**JavaScript与表单交互（数据校验）：**

在JavaScript中表单的验证是最重要的部分了。对于表单数据的取得以及验证是整个JavaScript里面使用的最重要的一个场景。

所有的JavaScript中事件一定要与函数捆绑在一起使用，那么下面就利用这些事件来进行表单元素的操作。

范例：取得输入的文本信息

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>JavaScript测试二</title>  
</head>  
 <script type="text/javascript">  
 **function** showEmail() { //定义事件的处理函数  
 **var** emailElement = document.getElementById("email");  
 alert("email = " + emailElement.value)  
 }  
 </script>  
<body>  
 <form >  
 请输入email地址：<input type="text" **name = "email"** id = "email"  
 >  
 <button onclick="showEmail()">显示内容</button>  
  
 </form>  
</body>  
</html>

在表单控件里面name和ID的内容一定要保持一致。

以上是通过**getElementById()**的函数取得了一个指定的对象但是除了以上的操作之外还有另外一种取得方式（不用了），

在取得了输入的内容之后，更多的是希望进行一下数据的验证，因为只有合法的email才应该进行正常的显示，如果要想使用数据验证，使用正则表达式是最方便的，而JavaScript也支持正则，但是需要记住的是要写上开始于结束标记。而且所有的正则都不需要进行转义。

**正则表达式格式：/^正则表达式$/.test(要检测数据)**

<!DOCTYPE html>  
<html lang=**"en"**>  
<head>  
 <meta charset=**"UTF-8"**>  
 <title>**JavaScript测试二**</title>  
</head>  
 <script type=**"text/javascript"**>  
 function showEmail() { //定义事件的处理函数  
 var emailElement = document.getElementById("email");  
 ///^正则表达式$/.test(要检测数据);  
 if(/^\w+@\w+\.\w+$/.test(emailElement.value)){  
 alert("输入的数据合法")  
 }else{  
 alert("输入的数据不合法")  
 }  
 }  
 </script>  
<body>  
 <form >  
 **请输入email地址：**<input type=**"text"** name = **"email"** id = **"email"** >  
 <button onclick=**"**showEmail()**"**>**显示内容**</button>  
  
 </form>  
</body>  
</html>

以上就已经成功的实现了数据的取出以及验证操作。

虽然表面上看已经实现了，但是实际上使不了，一个正常的表单验证操作之中，其基本流程：由用户输入表单内容，如果输入的数据合法则允许提交，如果不合法则不允许提交，但是此时的内容即使输入的数据不合法也可以进行提交。

如果要想针对于表单验证进行拦截，则必须使用onsubmit事件，这个事件是在**”<form>”元素上定义的**，表示进行表单提交时触发，如果 需要对表单进行蓝企鹅，那么就在onsubmit处理函数上返回false，否则返回true。

范例：

<!DOCTYPE html>  
<html lang=**"en"**>  
<head>  
 <meta charset=**"UTF-8"**>  
 <title>**JavaScript测试二**</title>  
</head>  
 <script type=**"text/javascript"**>  
 function validateEmail() { //定义事件的处理函数  
 var emailElement = document.getElementById("email");  
 // /^正则表达式$/.test(要检测数据);  
 if(/^\w+@\w+\.\w+$/.test(emailElement.value)){  
 alert("输入的数据合法");  
 return true;  
 }else{  
 alert("输入的数据不合法");  
 return false;  
 }  
 }  
 function validate() {  
 return validateEmail();  
 }  
 </script>  
<body>  
 <form action=**"show.html"** method=**"post"** onsubmit=**"**return validate()**"**>  
 **请输入email地址：**<input type=**"text"** name = **"email"** id = **"email"** >  
 <button type=**"submit"**>**显示内容**</button>  
  
  
 </form>  
</body>  
</html>

范例：定义一个style.css的样式文件

.init{ //初始化  
 **background**: #F5F5F5;  
 font-weight: **bold**;  
 color: **white**;  
}  
.wrong{  
 background: #F5F5F5;  
 font-weight: **bold** ;  
 border: 1**px** #FF0000 **solid**;  
 color: #000000;  
}  
.right{  
 background: #F5F5F5;  
 font-weight: **bold**;  
 border: 1**px** #FF0000 **solid**;  
 color: #000000;  
}

