1.非负整数 /^\d+$/  
2.正整数 /^[0-9]\*[1-9][0-9]\*$/  
3.非正整数 /^((-\d+)|(0+))$/  
4.负整数 /^-[0-9]\*[1-9][0-9]\*$/  
5.整数 /^-?\d+$/  
6.非负浮点数 /^\d+(\.\d+)?$/  
7.正浮点数 /^(([0-9]+\.[0-9]\*[1-9][0-9]\*)|([0-9]\*[1-9][0-9]\*\.[0-9]+)|([0-9]\*[1-9][0-9]\*))$/  
8.非正浮点数 /^((-\d+(\.\d+)?)|(0+(\.0+)?))$/  
9.负浮点数 /^(-(([0-9]+\.[0-9]\*[1-9][0-9]\*)|([0-9]\*[1-9][0-9]\*\.[0-9]+)|([0-9]\*[1-9][0-9]\*)))$/  
10.浮点数 /^(-?\d+)(\.\d+)?$/  
11.数字 /^\d+(\.{1}\d+)?$/  
12.由26个英文字母组成的字符串 /^[A-Za-z]+$/  
13.由26个英文字母的大写组成的字符串 /^[A-Z]+$/  
14.由26个英文字母的小写组成的字符串 /^[a-z]+$/  
15.由数字和26个英文字母组成的字符串 /^[A-Za-z0-9]+$/  
16.由数字、26个英文字母或者下划线组成的字符串 /^\w+$/  
17.匹配所有单字节长度的字符组成的字符串 /^[\x00-\xff]+$/  
18.匹配所有双字节长度的字符组成的字符串 /^[^\x00-\xff]+$/  
19.字符串是否含有双字节字 /[^\x00-\xff]+/  
20.email地址 /^[\w-]+(\.[\w-]+)\*@[\w-]+(\.[\w-]+)+$/  
或者 /w+([-+.]w+)\*@w+([-.]w+)\*.w+([-.]w+)\*/  
21.url地址 /^[a-zA-z]+://(\w+(-\w+)\*)(\.(\w+(-\w+)\*))\*(\?\S\*)?$/  
或者 /http://([w-]+.)+[w-]+(/[w- ./?%&=]\*)?/  
22.匹配中文字符的正则 /[u4e00-u9fa5]/  
23.匹配双字节字符(包括汉字在内) /[^x00-xff]/  
应用：计算字符串的长度(一个双字节字符长度计2，ASCII字符计1)  
String.prototype.len=function(){  
return this.replace([^x00-xff]/g,”aa”).length;  
}  
24.匹配空行的正则 /n[s| ]\*r/  
25.匹配HTML标记的正则 /<(.\*)>.\*</1>|<(.\*) />/  
26.匹配首尾空格的正则 /(^s\*)|(s\*$)/  
应用：javascript中没有像vbscript那样的trim函数，我们就可以利用这个表达式来实现，如下：  
String.prototype.trim = function(){  
return this.replace(/(^s\*)|(s\*$)/g, “”);  
}  
27.匹配IP地址的正则 /(d+).(d+).(d+).(d+)/  
应用：利用正则表达式匹配IP地址，并将IP地址转换成对应数值的Javascript程序：  
function IP2V(ip){  
re=/(d+).(d+).(d+).(d+)/g;  
if(re.test(ip)){  
return RegExp.$1\*Math.pow(255,3))+  
RegExp.$2\*Math.pow(255,2))+  
RegExp.$3\*255+RegExp.$4\*1;  
}  
else{  
throw new Error(“Not a valid IP address!”);  
}  
}  
其实直接用split函数来分解可能更简单，程序如下：  
var ip=”10.100.20.168″;  
ip=ip.split(“.”);  
alert(“IP值是：”+(ip[0]\*255\*255\*255+ip[1]\*255\*255+ip[2]\*255+ip[3]\*1));  
28.去除字串中重复的字符的javascript程序  
var s=”abacabefgeeii”;  
var s1=s.replace(/(.).\*1/g,”$1″);  
var re=new RegExp(“["+s1+"]“,”g”);  
var s2=s.replace(re,”");  
alert(s1+s2); //结果为：abcefgi  
/\*使用后向引用取出包括重复的字符，再以重复的字符建立第二个表达式，取到不重复的字符，  
两者串连。这个方法对于字符顺序有要求的字符串可能不适用。\*/  
29.用正则表达式从URL地址中提取文件名的javascript程序  
s=”http://www.9499.net/page1.htm“;  
s=s.replace(/(.\*/){0,}([^.]+).\*/ig,”$2″);  
alert(s); //结果为page1  
30.限制表单文本框输入内容  
只能输入中文：  
onkeyup=”value=value.replace(/[^u4E00-u9FA5]/g,”)”  
onbeforepaste=”clipboardData.setData(‘text’,  
clipboardData.getData(‘text’).replace(/[^u4E00-u9FA5]/g,”))”  
只能输入全角字符：  
onkeyup=”value=value.replace(/[^uFF00-uFFFF]/g,”)”  
onbeforepaste=”clipboardData.setData(‘text’,  
