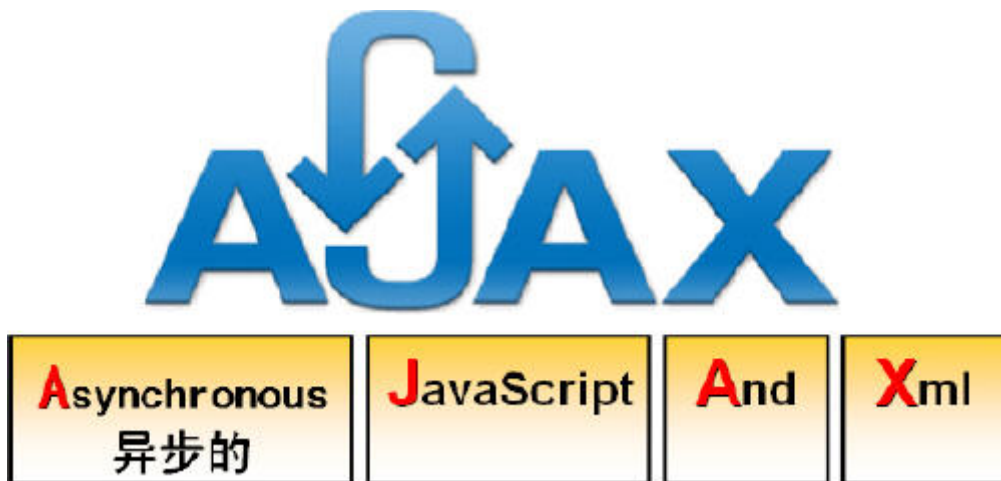


# 一、什么是Ajax



Ajax: (Asynchronous JavaScript And XML) 指异步 JavaScript 及 XML

不是一种新的编程语言，而是一种用于创建更好更快以及交互性更强的 Web 应用程序的技术，是基于JavaScript、XML、HTML、CSS新用法

Ajax: 只刷新局部页面的技术

- 1 JavaScript: 更新局部的网页
- 1 XML: 一般用于请求数据和响应数据的封装
- 1 XMLHttpRequest对象: 发送请求到服务器并获得返回结果
- 1 CSS: 美化页面样式
- 1 异步: 发送请求后不等返回结果，由回调函数处理结果

JavaScript中XMLHttpRequest对象是整个Ajax技术的核心，它提供了异步发送请求的能力

## 构造方法

不同浏览器，甚至相同浏览器的不同版本，获取该对象的方式是不同的。

```
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");
}
```

方法：

open()      “get|post” ， "url?name=tom" ， "true|false”

send()      发送请求 。可以带参数 或 null

setRequestHeader()

属性

readyState： 类型short； 只读

responseText： 类型String； 只读

responseXML： 类型Document； 只读

status： 类型short； 只读

事件处理器

onreadystatechange

## 二、常用方法

方法名	说 明
open(method,URL,async) )	建立与服务器的连接  method参数指定请求的HTTP方法,典型的值是GET或POST URL参数指请求的地址 async参数指定是否使用异步请求，其值为true或false
send(content)	发送请求 content参数指定请求的参数
setRequestHeader(header,value)	设置请求的头信息

## 三、常用属性

1 onreadystatechange： 指定回调函数

1 readyState： XMLHttpRequest的状态信息

就绪状态码	说 明
0	XMLHttpRequest对象没有完成初始化 即：刚刚创建。
1	XMLHttpRequest对象开始发送请求 调用了open方法，但还没有调用send方法。请求还没有发出

2	XMLHttpRequest对象请求发送完成 send方法已经调用，数据已经提交到服务器， 没有任何响应
3	XMLHttpRequest对象开始读取响应，还没有结束 收到了所有的响应消息头，但正文还没有完全收到
4	XMLHttpRequest对象读取响应结束 <b>一切都收到了</b>

1 status: HTTP的状态码

状态码	说 明
<b>200</b>	<b>服务器响应正常</b>
400	无法找到请求的资源
403	没有访问权限
404	访问的资源不存在
500	服务器内部错误

1 responseText: 获得响应的文本内容

1 responseXML: 获得响应的XML文档对象 document

注：就绪状态是4而且状态码是200，才可以处理服务器数据

使用Ajax验证用户名是否存在实现步骤：

- 1、 使用文本框的onBlur事件
- 2、 使用Ajax技术实现异步交互
  - a) 创建XMLHttpRequest对象
  - b) 通过 XMLHttpRequest对象设置请求信息
  - c) 向服务器发送请求
  - d) 创建回调函数，根据响应状态动态更新页面

## 四、案例：实现邮箱验证

my.js

```

//获取XMLHttpRequest对象
function getXMLHttpRequest() {
    var xmlhttp;
    if (window.XMLHttpRequest) { // code for all new browsers
        xmlhttp = new XMLHttpRequest();
    } else if (window.ActiveXObject) { // code for IE5 and IE6
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    }

    return xmlhttp;
}

```

register.jsp

```

window.onload=function(){
    var emailNode = document.f1.email; //得到email标签。等价于document.getElementsByName("email")[0];
    //alert(emailNode);
    emailNode.onblur = function(){ //注册事件
        var val = this.value; //得到email的文本
        var emailMsg = document.getElementById("emailMsg"); //得到span标签
        emailMsg.innerHTML = ""; //清空span中的内容
        if(val==""){
            emailMsg.innerHTML = "email不能为空"; //给span赋值
            return;
        }
    }
}

```

1、使用文本框的onBlur事件

```

function searchName(){
    //得到文本框的内容
    var nameNode = document.getElementById("name");
    if(nameNode.value==""){
        document.getElementById("content1").style.display="none";
        return;
    }
    //创建XMLHttpRequest对象
    var xhr = getXMLHttpRequest();
    //通过事件调用回调函数处理响应结果，
    xhr.onreadystatechange = function(){
        if(xhr.readyState==4){
            if(xhr.status==200){
                var str = xhr.responseText; // java,javascript,javaweb
                var ss = str.split(","); //把字符串用,分隔
                var s = "";
                for(var i=0;i<ss.length;i++){
                    s+="

servlet


```

```

public void doGet(HttpServletRequest request, HttpServletResponse response
    throws ServletException, IOException {
    PrintWriter out = response.getWriter();
    String email = request.getParameter("email");

    UserServiceImpl usi = new UserServiceImpl();
    try {
        boolean b = usi.findEmail(email);//查询email是否存在

        out.print(b);
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

```

## 五、案例2：搜索

1、编写显示数据的容器div

2、实现ajax响应数据

//创建XMLHttpRequest对象

//通过事件调用回调函数处理响应结果，

//创建一个服务器连接

//发送请求

```

<div id="content1"
    style="display:none; width:130px;height:100px;background-color:white;position:absolute;top:135px;left:858px;"
>
</div>

```



```
        //把最终结果添加到div中
        document.getElementById("content1").innerHTML = s;
        document.getElementById("content1").style.display="block";//元素将显示为块级元素
    }
}
//创建一个服务器连接
xhr.open("get", "${pageContext.request.contextPath}/servlet/searchNamesServlet?name="+nameNode.value+"&time="+new Date().getTime());
//发送请求
xhr.send(null);
}

function changeBackground_over(div){
    div.style.backgroundColor="gray";
}

function changeBackground_out(div){
    div.style.backgroundColor="";
}
```

```
function changeText(div){
    //得到文本框的内容
    var nameNode = document.getElementById("name");
    nameNode.value = div.innerHTML;
    document.getElementById("content1").style.display="none";
}
```

servlet



## 六、JSON对象

```
var arr = ["", "", 123, true];
```