调用样式表：

<form action=**"show.html"** method=**"post"** onsubmit=**"**return validate()**"**>  
 **请输入email地址：**<input type=**"text"** name = **"email"** id = **"email"** class=**"wrong"**>  
 <button type=**"submit"**>**显示内容**</button>  
  
  
</form>

在每一个表单的初始状态的时候使用“.init”样式完成，而如果验证失败则用“.wrong”样式完成，反之则用“.right”完成。最好的验证方式还是在文本框失去焦点之后进行验证。

·**获得焦点的事件为onfocus失去焦点的事件叫onblur**

**范例：**

<!DOCTYPE html>  
<html lang=**"en"**>  
<head>  
 <meta charset=**"UTF-8"**>  
 **<link rel="stylesheet" type="text/css" href="style.css">//导入.css文件** <title>**JavaScript测试二**</title>  
</head>  
 <script type=**"text/javascript"**>  
 function validateEmail() { //定义事件的处理函数  
 var emailElement = **document.getElementById("email");//取得文本框对象** var msgElement = document.getElementById("emailMsg");  
 // /^正则表达式$/.test(要检测数据);  
 if(/^\w+@\w+\.\w+$/.test(emailElement.value)){  
 emailElement.className = "right";  
 msgElement.innerHTML = "<font color='green'>email输入正确！</font>"  
 return true;  
 }else{  
 emailElement.className="wrong"; //设置文本框显示风格  
 msgElement.**innerHTML** = "<font color='red'>email输入错误！</font>"  
 return false;  
 }  
 }  
 function validate() {  
 return validateEmail();  
 }  
 </script>  
<body>  
 <form action=**"show.html"** method=**"post" onsubmit="return validate()">**  
 **请输入email地址：**<input type=**"text"** name = **"email"** id = **"email"** class=**"init"** onblur=**"**validate()**"**><span id = **"emailMsg"**></span><br>  
 <button type=**"submit"**>**显示内容**</button>  
  
  
 </form>  
</body>  
</html>

.init{ //初始化  
 **background**: #F5F5F5;  
 font-weight: **bold**;  
 color: **white**;  
}  
.wrong{  
 background: #F5F5F5;  
 font-weight: **bold** ;  
 border: 1**px** #FF0000 **solid**;  
 color: #000000;  
}  
.right{  
 background: #F5F5F5;  
 font-weight: **bold**;  
 border: 1**px** #00FF00 **solid**;  
 color: #00FF00;  
}

**以后的表单验证的基础结构都使用上面的方式进行控制。**

**在以上的程序中onsubmit是最为重要的，onsubmit="return validate() 必须要加return表示接受函数的返回值，当onsubmit的值为true的时候就执行表单提交，否则不进行表单提交。**

**<link rel="stylesheet" type="text/css" href="style.css">** 用于导入.css样式文件。

.**innerHTML** ：表示在取得的元素中追加内容，可以是HTML代码。

**密码框验证：**

对用户的输入密码进行验证，而且对用户的确认密码进行与输入密码的比较是否相同，若相同则通过验证，若不同则不能通过验证

在以下例子中将不再使用onblur=”函数（）”而是将JavaScript写到一个单独的demo.js文件中，然后直接导入，在js文件中利用window.onload = function(){}进行调用

**范例：**

window.onload = function () {  
 document.getElementById("pwd").addEventListener("blur",validatePwd,false);  
 document.getElementById("conf").addEventListener("blur",validateConf,false);  
}  
function validateEmpty(elementName) {//验证是否为空  
 var objElement = document.getElementById(elementName);  
 var msgElement = document.getElementById(elementName + "Msg");//取得验证元素  
  
 if(objElement.value != ""){ //不为空  
 objElement.className = "right";//设置颜色  
 msgElement.innerHTML = "<font color = 'green'>输入内容正确！</font>";  
 return true;  
 }else{  
 objElement.className = "wrong";  
 msgElement.innerHTML = "<font color = 'red'>输入的内容错误！</font>";  
 return false;  
 }  
}  
function validatePwd() {//验证密码  
 return validateEmpty("pwd");  
}  
function validateConf() {  
 return validateRepeat("pwd","conf");  
}  
function validateRepeat(srcName,descName) {//验证两次密码是否一致  
 var srcElement = document.getElementById(srcName);  
 var descElement = document.getElementById(descName);  
 var confMsg = document.getElementById(descName + "Msg");  
  
