

回顾:

AJAX

AJAX 全称 Asynchronous Javascript And XML（异步 JavaScript 和 XML），是一种用于创建快速动态网页的技术。从名字可以发现，Ajax 并不是一种全新的技术，而是整合了 JavaScript 和 XML 等现有技术。

AJAX = Asynchronous JavaScript and XML（异步的 JavaScript 和 XML）。

AJAX 不是新的编程语言，而是一种使用现有标准的新方法。

AJAX 最大的优点是在不重新加载整个页面的情况下，可以与服务器交换数据并更新部分网页内容。

AJAX 不需要任何浏览器插件，但需要用户允许 JavaScript 在浏览器上执行。

AJAX 简介

AJAX 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。

基础知识

- HTML / XHTML
 - CSS
 - JavaScript / DOM
-

什么是 AJAX ？

AJAX = 异步 JavaScript 和 XML。

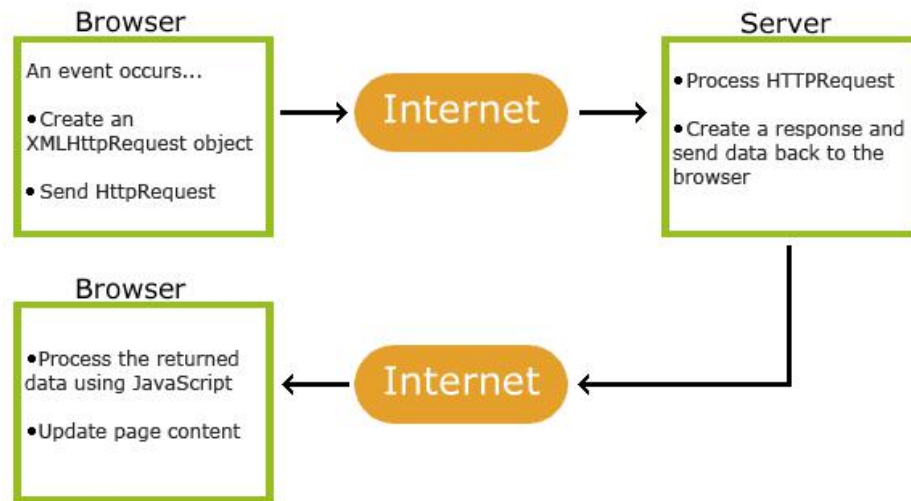
AJAX 是一种用于创建快速动态网页的技术。

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。

有很多使用 AJAX 的应用程序案例：新浪微博、Google 地图、开心网等等。

AJAX 工作原理



Json 数据格式

json 是一种与语言无关的数据交换的格式，作用：

使用 ajax 进行前后台数据交换

移动端与服务端的数据交换

Json 的格式与解析：

1) 对象格式: {key1:object,key2:object2.....} {"name": "tom", "age": 23, "sex": "m"}

2) 数组集合格式[obj,obj,obj.....] [{"stuno": "1001", "name": "tom"}, {}, {}]

案例分析

```
<script type="text/javascript" src="js/jquery-1.7.2.js" charset="utf-8" ></script>
```

```
<script type="text/javascript">
```

```
function validatePhoneNum()
```

```
{
```

```
    //获取用户输入的电话号码
```

```
    var pNum = document.getElementById("phoneId").value;
```

```
    //1.调用 ajax，创建 XMLHttpRequest 对象
```

```
    xmlhttpRequest = new XMLHttpRequest();
```

```
    //2.指定回调函数（用来处理服务器的返回数据）
```

```
    xmlhttpRequest.onreadystatechange=callBack;
```

```
    //3.准备发送请求
```

```
    xmlhttpRequest.open("post","ValidatePhoneNumberServlet",true);
```

```
    //4.设置请求头（只有 post 才可以设置请求头，get 不用设置）
```

```
    xmlhttpRequest.setRequestHeader("Content-Type", "application/x-www-form-urlencoded")
```

```
    //5.发送数据
```

```

        var data ="pNum="+pNum;    //pNum=199939399
        xmlhttpRequest.send(data);
    }
    function callBack()
    {
        if(xmlHttpRequest.readyState == 4 && xmlhttpRequest.status==200)
        {
            var result = xmlhttpRequest.responseText;
            if(result == "true")
            {
                $("#ajax_ok_id").css("display","inline");
                $("#ajax_error_id").css("display","none");
            }
            else
            {
                $("#ajax_ok_id").css("display","none");
                $("#ajax_error_id").css("display","inline");
            }
        }
    }

    function validatePhoneNum_JAjax()
    {
        var pNum = $("#phoneIdAjax").val();
        $.ajax({
            url:"ValidatePhoneNumberServlet",
            data:"pNum="+pNum,
            type:"get",
            dataType:"text",
            success:function(result){
                if(result=="false")
                {
                    alert("此号码已被用户绑定，请更换.....")
                }
                else
                {
                    alert("此号码可以使用，用户绑定成功.....")
                }
            },
            error:function(){
                alert("AJAX 请求失败");
            }
        })
    }

