

经典JavaScript正则表达式实战

菜鸟(<http://www.cainiao8.com/>)

邮箱:runningtortoise@hotmail.com

目录

正则表达式实战... 1
匹配结尾的数字... 2
统一空格个数... 3
判断字符串是不是由数字组成... 3
电话号码正则... 3
手机号码正则表达式... 4
使用正则表达式实现删除字符串中的空格: ... 4
限制文本框只能输入数字和小数点等等... 5
替换小数点前内容为指定内容... 5
只匹配中文的正则表达式... 5
返回字符串的中文字符个数... 6
正则表达式取得匹配IP地址前三段... 6
匹配与之间的内容... 7
用正则表达式获得文件名... 7
绝对路径变相对路径... 8
用户名正则... 8
匹配英文地址... 8
正则匹配价格... 9
身份证号码的匹配... 10
要求文本有指定行数... 10
单词首字母大写... 10
正则验证日期格式... 11
去掉文件的后缀名... 12
验证邮箱的正则表达式... 12
匹配源代码中的链接... 13
匹配链接的文字... 14
正则判断标签是否闭合... 14
用正则获得指定标签的内容... 15
正则判断是否为数字与字母的混合... 16
空格与英文同时存在... 17
显示或者保存正则表达式匹配的部分内容... 18
正则表达式替换变量... 18
正则替换指定属性中的文本... 19
将阿拉伯数字替换为中文大写形式... 20
替换文本中的URL为链接... 21
从HTML代码段删除指定标签极其内容... 21
用正则给文本分段... 22
转换源代码中的标签... 22
给属性添加双引号... 23
给table加上tbody. 25
去掉标签的所有属性... 25
正则替换特定单词... 25
指定文字高亮显示... 26
删除标签... 27
错误贴... 28
太难贴... 28

[匹配数字的“正则”. 28](#)
[正则资源... 28](#)

匹配结尾的数字

来源: [如何用 JAVASCRIPT 正则表达式 取出字符串最后一组数字, 谢谢](#)
如

30CAC0040 取出40

3SFASDF92 取出92

正则如下: `/\d+$/g`

统一空格个数

来源: [正则匹配空格的问题](#)

字符串内字符键有空格, 但是空格的数量可能不一致, 通过正则将空格的个数统一变为一个。

例如: 蓝色理想

变成: 蓝色理想

aobert的正则:

```
<script>
var str="蓝 色 理      想"
var reg=/\s+/g
str = str.replace(reg," ")
document.write(str)
</script>
```

判断字符串是不是由数字组成

来源: [有没有简单的方法判断字符串由数字组成?](#)

这个正则比较简单, 写了一个测试

```
<script type="text/javascript">
function isDigit(str){
    var reg = /^*\d*$/;
    return reg.test(str);
}
var str = "7654321";
document.write(isDigit(str));
var str = "test";
document.write(isDigit(str));
</script>
```

电话号码正则

来源: [想问一下关于电话号码的正则判断](#)

: [求一个验证电话号码的JS正则](#)

`/^\d{3,4}-\d{7,8}(-\d{3,4})?$/`

区号必填为3-4位的数字, 区号之后用“-”与电话号码连接

`^\d{3,4}-`

电话号码为7-8位的数字

`\d{7,8}`

分机号码为3-4位的数字, 非必填, 但若填写则以“-”与电话号码相连接

`(-\d{3,4})?`

手机号码正则表达式

正则验证手机号，忽略前面的0，支持130-139，150-159。忽略前面0之后判断它是11位的。

cloeft的正则：

```
/^0*(13|15)\d{9}$/
```

^0*匹配掉开头任意数量的0。

由于手机号码是13任意数字9位，和15任意数字9位，所以可以用(13|15)\d{9}匹配。

测试代码如下：

```
function testReg(reg,str){
    return reg.test(str);
}
var reg = /^0*(13|15)\d{9}$/;
var str = '13889294444';
var str2 = '12889293333';
var str3 = '23445567';
document.write(testReg(reg,str)+'<br />');
document.write(testReg(reg,str2)+'<br />');
document.write(testReg(reg,str3)+'<br />');
```

使用正则表达式实现删除字符串中的空格：

来源：[请问JS中有没有去掉空格的函数](#)

代码以及测试代码如下：

```
<script type="text/javascript">
//删除字符串两侧的空白字符。
function trim(str){
    return str.replace(/^s+|s+$/g,"");
}
//删除字符串左侧的空白字符。
function ltrim(str){
    return str.replace(/^s+/g,"");
}
//删除字符串右侧的空白字符。
function rtrim(str){
    return str.replace(/s+$/g,"");
}
//以下为测试代码
var trimTest = " 123456789 ";//前后各有一个空格。
document.write('length:'+trimTest.length+'<br />');//使用前
document.write('ltrim length:'+ltrim(trimTest).length+'<br />');//使用ltrim后
document.write('rtrim length:'+rtrim(trimTest).length+'<br />');//使用rtrim后
document.write('trim length:'+trim(trimTest).length+'<br />');//使用trim后
</script>
```

测试的结果如下：

```
length:11
ltrim length:10
rtrim length:10
trim length:9
```

限制文本框只能输入数字和小数点等等

来源：[文本框输入限制的问题????](#)

只能输入数字和小数点

```
var reg = /^d*\.\?d{0,2}$/
```

开头有若干个数字，中间有0个或者一个小数点，结尾有0到2个数字。

只能输入小写的英文字母和小数点，和冒号，正反斜杠(: ./\)

```
var reg = /[a-z\.\V\\:]+/;
```

a-z包括了小写的英文字母，\.是小数点，\V和\\分别是左右反斜线，最后是冒号。整个组成一个字符集和代码任一均可，最后在加上+，1或者多个。

替换小数点前内容为指定内容

来源: [求一正则表达式!](#)

请问 怎么把这个字符串的小数点前面的字符替换为我自定义的字符串啊?