 if(srcElement.value == descElement.value && srcElement.value != "" && descElement.value != ""){  
 descElement.className = "right";  
 confMsg.innerHTML = "<font color =‘green’>两次密码一致！</font>";  
 return true;  
 }else{  
 confMsg.innerHTML = "<font color =‘red’>两次密码不一致！</font>";  
 descElement.className = "wrong";  
 return false;  
 }  
  
}

<!DOCTYPE html>  
<html lang=**"en"**>  
<head>  
 <meta charset=**"UTF-8"**>  
 <link rel=**"stylesheet"**type=**"text/css"**href=**"style.css"**>  
 <script type=**"text/javascript"** src=**".idea/demo.js"**></script>  
 <title>**Title**</title>  
  
</head>  
<body>  
 **输入密码：**<input type=**"password"** name=**"pwd"**id=**"pwd"** >  
 <span id=**"pwdMsg"**></span><br>  
 **确认密码：**<input type=**"password"** name=**"conf"** id=**"conf"** >  
 <span id=**"confMsg"**></span><br>  
 <input type=**"submit"**>  
</body>  
</html>

以上程序中使用了：

window.onload = function () {  
 document.getElementById("pwd").addEventListener("blur",validatePwd,false);  
 document.getElementById("conf").addEventListener("blur",validateConf,false);  
}

在JavaScript中自动调用，就可以免去写onblur的麻烦，使得HTML页面更加干净。

**单选按钮操作：**

单选按钮的操作特点是多个元素的ID和name相同，下面首先来确定如何取得单选按钮的内容。在日后（特别是单选和复选框中）如果出现了多个ID和name名称相同的元素时那么JavaScript会认为它是一个对象数组，但是如果此时你只是使用document.getElementById()进行取得的话，那么你只能取得数组中的第一个元素。

如果想要取得一组全部的对象，则可以使用document.all()来完成，他返回的是对象数组。

**范例：**

<!DOCTYPE html>  
<html lang=**"en"**>  
<head>  
 <meta charset=**"UTF-8"**>  
 <title></title>  
 <script type=**"text/javascript"**>  
 function showSex() {  
 var sex = document.all("sex");  
 if(sex[0].checked){  
 alert("性别是：" + sex[0].value);  
 }  
 if(sex[1].checked){  
 alert("性别是：" + sex[1].value);  
 }  
 }  
 </script>  
  
</head>  
<body>  
 **请选择性别：**<input type = **"radio"** name = **"sex"** id=**"sex"** value=**"男"** checked>**男&nbsp;** <input type=**"radio"** name=**"sex"** id=**"sex"** value=**"女"**>**女** <button onclick=**"**showSex()**"**>**显示性别**</button>  
  
</body>  
</html>

**复选按钮：**

复选按钮和单选按钮很相似都是用对象数组存储的，

范例：实现一个复选框数据的取得：

<!DOCTYPE html>  
<html lang=**"en"**>  
<head>  
 <meta charset=**"UTF-8"**>  
 <title>**Title**</title>  
 <script>  
 function showInst() {  
 var inst = document.all("inst");//多个对象元素  
 var str = "文攀的兴趣：";  
 for(var x = 0;x < inst.length;x++){  
 if(inst[x].checked){  
 str = str + inst[x].value + "、";  
 }  
 }  
 alert(str);  
 }  
 </script>  
</head>  
<body>  
 **请选择文攀的爱好：**<br>  
 <input type = **"checkbox"** value=**"篮球"** name = **"inst"**id = **"inst"**>**篮球**<br>  
 <input type = **"checkbox"** value=**"排球"** name = **"inst"**id = **"inst"**>**排球**<br>  
 <input type = **"checkbox"** value=**"足球"** name = **"inst"**id = **"inst"**>**足球**<br>  
 <input type = **"checkbox"** value=**"跑步"** name = **"inst"**id = **"inst"**>**跑步**<br>  
 <input type = **"checkbox"** value=**"看书"** name = **"inst"**id = **"inst"**>**看书**<br>  
 <input type = **"checkbox"** value=**"写字"** name = **"inst"**id = **"inst"**>**写字**<br>  
 <input type = **"checkbox"** value=**"游泳"** name = **"inst"**id = **"inst"**>**游泳**<br>  
 <input type = **"checkbox"** value=**"画画"** name = **"inst"**id = **"inst"**>**画画**<br>  
  
