

正则表达式语法及常用用法

一、元字符

元字符是构造 **正则表达式** 的一种基本元素。

. : 匹配除换行符以外的任意字符

w: 匹配字母或数字或下划线或汉字

s: 匹配任意的空白符

d: 匹配数字

b: 匹配单词的开始或结束

^: 匹配字符串的开始

\$: 匹配字符串的结束

```
1 | 匹配有abc开头的字符串: abc或者^abc
2 |
3 | 匹配8位数字的QQ号码: ^ddddddd$
4 |
5 | 匹配1开头11位数字的手机号码: ^1ddddddddd$
```

二、重复限定符

正则没提供办法处理这些重复的元字符吗? 答案肯定是有。

*: 重复零次或更多次

+: 重复一次或更多次

?: 重复零次或一次

{n}: 重复n次

{n,}: 重复n次或更多次

{n,m}: 重复n到m次

有了这些限定符之后, 我们就可以对之前的正则表达式进行改造了, 比如:

```
1 | 匹配8位数字的QQ号码: ^d{8}$
2 |
3 | 匹配1开头11位数字的手机号码: ^1d{10}$
4 |
5 | 匹配银行卡号是14~18位的数字: ^d{14,18}$
6 |
7 | 匹配以a开头的, 0个或多个b结尾的字符串^ab*$
```

三、分组 ()

限定符是作用在与他左边最近的一个字符，那么问题来了，如果我想要ab同时被限定那怎么办呢？

正则表达式中用小括号()来做分组，也就是括号中的内容作为一个整体。

因此当我们要匹配多个ab时，我们可以这样。

```
如匹配字符串中包含0到多个ab开头: ^(ab)*
```

复制

四、转义

正则提供了转义的方式，也就是要把这些元字符、限定符或者关键字转义成普通的字符，做法很简答，就是在要转义的字符前面加个斜杠，也就是\即可。

```
匹配字符串中包含0到多个ab开头: ^(\(ab\))*
```

五、条件或 |

回到我们刚才的手机号匹配，我们都知道：国内号码都来自三大网，它们都有属于自己的号段，比如联通有130/131/132/155/156/185/186/145/176等号段，假如让我们匹配一个联通的号码，那按照我们目前所学到的正则，应该无从下手的，因为这里包含了一些并列的条件，也就是“或”，那么在正则中是如何表示“或”的呢？

正则用符号 | 来表示或，也叫做分支条件，当满足正则里的分支条件的任何一种条件时，都会当成是匹配成功。

那么我们就可以用或条件来处理这个问题

```
^(130|131|132|155|156|185|186|145|176)\d{8}$
```

复制

六、区间[]

看到上面的例子，是不是看到有什么规律？是不是还有一种想要简化的冲动？实际是有的

正则提供一个元字符中括号 [] 来表示区间条件。

1. 限定0到9 可以写成[0-9]
2. 限定A-Z 写成[A-Z]
3. 限定某些数字 [165]