例如: infomarket.php?id=197 替换为 test.php?id=197

应该可以把第一个点"."之前的所有单词字符替换为test就可以了。我写的正则如下:

```
<script type="text/javascript">
var str = "infomarket.php?id=197";
var reg = /^w*/ig;
//匹配字符串开头的任意个单词字符
str = str.replace(reg,'test');
document.write(str);
</script>
```

原帖的有点复杂，没太看明白。

只匹配中文的正则表达式

来源: [关于上传的时候，只能选择中文名称的图片（是否可以用TOASCII方法呢？）](#)

前两天看的《JavaScript开发王》里恰好有中文的unicode范围，正则如下:

```
/[\u4E00-\u9FA5\uF900-\uFA2D]/
```

写了一个简单的测试，会把所有的中文替换成“哦”。

```
<script type="text/javascript">
var str = "有中文?and English.";
var reg = /[\u4E00-\u9FA5\uF900-\uFA2D]/ig;
str = str.replace(reg,'哦');
document.write(str);
</script>
```

返回字符串的中文字符个数

来源: [有没返回中文字符字节的函数!](#)

一般的字符长度对中文和英文都是不分别的 如JS里的length，那么如何返回字符串中中文字符的个数呢? guoshuang老师在原帖中给出了解决方案，我又没看懂.....

不过我自己也想到了一个办法: 先去掉非中文字符，再返回length属性。函数以及测试代码如下:

```
<script type="text/javascript">
function cLength(str){
    var reg = /^[^\u4E00-\u9FA5\uF900-\uFA2D]/g;
    //匹配非中文的正则表达式
    var temp = str.replace(reg,"");
    return temp.length;
}
var str = "中文123";
document.write(str.length+'<br />');
document.write(cLength(str));
</script>
```

结果:

5
2

中文两个，数字三个，正确。

下面的测试也正确。

```
var str = "中文123tets@#!#%$#[[]{}";  
document.write(str.length+'<br />');  
document.write(cLength(str));
```

正则表达式取得匹配**IP**地址前三段

来源: [如何用正则取IP前3段](#)

192.168.118.101, 192.168.118.72, 192.168.118.1都替换成: 192.168.118

只要匹配掉最后一段并且替换为空字符串就行了，正则如下：

```
/\.\d{1,3}$/
```

匹配结尾的.n,.nn或者.nnn。

测试代码如下：

```
function replaceReg(reg,str){  
    return str.replace(reg,"")  
}  
var reg = /\.\d{1,3}$/;  
var str = '192.168.118.101';  
var str2 = '192.168.118.72';  
var str3 = '192.168.118.1';  
document.write(replaceReg(reg,str)+'<br />');  
document.write(replaceReg(reg,str2)+'<br />');  
document.write(replaceReg(reg,str3)+'<br />');
```

相似的有，这个帖子里有一个验证IP地址的方法: [求检验MAC地址的正则表达例子](#)

匹配与之间的内容

来源: [请教个正则的小问题吧](#)

```
<ul>safsf<ul>safsf</ul><ul>safsf</ul></ul>
```

用正则可以得到 起到下个 之间的内容。

正则如下：

```
/<ul>[\s\S]+?<ul>/i
```

首先匹配两侧的ul标签，中间的[\s\S]+?可以匹配一个或者多个任意字符，一定要非贪婪，否则会匹配safsfsafsf。

用正则表达式获得文件名

来源: [C:\006.JPG](#)

c:\images\tupian\006.jpg

可能是直接在盘符根目录下，也可能在好几层目录下，要求替换到只剩文件名。

xlez的正则如下：

```
/[^\V]*[\\V]+/g
```

首先匹配非左右斜线字符0或多个，然后是左右斜线一个或者多个。形如"xxx/"或者"xxx\"或者"/"或者"

函数以及测试代码：

```
<script type="text/javascript">  
function getFileName(str){  
    var reg = /[^\V]*[\\V]+/g;  
    //xxx\或者是xxx/  
    str = str.replace(reg,"");  
    return str;  
}
```

```
var str = "c:\\images\\tupian\\006.jpg";
document.write(getFileName(str)+'<br />');
var str2 = "c:/images/tupian/test2.jpg";
document.write(getFileName(str2));
</script>
```

注意，\需要转义。

绝对路径变相对路径

来源: [讨论一个正则](#)

将转换为: 。

其中网址可能改变, 例如http://localhost等等。

cloudchen的正则:

/http:VV[^V]+/

首先是http://, 然后[^V]+找过1个或者多个非/字符, 因为遇到第一个/表示已经到目录了, 停止匹配。

测试代码如下:

```
<script type="text/javascript">
var str = '<IMG height="120" width="800" src="http://23.123.22.12/image/
somepic.gif">';
var reg = /http:VV[^V]+/;
str = str.replace(reg,"");
alert(str)
</script>
```

用户名正则

来源: [求正则，急急呀！！](#)

用于用户名注册, , 用户名只 能用 中文、英文、数字、下划线、4-16个字符。

hansir和解决方案弄成正则:

/^[\\u4E00-\\u9FA5\\uf900-\\ufa2d\\w]{4,16}\$/

中文字符或者单词字符, 4到16个。实现4到16结成到正则里的关键就是开始^和结束\$, 这就等于整个字符串只能有这些匹配的内容, 不能有多余的。

函数和测试代码如下:

```
<script type="text/javascript">
function isEmail(str){
    var reg = /^[\\u4E00-\\u9FA5\\uf900-\\ufa2d\\w]{4,16}$/;
    return reg.test(str);
}
var str = '超级无敌用户名regExp';
var str2 = '捣乱的@';
var str3 = '太短'
var str4 = '太长longlonglonglonglonglonglonglong'
document.write(isEmail(str)+'<br />');
document.write(isEmail(str2)+'<br />');
document.write(isEmail(str3)+'<br />');
document.write(isEmail(str4)+'<br />');
</script>
```

匹配英文地址

来源: [-求助- 正则问题](#)

规则如下:

包含 "点", "字母", "空格", "逗号", "数字", 但开头和结尾不能是除字母外任何字符。

`[\.a-zA-Z\s,0-9]`这个字符集就实现了字母, 空格, 逗号和数字。最终正则如下:

`/^[a-zA-Z][\.a-zA-Z\s,0-9]*?[a-zA-Z]+$`

开头必须有字母, 结束也必须是一个以上字母。测试代码如下:

```
<script type="text/javascript">
function testReg(reg,str){
    return reg.test(str);
}
var reg = /^[a-zA-Z][\.a-zA-Z\s,0-9]*?[a-zA-Z]+$/;
var str = 'No.8,ChangAn Street,BeiJing,China';
var str2 = '8.No,ChangAn Street,BeiJing,China';
var str3 = 'No.8,ChangAn Street,BeiJing,China88';
document.write(testReg(reg,str)+'<br />')
document.write(testReg(reg,str2)+'<br />')
document.write(testReg(reg,str3)+'<br />')
</script>
```