 <button onclick=**"**showInst()**"**>**显示兴趣**</button>  
  
</body>  
</html>

此时的操作与单选按钮的功能是类似的，但是这个程序还不足，可以再做一个全选功能。

全选功能的设计主要是操作复选框中的checked属性内容。如果此时的内容是true，表示选中，否则没有选中。

**全选功能其实就是将表单中所有的复选框的checked状态都设置为全选按钮一样。**

范例：全选功能

<!DOCTYPE html>  
<html lang=**"en"**>  
<head>  
 <meta charset=**"UTF-8"**>  
 <title>**Title**</title>  
 <script>  
 function showInst() {  
 var inst = document.all("inst");//多个对象元素  
 var str = "文攀的兴趣：";  
 for(var x = 0;x < inst.length;x++){  
 if(inst[x].checked){  
 str = str + inst[x].value + "、";  
 }  
 }  
 alert(str);  
 }  
 function selectAll() {  
  **var insts = document.all("inst");  
 var select = document.getElementById("select");  
 for(var x = 0;x < insts.length;x++){  
 insts[x].checked = select.checked;** }  
 }  
 </script>  
</head>  
<body>  
 **请选择文攀的爱好：**<br>  
 <input type = **"checkbox"** value=**"篮球"** name = **"inst"**id = **"inst"**>**篮球**<br>  
 <input type = **"checkbox"** value=**"排球"** name = **"inst"**id = **"inst"**>**排球**<br>  
 <input type = **"checkbox"** value=**"足球"** name = **"inst"**id = **"inst"**>**足球**<br>  
 <input type = **"checkbox"** value=**"跑步"** name = **"inst"**id = **"inst"**>**跑步**<br>  
 <input type = **"checkbox"** value=**"看书"** name = **"inst"**id = **"inst"**>**看书**<br>  
 <input type = **"checkbox"** value=**"写字"** name = **"inst"**id = **"inst"**>**写字**<br>  
 <input type = **"checkbox"** value=**"游泳"** name = **"inst"**id = **"inst"**>**游泳**<br>  
 <input type = **"checkbox"** value=**"画画"** name = **"inst"**id = **"inst"**>**画画**<br>  
 <br><br>  
 <input type=**"checkbox"** name=**"select"** id=**"select"** value=**"全选"** onclick=**"**selectAll()**"**>**全选** <button onclick=**"**showInst()**"**>**显示兴趣**</button>  
  
</body>  
</html>

下拉列表框：

下拉列表框如果要进行表单交互使用的事件是onchange事件，表示下拉内容发生改变。

范例：观察下拉列表框

<!DOCTYPE html>  
<html lang=**"en"**>  
<head>  
 <meta charset=**"UTF-8"**>  
 <title>**Title**</title>  
</head>  
<script>  
 function showCity(city) {  
 **//alert("你的家乡是：" + city);**  
 //也可以通过传统方法取得值  
  **alert("你的家乡是：" + document.getElementById("city").value)** }  
</script>  
<body>  
 **你的家乡：** <select name = **"city"** id=**"city" onchange="showCity(this.value)">**  
 <option value=**"北京"**>**北京**</option>  
 <option value=**"成都"**>**成都**</option>  
 <option value=**"上海"**>**上海**</option>  
 <option value=**"天津"**>**天津**</option>  
  