那上面的正则我们还改成这样：

```
^((13[0-2])|(15[56])|(18[5-6])|145|176)\d{8}$
```

常用用法

一、校验数字的表达式

- 1 数字：`^[0-9]*$`
- 2 n位的数字：`^\d{n}$`
- 3 至少n位的数字：`^\d{n,}$`

4 m-n位的数字: `^\d{m,n}$`
 5 零和非零开头的数字: `^(0|[1-9][0-9]*)$`
 6 非零开头的最多带两位小数的数字: `^([1-9][0-9]*)(.[0-9]{1,2})?$`
 7 带1-2位小数的正数或负数: `^(\\-)?\\d+(\\.\\d{1,2})?$`
 8 正数、负数、和小数: `^(\\-|\\+)?\\d+(\\.\\d+)?$`
 9 有两位小数的正实数: `^[0-9]+(.[0-9]{2})?$`
 10 有1~3位小数的正实数: `^[0-9]+(.[0-9]{1,3})?$`
 11 非零的正整数: `^[1-9]\\d*$` 或 `^([1-9][0-9]*){1,3}$` 或 `^\\+?[1-9][0-9]*$`
 12 非零的负整数: `^-\\[1-9][0-9]*$` 或 `^-\\[1-9]\\d*$`
 13 非负整数: `^\\d+$` 或 `^[1-9]\\d*|0$`
 14 非正整数: `^-\\[1-9]\\d*|0$` 或 `^((-\\d+)|(0+))$`
 15 非负浮点数: `^\\d+(\\.\\d+)?$` 或 `^[1-9]\\d*\\.\\d*|0\\.\\d*[1-9]\\d*|0?\\.0+|0$`
 16 非正浮点数: `^((-\\d+(\\.\\d+)?)|(0+(\\.0+)?))$` 或 `^-([1-9]\\d*\\.\\d*|0\\.\\d*[1-9]\\d*)|0?\\.0+|0$`
 17 正浮点数: `^[1-9]\\d*\\.\\d*|0\\.\\d*[1-9]\\d*$` 或 `^((([0-9]+\\. [0-9]*[1-9][0-9]*)|([0-9]*[1-9][0-9]*\\. [0-9]+)|([0-9]*[1-9][0-9]*))$`
 18 负浮点数: `^-([1-9]\\d*\\.\\d*|0\\.\\d*[1-9]\\d*$` 或 `^-((([0-9]+\\. [0-9]*[1-9][0-9]*)|([0-9]*[1-9][0-9]*\\. [0-9]+)|([0-9]*[1-9][0-9]*))$`
 19 浮点数: `^(-?\\d+)(\\.\\d+)?$` 或 `^-?([1-9]\\d*\\.\\d*|0\\.\\d*[1-9]\\d*|0?\\.0+|0)$`

二、校验字符的表达式

1 汉字: `^[\\u4e00-\\u9fa5]{0,}$`
 2 英文和数字: `^[A-Za-z0-9]+$` 或 `^[A-Za-z0-9]{4,40}$`
 3 长度为3-20的所有字符: `^. {3,20}$`
 4 由26个英文字母组成的字符串: `^[A-Za-z]+$`
 5 由26个大写英文字母组成的字符串: `^[A-Z]+$`
 6 由26个小写英文字母组成的字符串: `^[a-z]+$`
 7 由数字和26个英文字母组成的字符串: `^[A-Za-z0-9]+$`
 8 由数字、26个英文字母或者下划线组成的字符串: `^\\w+$` 或 `^\\w{3,20}$`
 9 中文、英文、数字包括下划线: `^[\\u4E00-\\u9FA5A-Za-z0-9_]+$`
 10 中文、英文、数字但不包括下划线等符号: `^[\\u4E00-\\u9FA5A-Za-z0-9]+$` 或 `^[\\u4E00-\\u9FA5A-Za-z0-9]{2,20}$`
 11 可以输入含有`%&',';=?\$\"`等字符: `[^%&',';=?$\\x22]+`
 12 禁止输入含有~的字符: `[^~\\x22]+`

三、特殊需求表达式

1 Email地址: `^\\w+([-+.]\\w+)*@\\w+([-+.]\\w+)*\\.\\w+([-+.]\\w+)*$`
 2 域名: `[a-zA-Z0-9][-a-zA-Z0-9]{0,62}(/.[a-zA-Z0-9][-a-zA-Z0-9]{0,62})+/.?`
 3 InternetURL: `[a-zA-z]+://[^\\s]*` 或 `^http://([\\w-]+\\.)+[\\w-]+(/[\\w-./?%&=]*)?$`
 4 手机号码: `^(13[0-9]|14[5|7]|15[0|1|2|3|5|6|7|8|9]|18[0|1|2|3|5|6|7|8|9])\\d{8}$`
 5 电话号码("XXX-XXXXXXX"、"XXXX-XXXXXXX"、"XXX-XXXXXXX"、"XXX-XXXXXXX"、"XXXXXXX"和"XXXXXXX"): `^(\\(\\d{3,4}-)|\\d{3,4}-)?\\d{7,8}$`
 6 国内电话号码(0511-4405222、021-87888822): `\\d{3}-\\d{8}|\\d{4}-\\d{7}`
 7 身份证号(15位、18位数字): `^\\d{15}|\\d{18}$`
 8 短身份证号码(数字、字母x结尾): `^([0-9]){7,18}(x|X)?$` 或 `^\\d{8,18}|[0-9x]{8,18}|[0-9X]{8,18}$`
 9 帐号是否合法(字母开头,允许5-16字节,允许字母数字下划线): `^[a-zA-Z][a-zA-Z0-9_]{4,15}$`
 10 密码(以字母开头,长度在6~18之间,只能包含字母、数字和下划线): `^[a-zA-Z]\\w{5,17}$`

