



作者

wujunze 2015.11.11 19:48:00

写了41篇文章,回复382人,

PHP常用正则表达式

评论: 2 · 阅读: 25035 · 喜欢: 1

1. 正则表达式

是一大利器,参数验证,数据采集等很多地方都用到. 许多程序设计语言都支持利用正则表达式进行字符串操作。平常写程序经常用到正则表达式,就整理一些常用的正则表达式分享一下!

1. 平时写程序经常要用正则表达式, 下面是一些讲解和例子, 仅供大家参考和修改使用:

2. `"^\\d+$"` //非负整数 (正整数 + 0)

3. `"^[0-9][1-9][0-9]$"` //正整数

4. `"^((-\\d+)|(0+))$"` //非正整数 (负整数 + 0)

5. `"^-[0-9][1-9][0-9]$"` //负整数

6. `"^-[0-9]$"` //整数

7. `"^\\d+(\\.\\d+)?$"` //非负浮点数 (正浮点数 + 0)

8. `"^([0-9]+.[0-9][1-9][0-9]|([0-9][1-9][0-9].[0-9]+)|([0-9][1-9][0-9]))$"` //正浮点数

9. `"^((-\\d+(\\.\\d+)?)(0+(.0+)?))$"` //非正浮点数 (负浮点数 + 0)

10. `"^(-([0-9]+.[0-9][1-9][0-9]|([0-9][1-9][0-9].[0-9]+)|([0-9][1-9][0-9])))"` //负浮点数

11. `"^(-?\\d+)(\\.\\d+)?$"` //浮点数

12. `"^[A-Za-z]+$"` //由26个英文字母组成的字符串

13. `"^[A-Z]+$"` //由26个英文字母的大写组成的字符串

14. `"^[a-z]+$"` //由26个英文字母的小写组成的字符串

15. `"^[A-Za-z0-9]+$"` //由数字和26个英文字母组成的字符串

16. `"^\\w+$"` //由数字、26个英文字母或者下划线组成的字符串

17. `"^[\\w-]+(\\.\\w-+)*@[\\w-]+(\\.\\w-+)+$"` //email地址

18. `"^[a-zA-Z]+://(\\w+(-\\w+))\\.((\\w+(-\\w+)))(?|S)?$"` //url

20. `/^((0|[1-9]{1})|([1|2]))/((0-2)|([3|0|1]))/(d{2}|d{4})$/` // 月/日/年

21. `"^([w-.]+)@(((0-9){1,3}.[0-9]{1,3}.[0-9]{1,3}.)|((w-)+.))([a-zA-Z]{2,4}|[0-9]{1,3})([?])$"` //Emi
l

22. `/^((+?[0-9]{2,4}-[0-9]{3,4}-)|([0-9]{3,4}-))?[0-9]{7,8}(-[0-9]+)?$/` // 电话号码

23. `"^(d{1,2}|1dd|2[0-4]d|25[0-5]).(d{1,2}|1dd|2[0-4]d|25[0-5]).(d{1,2}|1dd|2[0-4]d|25[0-5]).(d{1,2}|1dd|2[0-4]d|25[0-5])$"` //IP地址

25. 匹配中文字符的正则表达式: `[\u4e00-\u9fa5]`

26. 匹配双字节字符(包括汉字在内): `[\x00-\xff]`

27. 匹配空行的正则表达式: `\n[\s| ]*\r`

28. 匹配HTML标记的正则表达式: `/<(.)>.</\1>|<(.* )> /> /`

29. 匹配首尾空格的正则表达式: `(^\s)|(\s$)`

30. 匹配Email地址的正则表达式: `\w+([-+.]w+)@w+([-./]w+).w+([-.]w+)*`

31. 匹配网址URL的正则表达式: `^[a-zA-z]+://(\w+(-\w+))|.(\w+(-\w+))(|?|S)?$`

32. 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}$`

33. 匹配国内电话号码: `(\d{3}-|\d{4}-)?(\d{8}|\d{7})?`

34. 匹配腾讯QQ号: `^[1-9]/[1-9]/[0-9]$`

元字符及其在正则表达式上下文中的行为:

`\` 将下一个字符标记为一个特殊字符、或一个原义字符、或一个后向引用、或一个八进制转义符。

`^` 匹配输入字符串的开始位置。如果设置了 `RegExp` 对象的 `Multiline` 属性, `^` 也匹配 '`\n`' !

