

js 验证表单大全

不错的 JS 验证

用途：校验 ip 地址的格式

输入：strIP：ip 地址

返回：如果通过验证返回 true,否则返回 false;

***/**

```
function isIP(strIP) {
if (isNull(strIP)) return false;
var re=/^(\d+)\.(\d+)\.(\d+)\.(\d+)/g //匹
配 IP 地址的正则表达式
if(re.test(strIP))
{
if( RegExp.$1 <256 && RegExp.$2<256
&& RegExp.$3<256 &&
RegExp.$4<256) return true;
}
return false;
}
/*
```

用途：检查输入字符串是否为空或者全部都是空格

输入：str

返回：

如果全是空返回 true,否则返回 false

*/

```
functionisNull( str ){
  if ( str == "" ) return true;
  var regu = "^[ ]+$";
  var re = new RegExp(regu);
  return re.test(str);
}
```

/*

用途：检查输入对象的值是否符合整数格式

输入：str 输入的字符串

返回：如果通过验证返回 true,否则返回 false

*/

```
functionisInteger( str ){
  var regu = /^[ -]{0,1}[0-9]{1,}$/;
  return regu.test(str);
}
```

```
/*
```

用途： 检查输入手机号码是否正确
输入：

s: 字符串

返回：

如果通过验证返回 true,否则返回 false

```
*/
```

```
function checkMobile( s ){
var regu =/^([1][3][0-9]{9});;
var re = new RegExp(regu);
if (re.test(s)) {
return true;
} else{
return false;
}
}
```

```
/*
```

用途： 检查输入字符串是否符合正整数
格式

输入：

s: 字符串

返回：

如果通过验证返回 true,否则返回 false

```
/*
function isNumber( s ){
var regu = "^[0-9]+$";
var re = new RegExp(regu);
if (s.search(re) != -1) {
return true;
} else {
return false;
}
}
*/
*/
```

用途：检查输入字符串是否是带小数的数字格式,可以是负数

输入：

s: 字符串

返回：

如果通过验证返回 true,否则返回 false

```
/*
*/
```

```
function isDecimal( str ){
if(isInteger(str)) return true;
var re = /^[-]{0,1}(\d+)[\.]+(\d+)/;
if (re.test(str)) {
if(RegExp.$1==0&&RegExp.$2==0)
```

```
return false;
return true;
} else {
return false;
}
}
/*
*/
```

用途：检查输入对象的值是否符合端口号格式

输入：str 输入的字符串
返回：如果通过验证返回 true,否则返回 false
*/

```
function isPort( str ){
return (isNumber(str) && str<65536);
}
/*
*/
```

用途：检查输入对象的值是否符合 E-Mail 格式

输入：str 输入的字符串
返回：如果通过验证返回 true,否则返回 false
*/

```
function isEmail( str ){
```

```
var myReg = /^[_A-Za-z0-9]+@[_A-
Za-z0-9]+\.\)+[A-Za-z0-9]{2,3}\$/;
if(myReg.test(str)) return true;
return false;
}
/*

```

用途：检查输入字符串是否符合金额格式

格式定义为带小数的正数，小数点后最多三位

输入：

s: 字符串

返回：

如果通过验证返回 **true**,否则返回 **false**

*/

```
function isMoney( s ){
var regu = "^[0-9]+\.\)[0-9]{0,3}\$";
var re = new RegExp(regu);
if (re.test(s)) {
return true;
} else {
return false;
}
}
```

```
/*
```

用途：检查输入字符串是否只由英文字母和数字和下划线组成

输入：

s: 字符串

返回：

如果通过验证返回 true,否则返回 false

```
*/
```

```
function isNumberOr_Letter( s ){//判断  
是否是数字或字母
```

```
var regu = "^[0-9a-zA-Z\_]+$";
```

```
var re = new RegExp(regu);
```

```
if (re.test(s)) {
```

```
return true;
```

```
}else{
```

```
return false;
```

```
}
```

```
}
```

```
/*
```

用途：检查输入字符串是否只由英文字母和数字组成

输入：

s: 字符串

返回：

如果通过验证返回 **true**,否则返回 **false**

***/**

function isNumberOrLetter(s){ //判断

是否是数字或字母

var regu = "^[0-9a-zA-Z]+\$";

var re = new RegExp(regu);

if (re.test(s)) {

return true;

}else{

return false;

}

}

/*

用途： 检查输入字符串是否只由汉字、
字母、 数字组成

输入：

value: 字符串

返回：

如果通过验证返回 **true**,否则返回 **false**

***/**

function isChinaOrNumbOrLett(s){ //

判断是否是汉字、字母、 数字组成

```
var regu = "^[0-9a-zA-Z\u4e00-\u9fa5]+$";
var re = new RegExp(regu);
if (re.test(s)) {
    return true;
} else{
    return false;
}
}
/*
*/
```