11 强密码(必须包含大小写字母和数字的组合,不能使用特殊字符,长度在8-10之间): `^(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,10}$`

12 日期格式: `^\d{4}-\d{1,2}-\d{1,2}`

13 一年的12个月(01~09和1~12): `^(0?[1-9]|1[0-2])$`

14 一个月的31天(01~09和1~31): `^((0?[1-9])|((1|2)[0-9])|30|31)$`

15 钱的输入格式:

16 1).有四种钱的表示形式我们可以接受:"10000.00" 和 "10,000.00", 和没有 "分" 的 "10000" 和 "10,000": `^[1-9][0-9]*$`

17 2).这表示任意一个不以0开头的数字,但是,这也意味着一个字符"0"不通过,所以我们采用下面的形式:
`^(0|[1-9][0-9]*)$`

18 3).一个0或者一个不以0开头的数字.我们还可以允许开头有一个负号: `^(0|-?[1-9][0-9]*)$`

19 4).这表示一个0或者一个可能为负的开头不为0的数字.让用户以0开头好了.把负号的也去掉,因为钱总不能是负的吧.下面我们要加的是说明可能的小数部分: `^[0-9]+(.[0-9]+)?$`

20 5).必须说明的是,小数点后面至少应该有1位数,所以"10."是不通过的,但是 "10" 和 "10.2" 是通过的: `^[0-9]+(.[0-9]{2})?$`

21 6).这样我们规定小数点后面必须有两位,如果你认为太苛刻了,可以这样: `^[0-9]+(.[0-9]{1,2})?$`

22 7).这样就允许用户只写一位小数.下面我们该考虑数字中的逗号了,我们可以这样: `^[0-9]{1,3}([0-9]{3})*(. [0-9]{1,2})?$`

23 8). 1到3个数字,后面跟着任意个 逗号+3个数字,逗号成为可选,而不是必须: `^([0-9]+|[0-9]{1,3}([0-9]{3})*)(.[0-9]{1,2})?$`

24 备注:这就是最终结果了,别忘了"+"可以用"*"替代如果你觉得空字符串也可以接受的话(奇怪,为什么?)最后,别忘了在用函数时去掉去掉那个反斜杠,一般的错误都在这里

25 xml文件: `^([a-zA-Z]+-?)+[a-zA-Z0-9]+\.[x|X][m|M][l|L]$`

26 中文字符的正则表达式: `[\u4e00-\u9fa5]`

27 双字节字符: `[\x00-\xff]` (包括汉字在内,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))

28 空白行的正则表达式: `\n\s*\r` (可以用来删除空白行)

29 HTML标记的正则表达式: `<(\S*?)[^>]*>.*?</\1>|<.*? />` (网上流传的版本太糟糕,上面这个也仅仅能部分,对于复杂的嵌套标记依旧无能为力)

30 首尾空白字符的正则表达式: `^\s*|\s*$或(^s*)|(\s*$)` (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)

31 腾讯QQ号: `[1-9][0-9]{4,}` (腾讯QQ号从10000开始)

32 中国邮政编码: `[1-9]\d{5}(?! \d)` (中国邮政编码为6位数字)

33 IP地址: `\d+\.\d+\.\d+\.\d+` (提取IP地址时有用)

34 IP地址: `((?:25[0-5]|2[0-4]\d|[01]?\d?\d)\.){3}(?:25[0-5]|2[0-4]\d|[01]?\d?\d?\d)`