**[表单验证代码全集](http://www.cnblogs.com/tecsoon/archive/2009/06/08/1498664.html)**

1:js 字符串长度限制、判断字符长度、js限制输入、限制不能输入、textarea 长度限制  
2.:js判断汉字、判断是否汉字 、只能输入汉字  
3:js判断是否输入英文、只能输入英文  
4:js只能输入数字,判断数字、验证数字、检测数字、判断是否为数字、只能输入数字  
5:只能输入英文字符和数字  
6: js email验证 、js 判断email 、信箱/邮箱格式验证  
7:js字符过滤,屏蔽关键字  
8:js密码验证、判断密码  
2.1: js 不为空、为空或不是对象 、判断为空、判断不为空  
2.2:比较两个表单项的值是否相同  
2.3:表单只能为数字和"\_",  
2.4:表单项输入数值/长度限定  
2.5:中文/英文/数字/邮件地址合法性判断  
2.6:限定表单项不能输入的字符  
2.7表单的自符控制  
2.8:form文本域的通用校验函数  
  
1. 长度限制  
<script>  
function test()  
{  
if(document.a.b.value.length>50)  
{  
alert("不能超过50个字符！");  
document.a.b.focus();  
return false;  
}  
}  
</script>  
<form name=a onsubmit="return test()">  
<textarea name="b" cols="40" wrap="VIRTUAL" rows="6"></textarea>  
<input type="submit" name="Submit" value="check">  
</form>  
  
2. 只能是汉字  
<input onkeyup="value="/oblog/value.replace(/[^\u4E00-\u9FA5]/g,'')">  
  
3." 只能是英文  
<script language=javascript>  
function onlyEng()  
{  
if(!(event.keyCode>=65&&event.keyCode<=90))  
event.returnvalue=false;  
}  
</script>  
  
<input onkeydown="onlyEng();">  
  
4. 只能是数字  
<script language=javascript>  
function onlyNum()  
{  
if(!((event.keyCode>=48&&event.keyCode<=57)||(event.keyCode>=96&&event.keyCode<=105)))  
//考虑小键盘上的数字键  
event.returnvalue=false;  
}  
</script>  
  
<input onkeydown="onlyNum();">  
  
5. 只能是英文字符和数字  
<input onkeyup="value="/oblog/value.replace(/[\W]/g,"'') "onbeforepaste="clipboardData.setData('text',clipboardData.getData('text').replace(/[^\d]/g,''))">  
  
6. 验证邮箱格式  
<SCRIPT LANGUAGE=javascript RUNAT=Server>  
function isEmail(strEmail) {  
if (strEmail.search(/^\w+((-\w+)|(\.\w+))\*\@[A-Za-z0-9]+((\.|-)[A-Za-z0-9]+)\*\.[A-Za-z0-9]+$/) != -1)  
return true;  
else  
alert("oh");  
}  
</SCRIPT>  
<input type=text onblur=isEmail(this.value)>  
  
7. 屏蔽关键字(这里屏蔽\*\*\*和\*\*\*\*)  
<script language="javascript1.2">  
function test() {  
if((a.b.value.indexOf ("\*\*\*") == 0)||(a.b.value.indexOf ("\*\*\*\*") == 0)){  
alert(":)");  
a.b.focus();  
return false;}  
}  
</script>  
<form name=a onsubmit="return test()">  
<input type=text name=b>  
<input type="submit" name="Submit" value="check">  
</form>  
  
8. 两次输入密码是否相同  
<FORM METHOD=POST ACTION="">  
<input type="password" id="input1">  
<input type="password" id="input2">  
<input type="button" value="test" onclick="check()">  
</FORM>  
<script>  
function check()  
{  
with(document.all){  
if(input1.value!=input2.value)  
{  
alert("false")  
input1.value = "";  
input2.value = "";  
}  
else document.forms[0].submit();  
}  
}  
</script>  
够了吧 :)  
屏蔽右键 很酷  
oncontextmenu="return false" ondragstart="return false" onselectstart="return false"  
加在body中  
  