用途：判断是否是日期

输入：date: 日期; fmt: 日期格式
返回：如果通过验证返回 true,否则返回 false
*/

```
function isDate( date, fmt ) {
    if (fmt==null) fmt="yyyyMMdd";
    var yIndex = fmt.indexOf("yyyy");
    if(yIndex== -1) return false;
    var year =
        date.substring(yIndex,yIndex+4);
    var mIndex = fmt.indexOf("MM");
    if(mIndex== -1) return false;
    var month =
```

```
date.substring(mIndex,mIndex+2);
var dIndex = fmt.indexOf("dd");
if(dIndex==1) return false;
var day =
date.substring(dIndex,dIndex+2);
if(!isNumber(year)||year>"2100" ||
year< "1900") return false;
if(!isNumber(month)||month>"12" ||
month< "01") return false;
if(day>getMaxDay(year,month) || day<
"01") return false;
return true;
}
function getMaxDay(year,month) {
if(month==4||month==6||month==9||mo
nth==11)
return "30";
if(month==2)
if(year%4==0&&year%100!=0 ||
year%400==0)
return "29";
else
return "28";
return "31";
}
```

```
/*
```

用途：字符 1 是否以字符串 2 结束

输入：str1：字符串； str2：被包含的字符串

返回：如果通过验证返回 true，否则返回 false

```
*/
```

```
function isLastMatch(str1,str2)
```

```
{
```

```
var index = str1.lastIndexOf(str2);
```

```
if(str1.length==index+str2.length)
```

```
return true;
```

```
return false;
```

```
}
```

```
/*
```

用途：字符 1 是否以字符串 2 开始

输入：str1：字符串； str2：被包含的字符串

返回：如果通过验证返回 true，否则返回 false

```
*/
```

```
function isFirstMatch(str1,str2)
```

```
{
```

```
var index = str1.indexOf(str2);
if(index==0) return true;
return false;
}
/*
```

用途：字符 1 是包含字符串 2

输入：str1：字符串； str2：被包含的字符串

返回：如果通过验证返回 true，否则返回 false

```
*/
```

```
function isMatch(str1,str2)
{
var index = str1.indexOf(str2);
if(index==1) return false;
return true;
}
```

```
/*
```

用途：检查输入的起止日期是否正确，规则为两个日期的格式正确，且结束日期 \geq 起始日期

输入：

startDate：起始日期，字符串

endDate: 结束日期，字符串
返回：
如果通过验证返回 **true**,否则返回 **false**

*/

```
function
checkTwoDate( startDate,endDate ) {
if( !isDate(startDate) ) {
alert("起始日期不正确!");
return false;
} else if( !isDate(endDate) ) {
alert("终止日期不正确!");
return false;
} else if( startDate > endDate ) {
alert("起始日期不能大于终止日期!");
return false;
}
return true;
}
```

用途：检查输入的 **Email** 信箱格式是否
正确

输入：

strEmail: 字符串

返回：

如果通过验证返回 **true**,否则返回 **false**

*/

```
function checkEmail(strEmail) {  
//var emailReg = /^[_a-zA-Z0-9]+@[(_a-zA-Z0-  
9]+\.)+[a-zA-Z0-9]{2,3}\$/;  
var emailReg = /^[\w-]+(\.[\w-]+)*@[^\w-]+(\.[\w-]+)+\$/;  
if( emailReg.test(strEmail) ){  
return true;  
}else{  
alert("您输入的 Email 地址格式不正  
确！ ");  
return false;  
}  
}  
/*
```

用途： 检查输入的电话号码格式是否正
确

输入：

strPhone: 字符串

返回：

如果通过验证返回 **true**,否则返回 **false**

*/

```
function checkPhone( strPhone ) {  
    var phoneRegWithArea = /^[0][1-  
9]{2,3}-[0-9]{5,10}$/;  
    var phoneRegNoArea = /^[1-9]{1}[0-  
9]{5,8}$/;  
    var prompt = "您输入的电话号码不正  
确!"  
    if( strPhone.length > 9 ) {  
        if( phoneRegWithArea.test(strPhone) ){  
            return true;  
        }else{  
            alert( prompt );  
            return false;  
        }  
    }  
    else{  
        if( phoneRegNoArea.test( strPhone ) ){  
            return true;  
        }else{  
            alert( prompt );  
            return false;  
        }  
    }  
}
```

```
/*
```

