**一、**

**不想含某些字符串的空行** ^((?!不想包含的字符串).)\*$**常见正则表达式**

　　匹配中文字符的正则表达式： [u4e00-u9fa5]     
　　评注：匹配中文还真是个头疼的事，有了这个表达式就好办了   
　　匹配双字节字符(包括汉字在内)：[^x00-xff]   
　　评注：可以用来计算字符串的长度（一个双字节字符长度计2，ASCII字符计1）   
　　匹配空白行的正则表达式：ns\*r   
　　评注：可以用来删除空白行   
　　匹配HTML标记的正则表达式：<(S\*?)[^>]\*>.\*?|<.\*? />   
　　评注：网上流传的版本太糟糕，上面这个也仅仅能匹配部分，对于复杂的嵌套标记依旧无能为力   
　　匹配首尾空白字符的正则表达式：^s\*|s\*$   
　　评注：可以用来删除行首行尾的空白字符(包括空格、制表符、换页符等等)，非常有用的表达式   
　　匹配Email地址的正则表达式：w+([-+.]w+)\*@w+([-.]w+)\*.w+([-.]w+)\*   
　　评注：表单验证时很实用   
　　匹配网址URL的正则表达式：^(http|https):\/\/[\w\-\_]+(\.[\w\-\_]+)+([\w\-\.,@?^=%&amp;:/~\+#]\*[\w\-\@?^=%&amp;/~\+#])?$  
　　评注：网上流传的版本功能很有限，上面这个基本可以满足需求 

     匹配帐号是否合法(字母开头，允许5-16字节，允许字母数字下划线)：^[a-zA-Z][a-zA-Z0-9\_]{4,15}$   
　　评注：表单验证时很实用   
　　匹配国内电话号码：d{3}-d{8}|d{4}-d{7}   
　　评注：匹配形式如 0511-4405222 或 021-87888822   
　　匹配腾讯QQ号：[1-9][0-9]{4,}   
　　评注：腾讯QQ号从10000开始   
　　匹配中国邮政编码：[1-9]d{5}(?!d)   
　　评注：中国邮政编码为6位数字   
　　匹配身份证：d{15}|d{18}   
　　评注：中国的身份证为15位或18位   
　　匹配ip地址：d+.d+.d+.d+   
　　评注：提取ip地址时有用   
　　匹配特定数字：   
　　^[1-9]d\*$　 　 //匹配正整数   
　　^-[1-9]d\*$ 　 //匹配负整数   
　　^-?[1-9]d\*$　　 //匹配整数   
　　^[1-9]d\*|0$　 //匹配非负整数（正整数 + 0）   
　　^-[1-9]d\*|0$　　 //匹配非正整数（负整数 + 0）   
　　^[1-9]d\*.d\*|0.d\*[1-9]d\*$　　 //匹配正浮点数   
　　^-([1-9]d\*.d\*|0.d\*[1-9]d\*)$　 //匹配负浮点数   
　　^-?([1-9]d\*.d\*|0.d\*[1-9]d\*|0?.0+|0)$　 //匹配浮点数   
　　^[1-9]d\*.d\*|0.d\*[1-9]d\*|0?.0+|0$　　 //匹配非负浮点数（正浮点数 + 0）   
　　^(-([1-9]d\*.d\*|0.d\*[1-9]d\*))|0?.0+|0$　　//匹配非正浮点数（负浮点数 + 0）   
　　评注：处理大量数据时有用，具体应用时注意修正   
　　匹配特定字符串：   
　　^[A-Za-z]+$　　//匹配由26个英文字母组成的字符串   
　　^[A-Z]+$　　//匹配由26个英文字母的大写组成的字符串   
　　^[a-z]+$　　//匹配由26个英文字母的小写组成的字符串   
　　^[A-Za-z0-9]+$　　//匹配由数字和26个英文字母组成的字符串   
　　^w+$　　//匹配由数字、26个英文字母或者下划线组成的字符串   
　　在使用RegularExpressionValidator验证控件时的验证功能及其验证表达式介绍如下:   
　　只能输入数字：“^[0-9]\*$”   
　　只能输入n位的数字：“^d{n}$”   
　　只能输入至少n位数字：“^d{n,}$”   
　　只能输入m-n位的数字：“^d{m,n}$”   
　　只能输入零和非零开头的数字：“^(0|[1-9][0-9]\*)$”   
　　只能输入有两位小数的正实数：“^[0-9]+(.[0-9]{2})?$”   
　　只能输入有1-3位小数的正实数：“^[0-9]+(.[0-9]{1,3})?$”   
　　只能输入非零的正整数：“^+?[1-9][0-9]\*$”   
　　只能输入非零的负整数：“^-[1-9][0-9]\*$”   
　　只能输入长度为3的字符：“^.{3}$”   
　　只能输入由26个英文字母组成的字符串：“^[A-Za-z]+$”   
　　只能输入由26个大写英文字母组成的字符串：“^[A-Z]+$”   
　　只能输入由26个小写英文字母组成的字符串：“^[a-z]+$”   
　　只能输入由数字和26个英文字母组成的字符串：“^[A-Za-z0-9]+$”   
　　只能输入由数字、26个英文字母或者下划线组成的字符串：“^w+$”   
　　验证用户密码:“^[a-zA-Z]w{5,17}$”正确格式为：以字母开头，长度在6-18之间， 只能包含字符、数字和下划线。   
　　验证是否含有^%&'',;=?$"等字符：“[^%&'',;=?$x22]+”   
　　只能输入汉字：“^[u4e00-u9fa5],{0,}$”   
　　验证Email地址：“^w+[-+.]w+)\*@w+([-.]w+)\*.w+([-.]w+)\*$”   
　　验证电话号码：“^((d{3,4})|d{3,4}-)?d{7,8}$”   
　　正确格式为：“XXXX-XXXXXXX”，“XXXX-XXXXXXXX”，“XXX-XXXXXXX”，   
　　“XXX-XXXXXXXX”，“XXXXXXX”，“XXXXXXXX”。   
　　验证身份证号（15位或18位数字）：“^d{15}|d{}18$”   
　　验证一年的12个月：“^(0?[1-9]|1[0-2])$”正确格式为：“01”-“09”和“1”“12”   
　　验证一个月的31天：“^((0?[1-9])|((1|2)[0-9])|30|31)$”   
　　正确格式为：“01”“09”和“1”“31”。   
　　匹配中文字符的正则表达式： [u4e00-u9fa5]   
　　匹配双字节字符(包括汉字在内)：[^x00-xff]   
　　匹配空行的正则表达式：n[s| ]\*r   
　　匹配HTML标记的正则表达式：/<(.\*)>.\*|<(.\*) />/   
　　匹配首尾空格的正则表达式：(^s\*)|(s\*$)  