正则匹配价格

来源: [为什么这个正则不起作用啊? 急啊?](#)

价格的格式应该如下:

开头数字若干位, 可能有一个小数点, 小数点后面可以有两位数字。hansir给出的对应正则如下:

`/^(\d*\.\d{0,2}|\d+).*$/`

hansir给出的测试代码如下:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>无标题文档</title>
<script type="text/javascript">
function checkPrice(me){
    if(!(/^(\d*\.\d{0,2}|\d+)$/.test(me.value))){
        me.value = me.value.replace(/^(\d*\.\d{0,2}|\d+).*$/, '$1');
    }
}
</script>
</head>
<body>
<input type="text" onkeyup="checkPrice(this);"/>
</body>
</html>
```

身份证号码的匹配

来源: [关于正则的,大家帮帮忙,急,在线等](#)

身份证号码可以是15位或者是18位, 其中最后一位可以是X。其它全是数字, 正则如下:

`/^(\d{14}|\d{17})(\d|[xX])$/`

开头是14位或者17位数字, 结尾可以是数字或者是x或者是X。

测试代码如下:

```
<script type="text/javascript">
function testReg(reg,str){
    return reg.test(str);
}
}
```

```

var reg = /^(d{14}|d{17})(d|[xX])$/;
var str = '123456789012345';//15位
var str2 = '123456789012345678';//18位
var str3 = '12345678901234567X';//最后一位是X
var str4 = '1234';//位数不对
document.write(testReg(reg,str)+'<br />');
document.write(testReg(reg,str2)+'<br />');
document.write(testReg(reg,str3)+'<br />');
document.write(testReg(reg,str4)+'<br />');
</script>

```

要求文本有指定行数

来源: [\[求助\]求一句正则表达式的写法](#)

匹配至少两行的字符串，每行都为非空字符。

只要匹配到[\n\r]就表示有换行了，再保证换行的两段都不是空字符就可以了。正则如下：

```
/\S+?[\n\r]\S+?/i
```

这个正则的应用应该是在用在textarea里，如果是如下要求：可以支持所有字符，中间可带空格，可以包括英文、数字、中文、标点

这样的话，只要针对空格再改一下就行了。（按照非空的要求，上面有不能匹配“字符+空格+换行+字符”的字符串）。修改如下：

```
/\S+?\s*?[\n\r]\s*?\S+?/i
```

单词首字母大写

来源: [求个正则，处理英文单词或词组的](#)

每单词首字大写，其他小写。如blue idea转换为Blue Idea，BLUE IDEA也转换为Blue Idea

cloeft的正则：

```
/\b(\w)|\s(\w)/g
```

所谓“首字母”包括两种情况：第一种是边界(开头)的单词字符，一种是空格之后的新单词的第一个字母。测试代码如下：

```

<script type="text/javascript">
function replaceReg(reg,str){
    str = str.toLowerCase();
    return str.replace(reg,function(m){return m.toUpperCase()})
}
var reg = /\b(\w)|\s(\w)/g;
var str = 'blue idea';
var str2 = 'BLUE IDEA';
var str3 = 'Test \n str is no good!';
var str4 = 'final test';
document.write(replaceReg(reg,str)+'<br />');
document.write(replaceReg(reg,str2)+'<br />');
document.write(replaceReg(reg,str3)+'<br />');
document.write(replaceReg(reg,str4)+'<br />');
</script>

```

正则验证日期格式

来源: [YYYY-MM-DD的正则怎样写啊？](#)

yyyy-mm-dd格式

正则如下：

```
/^\d{4}-\d{1,2}-\d{1,2}$/
```

4位数字，横线，1或者2位数字，再横线，最后又是1或者2位数字。

测试代码如下：

```
<script type="text/javascript">
function testReg(reg,str){
    return reg.test(str);
}
var reg = /^\\d{4}-\\d{1,2}-\\d{1,2}$/;
var str = '2008-8-8';
var str2 = '2008-08-08';
var str3 = '08-08-2008';
var str4 = '2008 08 08';
document.write(testReg(reg,str)+'<br />');
document.write(testReg(reg,str2)+'<br />');
document.write(testReg(reg,str3)+'<br />');
document.write(testReg(reg,str4)+'<br />');
</script>
```

第二种格式：来源：[求一正则表达式](#)

yyyy-mm-dd

或

yyyy/mm/dd

用“或”简单地修改一下就行了。

```
/^\\d{4}(-|\\V)\\d{1,2}(-|\\V)\\d{1,2}$/
```

去掉文件的后缀名

来源：[求一个正则](#)

www.abc.com/dc/fda.asp变为www.abc.com/dc/fda。

如果文件后缀已知的话这个问题就非常简单了，正则如下：

```
/\\.asp$/
```

匹配最后的.asp而已，测试代码如下：

```
<script type="text/javascript">
function delAspExtension(str){
    var reg = /\.asp$/;
    return str.replace(reg,"");
}
var str = 'www.abc.com/dc/fda.asp';
document.write(delAspExtension(str)+'<br />');
</script>
```

如果文件名未知的话就用这个正则：/\\.\\w+\$/，测试代码如下：

```
<script type="text/javascript">
function delExtension(str){
    var reg = /\\.\\w+$/;
    return str.replace(reg,"");
}
var str = 'example.com/dc/fda.asp';
document.write(delExtension(str)+'<br />');
var str2 = 'test/regular/fda.do';
document.write(delExtension(str2)+'<br />');
var str3 = 'example.com/dc/fda.strange_extension';
document.write(delExtension(str3)+'<br />');
</script>
```

验证邮箱的正则表达式

来源：找JAVASCRIPT写的表单检查代码！

fuchangxi的正则：

```
/^([a-zA-Z0-9_-])+@([a-zA-Z0-9_-])+(\.[a-zA-Z0-9_-])+/
```

开始必须是一个或者多个单词字符或者是-，加上@，然后又是一个或者多个单词字符或者是-。然后是点"."和单词字符和-的组合，可以有一个或者多个组合。