  
二  
  
2.1 表单项不能为空  
  
<script language="javascript">  
<!--  
function CheckForm()  
{  
if (document.form.name.value.length == 0) {  
alert("请输入您姓名!");  
document.form.name.focus();  
return false;  
}  
return true;  
}  
-->  
</script>  
  
2.2 比较两个表单项的值是否相同  
  
<script language="javascript">  
<!--  
function CheckForm()  
if (document.form.PWD.value != document.form.PWD\_Again.value) {  
alert("您两次输入的密码不一样！请重新输入.");  
document.ADDUser.PWD.focus();  
return false;  
}  
return true;  
}  
-->  
</script>  
  
2.3 表单项只能为数字和"\_",用于电话/银行帐号验证上,可扩展到域名注册等  
  
<script language="javascript">  
<!--  
function isNumber(String)  
{  
var Letters = "1234567890-"; //可以自己增加可输入值  
var i;  
var c;  
if(String.charAt( 0 )=='-')  
return false;  
if( String.charAt( String.length - 1 ) == '-' )  
return false;  
for( i = 0; i < String.length; i ++ )  
{  
c = String.charAt( i );  
if (Letters.indexOf( c ) < 0)  
return false;  
}  
return true;  
}  
function CheckForm()  
{  
if(! isNumber(document.form.TEL.value)) {  
alert("您的电话号码不合法！");  
document.form.TEL.focus();  
return false;  
}  
return true;  
}  
-->  
</script>  
  
  
2.4 表单项输入数值/长度限定  
  
<script language="javascript">  
<!--  
function CheckForm()  
{  
if (document.form.count.value > 100 || document.form.count.value < 1)  
{  
alert("输入数值不能小于零大于100!");  
document.form.count.focus();  
return false;  
}  
if (document.form.MESSAGE.value.length<10)  
{  
alert("输入文字小于10!");  
document.form.MESSAGE.focus();  
return false;  
}  
return true;  
}  
//-->  
</script>  
  
2.5 中文/英文/数字/邮件地址合法性判断  
  
<SCRIPT LANGUAGE="javascript">  
function isEnglish(name) //英文值检测  
{  
 if(name.length == 0)  
 return false;  
 for(i = 0; i < name.length; i++) {  
 if(name.charCodeAt(i) > 128)  
 return false;  
 }  
 return true;  
}  
  
function isChinese(name) //中文值检测  
{  
 if(name.length == 0)  
 return false;  
 for(i = 0; i < name.length; i++) {  
 if(name.charCodeAt(i) > 128)  
 return true;  
 }  
 return false;  
}  
  
function isMail(name) // E-mail值检测  
{  
 if(! isEnglish(name))  
 return false;  
 i = name.indexOf(" at ");  
 j = name dot lastIndexOf(" at ");  
 if(i == -1)  
 return false;  
 if(i != j)  
 return false;  
 if(i == name dot length)  
 return false;  
 return true;  
}  
  
function isNumber(name) //数值检测  
{  
if(name.length == 0)  
return false;  
for(i = 0; i < name.length; i++) {  
if(name.charAt(i) < "0" || name.charAt(i) > "9")  
return false;  
}  
return true;  
}  
  
function CheckForm()  
{  
if(! isMail(form.Email.value)) {  
alert("您的电子邮件不合法！");  
form.Email.focus();  
return false;  
}  
if(! isEnglish(form.name.value)) {  
alert("英文名不合法！");  
form.name.focus();  
return false;  
}  
if(! isChinese(form.cnname.value)) {  
alert("中文名不合法！");  
form.cnname.focus();  
return false;  
}  
if(! isNumber(form.PublicZipCode.value)) {  
alert("邮政编码不合法！");  
form.PublicZipCode.focus();  
return false;  
}  
return true;  
}  
//-->  
</SCRIPT>  
  