**二、正则表达式应用**

(1)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;} 

(2)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   
　　String.prototype.trim = function()   
　　{   
　　　　return this.replace(/(^s\*)|(s\*$)/g, "");   
　　}

(3)应用：利用正则表达式分解和转换IP地址 

[复制代码](javascript:void(0);)

function IP2V(ip) //IP地址转换成对应数值

{

　　re=/(d+).(d+).(d+).(d+)/g //匹配IP地址的正则表达式

　　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!")

　　}

}

[复制代码](javascript:void(0);)

(4)应用：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") ;//Page1.htm 

(5)应用：利用正则表达式限制网页表单里的文本框输入内容   
　　用正则表达式限制只能输入中文：onkeyup="value=value.replace(/[^u4E00-u9FA5]/g,'') "onbeforepaste="clipboardData.setData(''text'',clipboardData.getData(''text'').replace(/[^u4E00-u9FA5]/g,''))"   
　　用正则表达式限制只能输入全角字符： onkeyup="value=value.replace(/[^uFF00-uFFFF]/g,'') "onbeforepaste="clipboardData.setData(''text'',clipboardData.getData(''text'').replace(/[^uFF00-uFFFF]/g,''))"   
　　用正则表达式限制只能输入数字：onkeyup="value=value.replace(/[^d]/g,'') "onbeforepaste= "clipboardData.setData(''text'',clipboardData.getData(''text'').replace(/[^d]/g,''))"   
　　用正则表达式限制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,''

**三、/i,/g,/ig,/gi,/m**

正则表达式中/i,/g,/ig,/gi,/m的区别和含义

/i (忽略大小写)  
/g (全文查找出现的所有匹配字符)  
/m (多行查找)  
/gi(全文查找、忽略大小写)  
/ig(全文查找、忽略大小写)

站内相关：

http://www.cnblogs.com/xcsn/p/3157456.html   MVC Model

http://www.cnblogs.com/xcsn/p/3435437.html

作者：[心存善念](http://www.cnblogs.com/xcsn)  
本文地址：<https://www.cnblogs.com/xcsn/p/4677605.html>  
欢迎转载，请在明显位置给出出处及链接。