```
<script type="text/javascript">
function isEmail(str){
    var reg = /^([a-zA-Z0-9_-])+@([a-zA-Z0-9_-])+(\.[a-zA-Z0-9_-])+/;
    return reg.test(str);
}
var str = 'test@hotmail.com';
document.write(isEmail(str)+'<br />');
var str2 = 'test@sima.vip.com';
document.write(isEmail(str2)+'<br />');
var str3 = 'te-st@qq.com.cn';
document.write(isEmail(str3)+'<br />');
var str4 = 'te_st@sima.vip.com';
document.write(isEmail(str4)+'<br />');
var str5 = 'te.._st@sima.vip.com';
document.write(isEmail(str5)+'<br />');
</script>
```

我不太了解邮箱的具体规则。感觉这个正则比较简单，[EMAIL校验 正则 讨论 求解](#)里有比较详细的邮箱正则讨论。

匹配源代码中的链接

来源：[正则](#)

能够匹配HTML代码中链接的正则。

原帖正则：`/.+?/g`

感觉有点严格，首先要``有，而且href属性可以是一个或者多个除换行外任意字符（非贪婪）。后面是`.+?`，一个或者多个除换行外任意字符（非贪婪），再加上结束标签。

有个问题，如果a的起始标签最后有空格，或者除了href还有其它属性的话，上面的正则就不能匹配这个链接了。

例如：

`.....`多了个空格。

`.....`前面有属性。

.....

重写正则：`/<a\s(\s*\w*?=".+?")*(\s*href=".+?")(\s*\w*?=".+?")*\s*>[\s\S]*?/`

思路如下：首先要有`<a`和一个空格。`/<a\s/`

第一个`(\s*\w*?=".+?")*`

可以匹配一个属性，属性前面可能有或者没有多余的空格，用`\s*`匹配；属性名肯定是单词字符，用`\w*`匹配；`=".+?"`就是匹配属性值了非换行字符若干个；整个括号外面加个`*`表示可能有任意多个属性。

`(\s*href=".+?")`

匹配href，它也是一个属性，所以只要把上面子正则表达式中的`\w`修改为`href=`就行了。

`(\s*\w*?=".+?")*`重复第一个子正则表达式，再次接受任意个属性。

`\s*>`，属性最后再加上若干个空格和`>`。

`[\s\S]*?`，链接的文字，可能有任何字符组成，若干个，非贪婪。

``最后是结束标签。

补充：属性名和`=`之间，以及`=`和属性值之间也可能有空格。所以要再加上几个`\s*`。

最后的实例代码如下：

```
<script type="text/javascript">
function findLinks(str){
    var reg =
    /<a\s(\s*\w*?\s*=\s*".+?")*(\s*href\s*=\s*".+?")(\s*\w*?\s*=\s*".+?")
    *\s*>[\s\S]*?</a>/g;
    var arr = str.match(reg);
    for(var i=0;i<arr.length;i++){
```

```

        //alert(arr[i]);
        document.write('link:'+arr[i]+'<br />');
    }
}
var str = '<p>测试链接: <a id = "test" href="http://bbs.blueidea.com" title="无敌">经典论坛
</a></p><a href = "http://www.blueidea.com/" >蓝色理想</a>';
var arr = findLinks(str);
</script>

```

会把所有的链接在页面直接显示出来。注意，
 本帖遗留问题：如何执行从右到左的匹配。貌似 **JS** 或者**VBS** 没有提供这个功能 **2、JS** 或者**VBS** 不支持 后行断言。。用什么方法实现这个功能。

匹配链接的文字

来源: [求一正则](#)

代码: ``这里要保存``，只保存链接的文本内容，标签信息删掉。

前面写过一个匹配链接的正则：

```
/<a\s(\s*\w*?="\.+?")*(\s*href="\.+?")*\s*>[\s\S]*?</a>/
```

不过我们需要捕获的是文字内容，所以需要做一些修改。第一步就是在所有的括号内都加上?:表示不捕获。第二步就是再多加一个括号放在`[\s\S]*?`两侧，这样就可以捕获到链接的文字内容了。最后正则如下：

```
/<a\s(?:\s*\w*?\s*=\s*"\.+?")*(?:\s*href\s*=\s*"\.+?")(?:\s*\w*?\s*=\s*"\.+?")*\s*>([\s\S]*?)</a>/
```

测试代码如下：

```

<script type="text/javascript">
function anchorText(str){
    var reg =
    /<a\s(?:\s*\w*?\s*=\s*"\.+?")*(?:\s*href\s*=\s*"\.+?")(?:\s*\w*?\s*=\s*"\.+?")*\s*>([\s\S]*?)</a>/;
    str = str.replace(reg,'$1');
    return str;
}
var str = '<a id = "test" href="http://bbs.blueidea.com" title="无敌">经典论坛</a>';
document.write(anchorText(str));
</script>

```

正则判断标签是否闭合

来源: [求助 超难！正则表达式如何判断一个标签是否闭合](#)

例如: `<img xxx="xxx"` 就是没有闭合的标签；

`<p>p`的内容，同样也是没闭合的标签。

从简单的正则开始，先匹配起始标签

```
/<[a-z]+/i
```

再加上若干属性：

```
/<[a-z]+(\s*\w*?\s*=\s*"\.+?")*/i
```

下面就到关键点了，标签的闭合。标签可能有两种方式闭合，``

或者是`<p>xxx </p>`。

```
(\s*\V>)
```

匹配img类的结束，即`>`。

```
(\s*?>[\s\S]*?<\1>)
```

匹配p类标签的结束标签。`>`是其实标签末尾，之后是标签内容若干个任意字符，最后的`<\1>`就是结束标签了。

加上一个或就可以解决了，最后的完整正则表达式：

整个正则: `/<([a-z]+)(\s*\w*?\s*=\s*"\.+?")*(\s*?>[\s\S]*?<\1>|\s*\V>)/i`

拿这个正则，只要匹配到了就表示闭合，没匹配到则没有闭合。不过没有考虑相同标签嵌套的问题，例如

```
<div>aaaaaa<div>test</div>
```

也被判断为合格，可以通过把最后的匹配p类结束标签写成子正则表达式，并且更改为非贪心，然后在匹配结果中检查是否成对。正则如下：

```
/<([a-z]+)(\s*\w*?\s*=\s*".+?")*(\s*>[\s\S]*?(<\1>)+|\s*\V>)/i
```

用正则获得指定标签的内容

来源：[求一正则](#)

有如下代码：

```
<channel>
  <title>蓝色理想</title>
</channel>
<item>
  <title>界面设计测试规范</title>
</item>
<item>
  <title>《古典写实美女》漫画教程</title>
</item>
<item>
  <title>安远——消失的光年</title>
</item>
<item>
  <title>asp.net 2.0多语言网站解决方案</title>
</item>
```

要求匹配item里的title而不匹配channel里的title。

基本正则：