 </select><br>  
  
</body>  
</html>

下拉列表框之中最主要的事件就是onchange事件，你可以传递当前元素对象的内容，也可以按照传统方式取得元素对象。

**文本域：**

文本域主要是进行打文本信息的输入操作使用的，但是在文本域是哪个同样也可以进行一系列操作事件

范例：设置文本域的最大输入字符个数，若超过最大数字则不能提交

<!DOCTYPE html>  
<html lang=**"en"**>  
<head>  
 <meta charset=**"UTF-8"**>  
 <title>**Title**</title>  
 <script type=**"text/javascript"**>  
 function validateNote() {  
 var note = document.getElementById("note");  
 var msg = document.getElementById("noteMsg");  
 //console.log("执行了");  
 //msg.innerHTML = "输入数据长度：" + note.value.length;  
 **var len = note.value.length;**  
 //取得输入长度  
 if(len <= 10){  
 msg.innerHTML = "还可以输入"+(10-len)+"个长度的数据";  
 }else{  
 msg.innerHTML = "<font color = 'red'> 输入的数据内容长度过大</font>";  
  **document.getElementById("sub").disabled="disabled";**  //提交按钮点不动  
 }  
 }  
 </script>  
  
  
  
</head>  
<body>  
 <form action = **"show.html"** method=**"post"**>  
 **描述：<textarea id="note" name="note" cols="50" rows="10"  
 onkeydown="validateNote()" onkeypress="validateNote()">hello</textarea><span id="noteMsg"></span><br>**  
 <button type=**"submit"** name=**"sub"** id=**"sub"** value=**"保存"**>**保存**</button>  
  
 </form>  
  
</body>  
</html>

**总结：**

**以上所给出的例子是日常在网络中都是很常见的功能，当然整个的过程之中发现所有过程之中的动态检查都是通过JavaScript完成的，**

**对于表单的验证操作是本次的核心重点。**

**window对象：**

在之前调用的onload的时候曾经使用过window.onload=function(){},这就属于window对象。例如在之前警告框的弹出。

<script type=**"text/javascript"**>  
 window.alert("helloworld");  
</script>

在window对象里面有个稍微重要一点的函数叫confirm()，这个函数的功能是提供一个确认框

<!DOCTYPE html>  
<html lang=**"en"**>  
<head>  
 <meta charset=**"UTF-8"**>  
 <title>**Title**</title>  
 <script type=**"text/javascript"**>  
 function deleteConfirm() {  
 return window.confirm("确定要删除吗？");  
 }  
  
 </script>  
  
</head>  
<body>  
 <a href=**"Xxx"** onclick=**"**return deleteConfirm()**"**>**删除邮件**</a>  
</body>  
</html>

还有一个prompt()函数，此函数可以由用户输入数据。

<!DOCTYPE html>  
<html lang=**"en"**>  
<head>  
 <meta charset=**"UTF-8"**>  
 <title>**Title**</title>  
 <script type=**"text/javascript"**>  
  **var *str* = window.prompt("请输入你的姓名","姓名是？")**  
 document.write(*str*);  
 </script>  
  
</head>  
<body>  
</body>  
</html>

但是这个函数现在不怎么用了。

还可以在一个页面里面打开另外一个页面：

cript type=**"text/javascript"**>  
window.open("show.html","AD页","width=150;height=130;scollbars=yes","resizeable=no");  
</script>

在弹窗的时候也可以设置一些显示属性

也可以设置关闭窗口：

<head>  
 <meta charset=**"UTF-8"**>  
 <title>**Title**</title>  
 <script type=**"text/javascript"**>  
 function fun() {  
 window.close();  
 }  
 </script>  
  
</head>  
<body>  
 <button onclick=**"**fun()**"**>**关闭窗口**</button>  
</body>  
</html>

**在这里还有一个重要的操作，叫做重定向：**

<!DOCTYPE html>  
<html lang=**"en"**>  
<head>  
 <meta charset=**"UTF-8"**>  
 <title>**Title**</title>  
 <script type=**"text/javascript"**>  
 function go(url) {  
 **window.location = url;**  
 }  
 </script>  
  
</head>  
<body>  
 <select id=**"url"** name=**"url"** onchange=**"**go(this.value)**"**>  
 <option>**请选择要显示的页面**</option>  
 <option value=**"show.html"**>**show**</option>  
 <option value=**"www.baidu.com"**>**baidu**</option>  
 </select>  
</body>  
</html>

这个操作是整个wind对象里面最重要的操作。相当于一个超链接

wind对象表示的是窗口的操作