2.6 限定表单项不能输入的字符  
  
<script language="javascript">  
<!--  
  
function contain(str,charset)// 字符串包含测试函数  
{  
var i;  
for(i=0;i<charset.length;i++)  
if(str.indexOf(charset.charAt(i))>=0)  
return true;  
return false;  
}  
  
function CheckForm()  
{  
if ((contain(document.form.NAME.value, "%\(\)><")) || (contain(document.form.MESSAGE.value, "%\(\)><")))  
{  
alert("输入了非法字符");  
document.form.NAME.focus();  
return false;  
}  
return true;  
}  
//-->  
</script>  
  
1. 检查一段字符串是否全由数字组成  
---------------------------------------  
<script language="Javascript"><!--  
function checkNum(str){return str.match(/\D/)==null}  
alert(checkNum("1232142141"))  
alert(checkNum("123214214a1"))  
// --></script>  
  
2. 怎么判断是否是字符  
---------------------------------------  
if (/[^\x00-\xff]/g.test(s)) alert("含有汉字");  
else alert("全是字符");  
  
3. 怎么判断是否含有汉字  
---------------------------------------  
if (escape(str).indexOf("%u")!=-1) alert("含有汉字");  
else alert("全是字符");  
  
4. 邮箱格式验证  
---------------------------------------  
//函数名：chkemail  
//功能介绍：检查是否为Email Address  
//参数说明：要检查的字符串  
//返回值：0：不是 1：是  
function chkemail(a)  
{ var i=a.length;  
var temp = a.indexOf('@');  
var tempd = a.indexOf('.');  
if (temp > 1) {  
if ((i-temp) > 3){  
if ((i-tempd)>0){  
return 1;  
}  
  
}  
}  
return 0;  
}  
  
**5. 数字格式验证**  
---------------------------------------  
//函数名：fucCheckNUM  
//功能介绍：检查是否为数字  
//参数说明：要检查的数字  
//返回值：1为是数字，0为不是数字  
function fucCheckNUM(NUM)  
{  
var i,j,strTemp;  
strTemp="0123456789";  
if ( NUM.length== 0)  
return 0  
for (i=0;i<NUM.length;i++)  
{  
j=strTemp.indexOf(NUM.charAt(i));  
if (j==-1)  
{  
//说明有字符不是数字  
return 0;  
}  
}  
//说明是数字  
return 1;  
}  
  
6. 电话号码格式验证  
---------------------------------------  
//函数名：fucCheckTEL  
//功能介绍：检查是否为电话号码  
//参数说明：要检查的字符串  
//返回值：1为是合法，0为不合法  
function fucCheckTEL(TEL)  
{  
var i,j,strTemp;  
strTemp="0123456789-()# ";  
for (i=0;i<TEL.length;i++)  
{  
j=strTemp.indexOf(TEL.charAt(i));  
if (j==-1)  
{  
//说明有字符不合法  
return 0;  
}  
}  
//说明合法  
return 1;  
}  
  
**7. 判断输入是否为中文的函数**  
---------------------------------------  
function ischinese(s){  
var ret=true;  
for(var i=0;i<s.length;i++)  
ret=ret && (s.charCodeAt(i)>=10000);  
return ret;  
}  
 **8. 综合的判断用户输入的合法性的函数**  
---------------------------------------  
<script language="javascript">  
//限制输入字符的位数开始  
//m是用户输入，n是要限制的位数  
function issmall(m,n)  
{  
if ((m<n) && (m>0))  
{  
return(false);  
}  
else  
{return(true);}  
}  
  
**9. 判断密码是否输入一致**  
---------------------------------------  
function issame(str1,str2)  
{  
if (str1==str2)  
{return(true);}  
else  
{return(false);}  
}  
  
10. 判断用户名是否为数字字母下滑线  
---------------------------------------  
function notchinese(str){  
var reg=/[^A-Za-z0-9\_]/g  
if (reg.test(str)){  
return (false);  
}else{  
return(true); }  
}  
  