```
/<title>[\s\S]*?<\title>/gi
```

首先是title标签，内容为任意字符若干个，然后是title结束标签。这个正则已经能匹配到所有的title标签。

首先，我简单地修改了一下原正则：

```
/<title>[^<>]*?<\title>/gi,
```

因为title里面不应该再嵌有其它标签，这个正则同样是匹配所有标题的内容，最后再加上不去匹配channel中的title。整个正则如下：

```
/<title>[^<>]*?<\title>(?!s*<\channel>)/gi
```

(?!s*<\channel>)表示要匹配字符串的后面不能跟着若干个空格和一个channel的结束标签。

原帖里有很方便的测试工具，这里就不给测试代码了。

正则判断是否为数字与字母的混合

来源：[关于正则](#)

不能小于12位，且必须为字母和数字的混合。

验证字符串包含数字简单，验证字符串包含字母也简单，验证字符串不包含其它字符也简单，可以用这三个正则分别检查一次字符串，逻辑运算出最终结果。

但是怎么能把这些功能写进一个正则表达式里呢？这个问题真有点伤脑筋。

下面是lexrus的正则：

```
/^([a-z]+(?=[0-9])|[0-9]+(?=[a-z]))[a-z0-9]+$ /ig
```

思路非常的清晰啊：

```
[a-z]+(?=[0-9])
```

字母开头，后面必须紧跟着数字。

```
[0-9]+(?=[a-z])
```

数字开头，后面必须紧跟着字母。

```
[a-z0-9]+
```

后面的字符只要是数字或者字母就可以了。经过测试，发现不好使，123dd会被识别为不合法，dd123则为合法，可见“数字开头，紧跟字母”的正则没有起作用。测试代码如下：

```
<script type="text/javascript">
function istrue(str){
var reg=/^([a-z]+(?=[0-9])|[0-9]+(?=[a-z]))[a-z0-9]+$/ig;
return reg.test(str);
}
var str = 'AaBc';
var str2 = 'aaa123';
var str3 = '123dd';
var str4 = '1230923403982';
document.write(istrue(str)+'<br />');
document.write(istrue(str2)+'<br />');
document.write(istrue(str3)+'<br />');
document.write(istrue(str4)+'<br />');
</script>
```

结果为：

false, true, false, false

结果中的第三个，将'123dd'判断为非法是错误的。刚开始以为是g的问题，去掉了还是不好使。应该是浏览器bug，我认为lexrus的正则是正确的，可能是浏览器无法处理或“|”的两边都包含正向预查(?=)。

修改之后的正则如下：

```
/^(([a-z]+[0-9]+)|([0-9]+[a-z]+))[a-z0-9]*$/i
```

意思和上面差不多，但是没有使用正向预查，测试代码如下：

```
<script type="text/javascript">
function istrue(str){
var reg=/^(([a-z]+[0-9]+)|([0-9]+[a-z]+))[a-z0-9]*$/i;
return reg.test(str);
}
var str = 'AaBc';
var str2 = 'aaa123';
var str3 = '123dd';
var str4 = '1230923403982';
document.write(istrue(str)+'<br />');
document.write(istrue(str2)+'<br />');
document.write(istrue(str3)+'<br />');
document.write(istrue(str4)+'<br />');
</script>
```

结果为

false, true, true, false

正确。

空格与英文同时存在

来源：[正则问题请指教啊！](#)

匹配英文以及空格，要求必须既有英文字母又有空格。

这个思路和上面的差不多，只要把数字改成空格就可以了。正则如下：

```
/^(([a-z]+\s+)|(\s+[a-z]+))[a-z\s]*$/i
```

英文开头加空格，或者是空格开头加英文，后面可以是英文或者空格。测试代码如下：

```
<script type="text/javascript">
function istrue(str){
var reg=/^(([a-z]+\s+)|(\s+[a-z]+))[a-z\s]*$/i;
return reg.test(str);
}
var str = 'asdf';
var str2 = 'sadf sdf';
```

```
var str3 = 'asdf ';
document.write(istrue(str)+'<br />');
document.write(istrue(str2)+'<br />');
document.write(istrue(str3)+'<br />');
</script>
```

利用这个思路也可以实现英文空格英文，英文单词多于两个的匹配。同样，也可以把英文字母换成单词字符\w。

显示或者保存正则表达式匹配的部分内容

有如下电话号码：

```
13588888333
13658447322
13558885354
13587774654
13854554786
```

要求，要求只匹配135开头的电话，但是匹配结果只保留135后面的数字。

由于JavaScript里的正则不支持(?=xx)xxx的模式，只支持xxx(=xx)的模式。所以只能将135后面的内容作为一个子正则表达式匹配的内容，然后再在后面引用。

Carl给出的函数如下：

```
function f(phoneNumber) {
    var pattern = /^(135)(\d{8})$/;
    if(pattern.test(phoneNumber))
        return phoneNumber.replace(pattern,"$2");
    else
        return "不是135打头的手机号码！";
}
```

```
/^(135)(\d{8})$/
```

正则中，135作为开头表示第一个子正则表达式，第二个括号内的子正则表达式则匹配后面的8个数字，然后在replace中使用\$2就可以引用这个子正则表达式匹配的内容了。测试代码如下：

```
<script type="text/javascript">
function f(phoneNumber) {
    var pattern = /^(135)(\d{8})$/;
    if(pattern.test(phoneNumber))
        return phoneNumber.replace(pattern,"$2");
    else
        return "不是135打头的手机号码！";
}
var arr = new Array(
    "13588888333",
    "13658447322",
    "13558885354",
    "13587774654",
    "13854554786"
);
for(var i = 0; i < arr.length; i++)
    document.write(f(arr[i])+'<br />');
</script>
```

正则表达式替换变量

来源：[求教正则](#)

有一个数组：

```
var _A = ['A','B','C','D'];
```

有一个有“变量”的字符串。

```
var _B = '<ul><li>$0$< / li><li>$1$< / li><li>$2$< / li><li>$3$< / li></ul>';
```

说是变量，其实只是字符串中的特殊字符，例如\$0\$，就称这个为伪变量吧。

最后的要求就是使用正则获得下面这样一个字符串：

```
_C = '<ul><li>A< / li><li>B< / li><li>C< / li><li>D< / li></ul>';
```

IamUE给出了代码：