用途： 检查复选框被选中的数目

输入：

checkboxID： 字符串

返回：

返回该复选框中被选中的数目

```
*/
```

```
function checkSelect( checkboxID ) {  
    var check = 0;  
    var i=0;  
    if( document.all(checkboxID).length > 0 )  
    {  
        for( i=0;  
            i<document.all(checkboxID).length;  
            i++ ) {  
            if( document.all(checkboxID).item( i ).c  
                hecked ) {  
                check += 1;  
            }  
        }  
    }else{  
        if( document.all(checkboxID).checked )
```

```
check = 1;
}
return check;
}
function getTotalBytes(varField) {
if(varField == null)
return -1;
var totalCount = 0;
for (i = 0; i< varField.value.length; i++)
{
if (varField.value.charCodeAt(i) > 127)
totalCount += 2;
else
totalCount++ ;
}
return totalCount;
}
function
getFirstSelectedValue( checkboxID ){
var value = null;
var i=0;
if( document.all(checkboxID).length >
0 ){
for( i=0;
i<document.all(checkboxID).length;
```

```
i++ ){  
if( document.all.checkboxID.item(i).checked ){  
value =  
document.all.checkboxID.item(i).value;  
break;  
}  
}  
}  
} else {  
if( document.all.checkboxID.checked )  
value = document.all.checkboxID.value;  
  
}  
return value;  
}
```

```
function  
getFirstSelectedIndex( checkboxID ){  
var value = -2;  
var i=0;  
if( document.all.checkboxID.length >  
0 ){  
for( i=0;  
i<document.all.checkboxID.length;  
i++ ) {
```

```
if( document.all.checkboxID).item( i ).checked ) {  
    value = i;  
    break;  
}  
}  
}  
} else {  
    if( document.all.checkboxID).checked )  
        value = -1;  
    }  
return value;  
}  
function selectAll( checkboxID,status ){  
    if( document.all.checkboxID) == null)  
        return;  
    if( document.all.checkboxID).length >  
        0 ){  
        for( i=0;  
            i<document.all.checkboxID).length;  
            i++ ){  
            document.all.checkboxID).item( i ).checked = status;  
        }  
    } else {  
        document.all.checkboxID).checked =
```

```
status;  
}  
}  
function selectInverse( checkboxID ) {  
if( document.all(checkboxID) == null)  
return;  
if( document.all(checkboxID).length > 0 )  
{  
for( i=0;  
i<document.all(checkboxID).length;  
i++ ) {  
document.all(checkboxID).item( i ).checked  
= !document.all(checkboxID).item( i ).checked;  
}  
}  
}  
function checkDate( value ) {  
if(value=='') return true;  
if(value.length!=8 || !isNumber(value))  
return false;
```

```
var year = value.substring(0,4);
if(year>"2100" || year< "1900")
return false;
var month = value.substring(4,6);
if(month>"12" || month< "01") return
false;
var day = value.substring(6,8);
if(day>getMaxDay(year,month) || day<
"01") return false;
return true;
}
/*
```

用途： 检查输入的起止日期是否正确，
规则为两个日期的格式正确或都为空
且结束日期>=起始日期

输入：

startDate: 起始日期，字符串

endDate: 结束日期，字符串

返回：

如果通过验证返回 **true**,否则返回 **false**

*/

function

checkPeriod(startDate,endDate) {
if(!checkDate(startDate)) {

```
    alert("起始日期不正确!");
    return false;
} else if( !checkDate(endDate) ) {
    alert("终止日期不正确!");
    return false;
} else if( startDate > endDate ) {
    alert("起始日期不能大于终止日期!");
    return false;
}
return true;
}
/*
```

用途： 检查证券代码是否正确

输入：

secCode:证券代码

返回：

如果通过验证返回 **true**,否则返回 **false**

*/

```
function checkSecCode( secCode ) {
if( secCode.length !=6 ){
    alert("证券代码长度应该为 6 位");
    return false;
}
if(!isNumber( secCode )){
```

```
alert("证券代码只能包含数字");
```

```
return false;
```

```
}
```

```
return true;
```

```
}
```

```
*****
```

```
*****
```

function:cTrim(sInputString,iType)

description:字符串去空格的函数

parameters:iType: 1=去掉字符串左边的空格

2=去掉字符串左边的空格

0=去掉字符串左边和右边的空格

return value:去掉空格的字符串

```
*****
```

```
******/
```

function cTrim(sInputString,iType)

```
{
```

```
var sTmpStr = ' ';
```

```
var i = -1;
```

```
if(iType == 0 || iType == 1)
```

```
{
```

```
while(sTmpStr == ' ')
```

```
{  
++i;  
sTmpStr = sInputString.substr(i,1);  
}  
sInputString = sInputString.substring(i);  
  
}  
if(iType == 0 || iTyPe == 2)  
{  
sTmpStr = ' ';  
i = sInputString.length;  
while(sTmpStr == ' ')  
{  
--i;  
sTmpStr = sInputString.substr(i,1);  
}  
sInputString =  
sInputString.substring(0,i+1);  
}  
return sInputString;  
}  
/*
```