2.8. form文本域的通用校验函数  
---------------------------------------  
作用：检测所有必须非空的input文本，比如姓名，账号，邮件地址等等。  
该校验现在只针对文本域，如果要针对form里面的其他域对象，可以改变判断条件。  
  
使用方法：在要检测的文本域中加入title文字。文字是在提示信息，你要提示给用户的该字段的中文名。比如要检测用户名  
html如下<input name="txt\_1" title="姓名">,当然，最好用可视化工具比如dreamweaver什么的来编辑域。  
如果要检测数字类型数据的话，再把域的id统一为sz.  
javascript判断日期类型比较麻烦，所以就没有做日期类型校验的程序了.高手可以补充。  
  
程序比较草，只是提供一个思路。抛砖引玉！：）  
哦，对了，函数调用方法：< form onsubmit="return dovalidate()">  
  
function dovalidate()  
{  
fm=document.forms[0] //只检测一个form,如果是多个可以改变判断条件  
for(i=0;i<fm.length;i++)  
{  
//检测判断条件，根据类型不同可以修改  
if(fm*.tagName.toUpperCase()=="INPUT" &&fm*.type.toUpperCase()=="TEXT" && (fm*.title!=""))  
  
 if(fm*.value="/blog/="")//  
{  
str\_warn1=fm*.title+"不能为空!";  
 alert(str\_warn1);  
 fm*.focus();  
return false;  
}  
if(fm*.id.toUpperCase()=="SZ")//数字校验  
 {  
 if(isNaN(fm*.value))  
{ str\_warn2=fm*.title+"格式不对";  
 alert(str\_warn2);  
 fm*.focus();  
return false;  
}  
}  
}  
return true;  
}

posted @ 2009-06-08 13:38 DELPHI&.NET初学者[TECSOON WENDAY] 阅读(23) | 评论(0) | [编辑](#)

**[建立JavaScript正则表达式库简化表单验证](http://www.cnblogs.com/tecsoon/archive/2009/06/08/1498663.html)**

　　 本文介绍如何创建一个可用于任何Web页面的“正则表达式库”——通过将表单验证代码保存为一个独立的文件，我们可以避免为不同表单重写类似的验证代码，只需在HTML页面中包含这个库文件即可。  
  
　　 一、HTML元素属性与JavaScript对象属性  
  
　　 我们知道，HTML允许我们为元素指定自定义属性。如对于表单中的文本输入框元素，我们可以给它指定一个称为validator的属性：  
  
　　<　form id="exampleForm" >  
　　<　input type="text" name="input1" validator="whatsThisFor?" >  
　　...  
　<　/form >  
  
　　 该属性将被浏览器的表现引擎忽略，即它对于页面的显示是没有任何影响的。不过，就像所有其它“正式的”属性一样，自定义属性对脚本语言也是可见的，即我们可以在JavaScript代码中引用它并分析它的值。当然，在IE4和N4中引用元素的方法是不同的。下面这个引用自定义属性的示例只能在IE4（及其更高版本）下工作，但其原理也适用于Netscape浏览器：  
  
　if(document.all.exampleForm.input1.validator=="whatsThisFor")  
　　alert("Hello !");  
　else ...  
  
　　 如果属性没有定义，则试图引用它时将返回空值，在if语句的表达式里它被视为false：  
  
　if(!document.all.exampleForm.input1.validator)  
　　alert("No validator!");  
  
　　 下面我们来看看如何在表单验证中应用这种自定义属性。  
  
　　 二、正则表达式与模式匹配  
  
　　 许多表单验证任务包含了模式匹配操作。例如我们要验证这样一个图书编号值的合法性：它或者为5个数字字符，或为10个字符——5个数字，一个连字号，再加上4个数字。虽然这些验证任务也可以用一个专用的函数来完成，但用下面这个正则表达式模式与用户输入值比较更为简便：  
  