```
<script type="text/javascript">
var _A = ['A','B','C','D'];
var _B = '<ul><li>$0$< / li><li>$1$< / li><li>$2$< / li><li>$3$< / li></ul>';
var reg=/\$\\d+\\$/ig;
C=_B.replace(reg,function($1){
var indexnum=$1.replace(/\\$/ig,"");
if (indexnum<_A.length)
{return _A[indexnum];}
else{return ""}
});
alert(C);
</script>
```

代码分析：看到代码之后感觉有点晕，首先，正则reg中没有任何的括号，应该还是没有捕获内容的，那么后面怎么又使用\$1了引用了呢？通过alert测试，发现它是整个正则匹配的内容，而且不一定要写作\$1，可以写为\$0，甚至是写为x都没关系，它总是整个匹配。

第一次，\$1匹配到_B中的"\$0\$"，匿名函数中将它的\$去掉，变成了0，检查是否越界之后，用这个0作为下标去访问数组_A。

由于正则reg定义了g属性，所以会继续替换\$1\$、\$2\$等等。步骤都和上面一样。

正则替换指定属性中的文本

来源：[怎样用正则来查找替换？](#)

有如下代码：

```
<td align="center"><br>
O'Malley's West</td>
```

要求将所有onclick属性中的'替换成\'，也就是将单引号转义。

首先，需要匹配onclick属性：

```
/onclick\s*=\s*".+?"/ig
```

然后再将所有的'都替换成\'就可以了。整个测试的代码如下：

```
<script type="text/javascript">
<!--
function $(id) {
    return document.getElementById(id);
}
function replace() {
    $("txtDes").value = $("txtSrc").value.replace(/onclick\s*=\s*".+?"/ig, function (m) {
        return m.replace(/'/g, "\\'");
    });
}
//-->
</script>
<textarea id="txtSrc" rows="15" cols="80">
<td align="center"><br>
O'Malley's West</td>
```



```

<td align="center"><br>
Coach's Bar & Grill</td>
<td align="center"><br>
Betty's Burger (Hamburger Mary's)</td>
</textarea>
<br/>
<input type="button" onclick="replace();" value=" replace " />
<br/>
<textarea id="txtDes" rows="15" cols="80">
</textarea>

```

将阿拉伯数字替换为中文大写形式

来源: [正则问题](#)

将123替换成壹贰叁。

只要匹配一个数字就可以了，测试代码如下（如果显示为乱码就调整一下浏览器的字符编码）：

```

function replaceReg(reg,str){
    return str.replace(reg,function(m){return arr[m];})
}
arr=new Array("零","壹","贰","叁","肆","伍","陆","柒","捌","玖");
var reg = /\d/g;
var str = '13889294444';
var str2 = '12889293333';
var str3 = '23445567';
document.write(replaceReg(reg,str)+'<br />');
document.write(replaceReg(reg,str2)+'<br />');
document.write(replaceReg(reg,str3)+'<br />');

```

替换文本中的URL为链接

来源: [求一链接替换正则](#)

将一个用户输入的一段文字中的url替换成可以点击的link地址。例如：http://www.blueidea.com可以替换成 [url]http://www.cctv.com[/url]

或http://www. blueidea.com.

这个正则的关键就在于匹配链接，匹配之后，在两边加上a标签和属性不是问题。

/http:VV[\w-]*([\w-]*)+/ig

首先匹配http://。

[\w-]*是可能的www和bbs等。

.[\w-]*匹配.xxx形式，至少有一个。

测试代码如下：

```

<script type="text/javascript">
function replaceReg(reg,str){
    return str.replace(reg,function(m){return '<a href="'+m+'"'>' + m + '</a>';})
}
var reg = /http:VV[\w-]*([\w-]*)+/ig;
var str = '将一个用户输入的一段文字中的url替换成可以点击的link地址。测试一
下：http://www.blueidea.com紧接着中文，还有http://bbs.blueidea.com is very
good!http://blueidea.com!最后在看看带.cn的：http://www.sina.com.cn呵呵。';
document.write(replaceReg(reg,str)+'<br />');
</script>

```


从HTML代码段删除指定标签及其内容

来源: [关于正则的问题](#)

在一段代码中去除<script /script>, <head>...</head>, <%.....%>等代码块的正则:

```
/<(script|meta|%)[\s\S]*?\/(script|meta|%)>/
```

试了一下, 匹配如下文本正常:

```
<script type="text/javascript">
```

我是要被删除的脚本

```
</script>
```

哎。就剩下我了。

但是, 如果使用类似的正则:

```
/<(script|head|%)[\s\S]*?\/(script|head|%)>/ig
```

匹配有嵌套的标签:

```
<head>
```

```
<script type="text/javascript">
```

我是要被删除的脚本

```
</script>
```

```
</head>
```

哎。就剩下我了。

实际匹配的内容是:

```
<head>
```

```
<script type="text/javascript">
```

我是要被删除的脚本

```
</script>
```

这是因为[\s\S]*?里的非贪婪造成的。可以使用JavaScript正则里的反向引用来解决这个问题, 如果起始标签匹配了head, 那么结束标签也必须是head。最后的正则如下:

```
/<(script|head|%)[\s\S]*?\/\1>/ig
```

用正则给文本分段

来源: [怎样用正则分段落](#)

源代码: [title]标题一[/title]内容一232323sdfga [title]标题二[/title]内容二2232323 [title]标题三[/title]内容三2232323

要把文本分段成如下格式:

一、[title]标题一[/title]内容一232323sdfga

二、[title]标题二[/title]内容二2232323

三、[title]标题三[/title]内容三2232323

只要用正则匹配title就可以了, 所以正则比较简单

```
/\[title\]/ig
```

至于开始的的汉字序号, 只要一个数组就解决了, 最终代码如下:

```
<script type="text/javascript">
```

```
function replaceReg(reg,str){
```

```
    var mark =0;
```

```
    return str.replace(reg,function(m){mark++;return '<br />'+arr[mark]+'、'+m;})
```

```
}
```

```
var arr = ["零","壹","贰","叁","肆","伍","陆","柒","捌","玖"];
```

```
var reg = /\[title\]/ig;
```

```
var str = '[title]标题一[/title]内容一232323sdfga [title]标题二[/title]内容二2232323 [title]标题三[/title]内容三2232323';
```

```
document.write(replaceReg(reg,str)+'<br />');
```

```
</script>
```

转换源代码中的标签

来源: [一个正则上的问题](#)

将代码中的HTML标签

/<img(?:\s*\w*\s*\s*=\s*"(.+?)"(?:\s*\w*\s*\s*=\s*"(.+?)"*)\s*>/ig

这段正则和匹配链接标签的正则基本一样,修改如下,标签名img,没有结束标签而是>结束。

测试代码如下:

```
<script type="text/javascript">
function replaceReg(reg,str){
    return str.replace(reg,'[img]$1[/img]')
}
var reg =
/<img(?:\s*\w*\s*\s*=\s*"(.+?)"(?:\s*\w*\s*\s*=\s*"(.+?)"*)\s*>/ig;
```

var str = '我就是传说中的图片了哎。';

document.write(replaceReg(reg,str)+'
');

</script>

第二个是替换object代码嵌入的flash代码替换为[swf]url[/swf]。

针对原文的正则如下:

/<object[\s\S]*?src=(?=[\s\S]+?)(?=\s)[\s\S]*<\object>/i

如果是所有的属性都有双引号的话正则也需要修改。

测试如下:

```
<script type="text/javascript">
function replaceReg(reg,str){
    return str.replace(reg,'[swf]$1[/swf]')
}
var reg = /<object[\s\S]*?src=(?=[\s\S]+?)(?=\s)[\s\S]*<\object>/i;
var str = '<object classid=clsid:D27CDB6E-AE6D-11cf-96B8-444553540000
codebase=http://download.macromedia.com/pub/shockwave/cabs/flash/
swflash.cab#version=5,0,0,0 width=255 height=250><param name=movie
value=url><param name=quality value=high><embed src=url quality=high
pluginspage=http://www.macromedia.com/shockwave/download/
index.cgi?P1_Prod_Version=ShockwaveFlash type=application/x-shockwave-flash
width=255 height=250></embed></object>';
document.write(replaceReg(reg,str)+'<br />');
</script>
```

给属性添加双引号

来源: [请教正则表达式高手](#)

给HTML标签中的属性添加双引号。

改为:

LeXRus的第一个正则如下:

/(!<(\w+)(\s+(\w+))\s*=\s*"(.+)"\s*)/ig

第一个括号没看明白,JS应该是不支持。所以我擅自给删掉了,剩下的正则如下:

/(\s+(\w+))\s*=\s*"(.+)"\s*/ig

第一个括号里的\s+(\w+)匹配的是属性名。

然后是=,不用转义。

第二个括号里的[^\s"]匹配属性值。不匹配>和空格。这里的引号不用转义。在意思不改变的情况下,稍微改了改,正则如下:

/(\s+(\w+))\s*=\s*"(.+)"\s*/ig

需要注意的是这个正则不匹配=两边有空格的属性,例如href = xxx。相匹配的话就改成:

/(\s+(\w+))\s*=\s*"(.+)"\s*/ig

代码:

```
str=str.replace(/(!<(\w+)(\s+(\w+))\s*=\s*"(.+)"\s*)/ig,'$1="$2"');
```

其中'\$1="\$2"'就实现了给属性值添加上双引号。不过ncs指出了这个正则替换的几个问题，一是上面的空格问题，二是如果非标签内部有等号，且前面又恰巧有空白字符的话，它将会被误识别为属性，例如：

```
<a href=xxx target=yyy title = asdfasf> test=sd
```

里面的test=sd也会被匹配。三是如果属性原来使用了单引号，会被再包上一层双引号.....

来看看LeXRus前辈的新正则替换方法：

```
str=str.replace(/(?!<\w+)(\s+\w+)\s*=\s*([^\>\"'\s]+)(?=[^\>]*>)/ig,'$1="$2"').replace(/\\"'([^\\"']+)\\"'/ig,\"$1\");
```

先来看第一个正则：

```
/(\s+\w+)\s*=\s*([^\>\"'\s]+)(?=[^\>]*>)/ig
```

结尾新添的(?=[^\>]*>)意在解决普通文本中有等号被误识别为属性的问题：

```
<a href=xxx target=yyy title = asdfasf> test=sd
```

就没问题了，但是

```
<a href=xxx target=yyy title = asdfasf> test=sd<tag> 又一个标签</tag>
```

中的test=sd<tag> 又会被识别为属性。

我觉得改成下面的正则就没问题了：

```
/(\s+\w+)\s*=\s*([^\<>\"'\s]+)(?=[^\<>]*>)/ig
```

分别在第二个括号的字符集和最后的反向预查的字符集中添加了一个<。

下面再来分析第二个正则，

```
/\"'([^\\"']+)\\"'/ig
```

这个正则用于匹配双引号，单引号多层嵌套的情况，同样，不用转义，修改正则如下：

```
/\"'([^\\"']*)\\"'/ig
```

这样基本任务就完成了。测试代码如下：

```
<script>
```

```
function rp(str,trg){
    var reg1 = /(\s+\w+)\s*=\s*([^\<>\"'\s]+)(?=[^\<>]*>)/ig
    var reg2 = /\"'([^\\"']*)\\"'/ig;
    str=str.replace(reg1,'$1="$2"').replace(reg2,\"$1\");
    trg.value=str;
}
```

```
}
```

```
</script>
```

```
<textarea id="sou" style="width:100%">
```

```
<a href = xxx name=aaa target=_blank title='asdfasf' onclick=alert('blueidea')>
```

```
asfd=asfd
```

```
</textarea>
```

```
<input type="button" onclick="rp(sou.value,sou)" value="replace"/>
```

原帖里LeXRus又提出了新问题：

hint=i am lexrus

这样的属性会有问题，不过我感觉不加引号的话，属性值里就不可能有空格，否则会被识别为多个属性了。不过看到最后ncs的回帖我就哭了：

```
onclick=if(document.forms.length>0)
```

这样的属性怎么办？大于号会被识别为标签结束.....还是分离行为与文档吧。补充一下，其实修补一下

正则也可以解决，只要改成如下正则即可：

```
/(\s+\w+)\s*=\s*([^\\"'\s]+)(?=[^\<>]*>)/ig
```

就是去掉第二个括号内字符集合里的<>。最后这个问题也解决。

给table加上tbody

来源：[通过正则表达式快速在后加上](#)

有若干table，但是没有tbody。现在需要用正则批量加上。

匹配table结束标签</table>比较简单，在前面加上一个</tbody>就行了。

但是，匹配table的起始标签有点难度，因为可能有属性。不过之前匹配过链接了，这个也大同小异。

实例table代码如下：