```

```

    });
}

function validatePhoneNum_get()
{
    var pNum = $("#phoneIdGet").val();
    $.get(
        "ValidatePhoneNumberServlet",
        "pNum="+pNum,
        function(result){
            if(result == "false")
            {
                alert("此号码已被用户绑定，请更换.....")
            }
            else
            {
                alert("此号码可以使用，用户绑定成功.....")
            }
        },
        "text"
    );
}

function validatePhoneNum_post()
{
    var pNum = $("#phoneIdPost").val();
    $.post(
        "ValidatePhoneNumberServlet",
        "pNum="+pNum,
        function(result){
            if(result == "false")
            {
                alert("此号码已被用户绑定，请更换.....")
            }
            else
            {
                alert("此号码可以使用，用户绑定成功.....")
            }
        },
        "text"
    );
}

function validatePhoneNum_load()
{

```

```

        var pNum = $("#phoneIdLoad").val();
        $("#ajax_ok_error_id").load(
            "ValidatePhoneLoadServlet",
            "pNum="+pNum
        );
    }

    function validatePhoneNum_Json()
    {
        var jText = $("#jsonTest").val();
        $.ajax({
            url:"ValidatePhoneNumberJsonServlet",
            data:{"jText":jText},
            dataType:"json",
            success:function(result){
                var s = eval(result.stu);
                document.getElementById("div1").innerHTML=s.name;
                alert(s.name+" "+s.age);
            },
            error:function(XMLHttpRequest,textStatus,errorThrown){
                alert("AJAX 请求失败"+errorThrown);
            }
        });
    }
}

</script>
</head>
<body>
传统 Ajax<br>
请绑定电话号码<input type="text" id="phoneId" onblur="validatePhoneNum()">
<font id="ajax_ok_id" style="display:none" color="red">号码可以用，以成功绑定！！</font>
<font id="ajax_error_id" style="display:none" color="green">此号码已成功绑定，请更换！！</font><br>
JQUERY:AJAX<br>
请 绑 定 电 话 号 码 <input type="text" id="phoneIdAjax"
onblur="validatePhoneNum_JAjax()"><br>
JQUERY:GET<br>
请绑定电话号码<input type="text" id="phoneIdGet" onblur="validatePhoneNum_get()"><br>
JQUERY:post<br>
请绑定电话号码<input type="text" id="phoneIdPost" onblur="validatePhoneNum_post()"><br>
JQUERY:load<br>
请绑定电话号码<input type="text" id="phoneIdLoad" onblur="validatePhoneNum_load()"><br>
<font id="ajax_ok_error_id"></font><br>
测试返回 JSON<BR>

```

```
<input type="text" id="jsonTest" onblur="validatePhoneNum_Json()"><br>
<div id="div1"></div>
```

Query 的 each 方法的几种常用的用法

Js 代码

```
var arr = [ "one", "two", "three", "four" ];
$.each(arr, function() {
    alert(this);
});
```

上面这个 each 输出的结果分别为: one,two,three,four

```
var arr1 = [[1, 4, 3], [4, 6, 6], [7, 20, 9]]
$.each(arr1, function(i, item) {
    alert(item[0]);
});
```

其实 arr1 为一个二维数组, item 相当于取每一个一维数组,
item[0]相对于取每一个一维数组里的第一个值

所以上面这个 each 输出分别为: 1 4 7

```
var obj = { one:1, two:2, three:3, four:4 };
$.each(obj, function(key, val) {
    alert(obj[key]);
});
```

这个 each, 能循环每一个属性

输出结果为: 1 2 3 4

ajax 实现页面无刷新完成 crud 操作