)[0].fristChild.data+”<br>”
}
else{
alert(“请求的页面有错误!”)
}
}
}
</script>
<div id=”msg”> <div>
```