```
<table width="100%" border="0" cellpadding="2" cellspacing="3">
```

```
<table width="100%">
```

正则：

```
/<table\s(\s*\w*?\s*=\s*".+?")*?\s*?>/g
```

匹配一个<table，在匹配若干个属性，最后只要再找到>就代表标签结束。

之后再replace一下，加上<tbody>就可以了。

去掉标签的所有属性

来源: [正则表达式\(DREAMWEAVER8\)](#)

```
<td style="width: 23px; height: 26px;" align="left">***</td>
```

变成没有任何属性的

```
<td>***</td>
```

思路: 非捕获匹配属性, 捕获匹配标签, 使用捕获结果替换掉字符串。正则如下:

```
/(<td>\s(?:\s*\w*?\s*=\s*".+?")*?\s*?(>)/
```

首先, td匹配掉了标签, 后面可以用\$1引用, 后面的若干属性被(?:)匹配掉, 而最后匹配的>则可以在后面用\$2引用。

示意代码:

```
str = str.replace(reg,'$1$2');
```

正则替换特定单词

来源: [正则表达式,拒绝几个单词的输入该怎样表达?](#)

要求禁止输入某几个单词, 如果拒绝red,yellow,white。这个帖子到时不难, 但是让我弄清楚了好几个概念。

第一个, 小心字符集合里的“或”

```
/[^red|yellow|white]/
```

这个正则里的所有或都没有意义, 等同于:

```
/[^redyellowwhite]/
```

意思就是不能含有以下列出的所有字母。

正解:

```
/red|yellow|white/
```

第二个概念:

只要整个正则匹配成功, 无论子正则表达式是否匹配成功, 括号都会捕捉。例如

```
/((red)|(yellow)|(white))/
```

会捕捉到三个结果, 尽管实际上最多只可能有一个括号匹配成功。但是只要有一个匹配到了, 两外两个也会记录空串。

指定文字高亮显示

来源: [如何用正则把网页中的指定的字高亮显示](#)

[请教正则表达式: 如何替换搜索结果中的关键字为高亮显示?](#)

不劳而获一次, 这个子虚乌有前辈已经给出了非常好的解决方案: 我直接把代码贴出来了:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/2000/REC-xhtml1-20000126/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
```

```
<META name="Author" content="Sheneyan" />
```

```
<script type="text/javascript">
```

```
function encode(s){
```

```
    return s.replace(/&/g,"&amp;").replace(/</g,"&lt;").replace(/>/g,"&gt;").replace(/
```

```
([\\\.\\*\\[\\]\\(\\)$\\^])/g,"\\$1");
```

```
}
```

```
function decode(s){
```

```
    return s.replace(/\\([\\\.\\*\\[\\]\\(\\)$\\^])/g,"$1").replace(/&gt;/g,">").replace(/&lt;/g,"<").replace(/&amp;/g,"&");
```

```

}
function highlight(s){
  if (s.length==0){
    alert('搜索关键词未填写! ');
    return false;
  }
  s=encode(s);
  var obj=document.getElementsByTagName("body")[0];
  var t=obj.innerHTML.replace(/<span\s+class=\.?highlight\.?>([^\<>]*)<\span>/
gi,"$1");
  obj.innerHTML=t;
  var cnt=loopSearch(s,obj);
  t=obj.innerHTML
  var r=/\{searchHL\}((\{?!\/searchHL\})|[\^{\}])*{\\/searchHL\}/g
  t=t.replace(r,"<span class='highlight'>$1</span>");
  obj.innerHTML=t;
  alert("搜索到关键词"+cnt+"处")
}
function loopSearch(s,obj){
  var cnt=0;
  if (obj.nodeType==3){
    cnt=replace(s,obj);
    return cnt;
  }
  for (var i=0,c=obj.childNodes[i];i++){
    if (!c.className||c.className!="highlight")
      cnt+=loopSearch(s,c);
  }
  return cnt;
}
function replace(s,dest){
  var r=new RegExp(s,"g");
  var tm=null;
  var t=dest.nodeValue;
  var cnt=0;
  if (tm=t.match(r)){
    cnt=tm.length;
    t=t.replace(r,"{searchHL}"+decode(s)+"{/searchHL}")
    dest.nodeValue=t;
  }
  return cnt;
}
</script>
<style type="text/css">
.highlight{background:green;font-weight:bold;color:white;}
</style>
</head>
<body>
<form onsubmit="highlight(this.s.value);return false;">
<p><input name="s" id="s" title="搜索内容: "/><input type="submit" value="搜索"/></p>
</form>
<div id="content">
测试高亮的代码。很长很长的代码.....
</div>
</body>
</html>

```

删除标签

来源: [如何使用正则表达式去除大部分HTML标记?](#)

删除除了、
、<p>之外所有的标签。子虚乌有给出代码中关键的一句:

```
o.innerHTML.replace(/(<V?(?!br|p|img)[^>V]*)V?>/gi,"");
```

刚开始没反应过来,后来才想起来,这个正则不用区分起始和结束标签。

<V?(?!br|p|img)

匹配除了保护标签外标签的起始标签或者是结束标签的一部分。

[^>V]*

匹配到>或者/就结束。

V?>

起始标签或者结束标签的结尾。

错误贴

[熟悉正则的帮忙看看,一个小问题:反斜线灾难。](#)

[正则表达式通过变量传递后,\T转义字符的奇怪表现。](#):同上。

太难贴

我从来没用过UBB,这个关于UBB的帖子,我觉得比较难,就没有总结: [关于表格的正则表达式讨论\(已实现,特别感谢LEXRUS\)](#)。

JSON我也不太明白,这个贴还没看,以后学了JSON再说。 [正则替换JSON数据问题](#)。

还有一个挺难的帖子, [正则的难题](#),感觉不应该用正则。

正则表达式做打字练习游戏: [有什么特别好的办法\(正则表达式问题\)](#)。里面有位高手做出来了,不过是只在IE下有效。

匹配数字的“正则”

来源: [\[求助\]匹配数字的正则式](#)

匹配数字字符挺简单,但是匹配正数,负数之类的还真挺麻烦。不过子虚乌有前辈的一句话到是非常经典:“不要一想到验证就想去用正则”。

是啊,不会正则的时候使劲躲,看了两天正则恨不得JavaScript编程“正则Script”。

类似的:

[请问这个问题是不是必须要用正则表达式才能结局啊解决啊?](#)

[哪位高手知道怎么用正则替换图片名称吗?](#)

[复杂正则求助。。。。。。](#)

正则资源

主题: [JS正则表达式详解\[收藏\]](#)

[发个正则测试工具](#)

<http://www.regexlib.com>