　 /\d{5}(-\d{4})?/  
  
　　 如果你以前从来没有用过正则表达式，可以将它理解为一种指定字符模式的语言，其主要操作就是匹配。大多数字符只同它们自己匹配，如正则表达式/abc/只匹配字符串“abc”；但也提供了引用某一组字符（如全部数字）以及指定匹配字符个数的方法。在上例中：d{5}匹配的是任意5个数字，-d{4}匹配的是一个连字号加4个数字，?说明模式中的最后一个元素是可选的，可包含也可不包含。  
  
　　 三、正则表达式库及其应用  
  
　　 所谓的正则表达式库就是包含验证常用表单元素的正则表达式的脚本文件，如：  
  
　var PatternsDict = new Object();  
　// 匹配图书编号  
　PatternsDict.bookPat = /^\d{5}(-\d{4})?$/;  
　// 匹配12:34以及75:83  
　PatternsDict.timePat = /^\d{2}:\d{2}$/;  
　// 匹配5:04 或12:34,但不匹配75:83  
　PatternsDict.timePat2=/^([1-9]|1[0-2]):[0-5]d$/;  
  
　　 文件中还要包含下面这个提供验证功能的函数：  
  
　function validateForm(theForm){// 若验证通过则返回true  
　 var elArr = theForm.elements; // 将表单中的所有元素放入数组  
　 for(var i = 0; i < elArr.length; i++)  
　　with(elArr[i]){　　　　　　　// 对于表单中的每一个元素...  
　　 var v = elArr[i].validator; // 获取其validator属性  
　　 if(!v) continue;　　　　　　// 如果该属性不存在,忽略当前元素  
　　 var thePat = PatternsDict[v];　 // 选择验证用的正则表达式  
　　 var gotIt = thePat.exec(value); // 用正则表达式验证elArr[i]的值  
　　 if(!gotIt){  
　　　alert(name + ": 输入值与正则表达式不匹配（" + v + " —— " + value + "）"); return false;}  
　　} return true;  
　 }  
  
　　 validateForm函数检查每一个表单元素的validator属性，若该属性不存在，脚本忽略此元素并返回循环的开头。否则，从PatternDict中获得当前元素的匹配模式。每一个正则表达式都有对应的RegExp对象，RegExp对象的exec()方法在参数与它所关联的正则表达式不匹配时返回null——此时脚本显示警告对话框。在实际应用场合，为提高效率可以在执行验证之前将这些正则表达式编译。某些输入域可能需要除了简单匹配之外更为复杂的验证。因此，除了模式库之外，还可以为常用的验证任务建立一个函数库。  
  
　　 设上述验证模式文件为valPatterns.js，将它包含到所有必需用它来验证输入的页面中，然后为每个输入框指定合适的validator属性即可。下面是应用上述valPatterns.js的一个完整示例：  
  
　<　HTML　><　HEAD　>  
　<　TITLE　> formVal.HTM <　/TITLE　>  
　<　SCRIPT src=" ValPatterns.js" >  
　<　/SCRIPT　>  
　<　SCRIPT　>  
　function showForm(theForm){  
　 alert("表单：" + theForm.id + "已经提交");  
　}  
　<　/SCRIPT　><　/HEAD　>  
　  
　<　BODY　><　P　>用正则表达式库验证表单示例.<　/P　>  
　<　FORM id = "theForm" onSubmit = "return validateForm(theForm)"  
　　Action = "javascript:showForm(theForm)" >  
　图书编号:  
　<　INPUT TYPE="TEXT" name="book" value="12345-6789" validator=bookPat >< BR >  
　时 间 值:  
　<　INPUT TYPE="TEXT" name="time" value="12:45" validator=timePat2 > < BR >  
　单击按钮提交表单：  
　<　INPUT TYPE = "SUBMIT" >  
　<　/FORM >  
　<　/BODY >< /HTML >