

最全的常用正则表达式大全——包括校验数字、字符、一些特殊的需求等等

2015-12-18 博客园-zxin

很多不太懂正则的朋友，在遇到需要用正则校验数据时，往往是在网上去找很久，结果找来的还是不很符合要求。所以我最近把开发中常用的一些正则表达式整理了一下，在这里分享一下。给自己留个底，也给朋友们做个参考。

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}$`
- 3、长度为3-20的所有字符：`^. {3,20}$`
- 3、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_]+$`
- 10、中文、英文、数字但不包括下划线等符号：`^[\\u4E00-\\u9FA5A-Za-z0-9]{2,20}$`
- 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}$`
- 5、电话号码("XXX-XXXXXXX"、"XXXX-XXXXXXXX"、"XXX-XXXXXXX"、"XXX-XXXXXXXXXX"、"XXXXXXXX"和"XXXXXXXXX")：`^(\\(\\d{3,4}-\\d{3,4})?\\d{7,8}$`
- 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:

- 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]*$`
- 这表示任意一个不以0开头的数字,但是,这也意味着一个字符"0"不通过,所以我们采用下面的形式: `^(0|[1-9][0-9]*)$`
- 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]*)$`
- 这表示一个0或者一个可能为负的开头不为0的数字.让用户以0开头好了.把负号的也去掉,因为钱总不能是负的吧.下面我们要加的是说明可能的小数部分: `^[0-9]+(.[0-9]+)?$`
- 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})?$`
- 6.这样我们规定小数点后面必须有两位.如果你认为太苛刻了,可以这样: `^[0-9]+(.[0-9]{1,2})?$`
- 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})?$`
- 1到3个数字,后面跟着任意个 逗号+3个数字,逗号成为可选,而不是必须: `^([0-9]+|[0-9]{1,3}(.[0-9]{3})*)(.[0-9]{1,2})?$`
- 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

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

17、中文字符的正则表达式: `[\u4e00-\u9fa5]`

18、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))

19、空白行的正则表达式: `\n\s*\r` (可以用来删除空白行)

20、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)

21、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)

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

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

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

25、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))`