clipboardData.getData(‘text’).replace(/[^uFF00-uFFFF]/g,”))”  
只能输入数字：  
onkeyup=”value=value.replace(/[^d]/g,”)”  
onbeforepaste=”clipboardData.setData(‘text’,  
clipboardData.getData(‘text’).replace(/[^d]/g,”))”  
只能输入数字和英文：  
onkeyup=”value=value.replace(/[W]/g,”)”  
onbeforepaste=”clipboardData.setData(‘text’,  
clipboardData.getData(‘text’).replace(/[^d]/g,”))”  
31.验证文件名由字母，数字，下滑线组成 /^((\w+)(\.{1})(\w+))$/  
32.匹配日期(1900-1999)  
/^19\d{2}-((0[1-9])|(1[0-2]))-((0[1-9])|([1-2][0-9])|(3([0|1])))$/  
33.匹配日期(2000-2999)  
/^20\d{2}-((0[1-9])|(1[0-2]))-((0[1-9])|([1-2][0-9])|(3([0|1])))$/  
34.匹配日期时间  
/^(1|2\d{3}-((0[1-9])|(1[0-2]))-((0[1-9])|([1-2][0-9])|(3([0|1]))))( (\d{2}):(\d{2}):(\d{2}))?$/  
网页验证的例子：  
<SCRIPT language=”javascript”><!–  
function validate(){  
var frm = document.form1;  
if ( frm.orgname.value==”" ){  
alert(“机构名称必须填写,请输入!”);  
frm.orgname.focus();  
return false;  
}  
if ( frm.orgfunction.value==”" ){  
alert(“机构职能必须填写,请输入!”);  
frm.orgfunction.focus();  
return false;  
}  
if ( frm.orgpostcode.value==”" ){…}  
else{  
var pattern=/^d…{6}$/;  
if(pattern.test(frm.orgpostcode.value)){…}  
else{  
alert(“输入的邮政编码不是合法的格式!”);  
frm.orgpostcode.focus();  
return false;  
}  
}  
if ( frm.orgphone.value==”" ){…}  
else{  
var str=frm.orgphone.value;  
var reg=/(^[0-9]…{3,4}-[0-9]…{3,8}$)|(^[0-9]…{3,8}$)|(^([0-9]…{3,4})[0-9]…{3,8}$)|(^0…{0,1}13[0-9]…{9}$)/;  
if (isNaN(str)){  
if(reg.test(str)==false){  
alert(“电话号码输入有误,请重新输入!”);  
frm.orgphone.value=”";  
frm.orgphone.focus();  
return false;  
}  
}  
else if(str.length<6){  
alert(“电话输入有误,请重新输入!”);  
frm.orgphone.focus();  
return false;  
}  
}  
if ( frm.orgfax.value==”" ){…}  
else{  
var str=frm.orgfax.value;  
var reg=/(^[0-9]…{3,4}-[0-9]…{3,8}$)|(^[0-9]…{3,8}$)|(^([0-9]…{3,4})[0-9]…{3,8}$)|(^0…{0,1}13[0-9]…{9}$)/;  
if (isNaN(str)){  
if(reg.test(str)==false){  
alert(“传真输入有误,请重新输入!”);  
frm.orgfax.value=”";  
frm.orgfax.focus();  
return false;  
}  
}  
else if(str.length<6){  
alert(“传真输入有误,请重新输入!”);  
frm.orgfax.focus();  
return false;  
}  
}  
if ( frm.orgemail.value==”" ){…}  
else{  
var pattern=/^w+((-w+)|(.w+))\*@[A-Za-z0-9]+((.|-)[A-Za-z0-9]+)\*.[A-Za-z0-9]+$/;  
if(pattern.test(frm.orgemail.value)){…}  
else{  
alert(“输入的电子邮箱不是合法的格式!”);  
frm.orgemail.focus();  
return false;  
}  
}  
if ( frm.orgurl.value==”" ){…}  
else{  
var pattern=new RegExp(“((^http)|(^https)|(^ftp))://(\w)+.(\w)+”);  
if(pattern.test(frm.orgurl.value)){…}  
else{  
alert(“输入的URL不是合法的格式!”);  
frm.orgurl.focus();  
return false;  
}  
}  
}  
//–></SCRIPT>