`$` 匹配输入字符串的结束位置。如果设置了 `RegExp` 对象的 `Multiline` 属性, `$` 也匹配 '`\n`' !

`*` 匹配前面的子表达式零次或多次。

`+` 匹配前面的子表达式一次或多次。`+` 等价于 `{1,}`。

`?` 匹配前面的子表达式零次或一次。`?` 等价于 `{0,1}`。

`{n}` `n` 是一个非负整数, 匹配确定的 `n` 次。

`{n,}` `n` 是一个非负整数, 至少匹配 `n` 次。

`{n,m}` `m` 和 `n` 均为非负整数, 其中 `n <= m`。最少匹配 `n` 次且最多匹配 `m` 次。在逗号和两个数之间不能有空格。

? 当该字符紧跟在任何一个其他限制符 (`*`, `+`, `?`, `{n}`, `{n,}`, `{n,m}`) 后面时, 匹配模式是非贪婪的。非贪婪模式尽可能少的匹配所搜索的字符串, 而默认的贪婪模式则尽可能多的匹配所搜索的字符串。

`.` 匹配除 "`\n`" 之外的任何单个字符。要匹配包括 '`\n`' 在内的任何字符, 请使用象 '`[\.\n]`' 的 `(pattern)` 匹配 `pattern` 并获取这一匹配。

`(?:pattern)` 匹配 `pattern` 但不获取匹配结果, 也就是说这是一个非获取匹配, 不进行存储供以后使用。

`(?=pattern)` 正向预查, 在任何匹配 `pattern` 的字符串开始处匹配查找字符串。这是一个非获取匹配, 因为引擎只会先检查是否满足后面的预查, 而不是将捕获的内容返回给 `RegExp` 对象的 `exec` 方法。

`(?!pattern)` 负向预查, 与 `(?=pattern)` 作用相反

输入关键字搜索

```
12 [xyz] 字符集合。
13
14 [^xyz] 负值字符集合。
15
16 [a-z] 字符范围，匹配指定范围内的任意字符。
17
18 [^a-z] 负值字符范围，匹配任何不在指定范围内的任意字符。
19
20 \b 匹配一个单词边界，也就是指单词和空格间的位置。
21
22 \B 匹配非单词边界。
23
24 \cx 匹配由x指明的控制字符。
```

\d 匹配一个数字字符。等价于 [0-9]。

\D 匹配一个非数字字符。等价于 [^0-9]。

```
1 \f 匹配一个换页符。等价于 \x0c 和 \cL。
```

\n 匹配一个换行符。等价于 \x0a 和 \cJ。

```
1 \r 匹配一个回车符。等价于 \x0d 和 \cM。
2
3 \s 匹配任何空白字符，包括空格、制表符、换页符等等。等价于 [ \f\n\r\t\v]。
4
5 \S 匹配任何非空白字符。等价于 [^ \f\n\r\t\v]。
```

\t 匹配一个制表符。等价于 \x09 和 \cI。

\v 匹配一个垂直制表符。等价于 \x0b 和 \cK。

```
1 \w 匹配包括下划线的任何单词字符。等价于 '[A-Za-z0-9_]'。
```

\W 匹配任何非单词字符。等价于 '[^A-Za-z0-9_]'。

```
1 \xn 匹配 n，其中 n 为十六进制转义值。十六进制转义值必须为确定的两个数字长。
2
3 \num 匹配 num，其中num是一个正整数。对所获取的匹配的引用。
```

\n 标识一个八进制转义值或一个后向引用。如果 \n 之前至少 n 个获取的子表达式，则 n 为后向引用。否则，如果 n 为八进制数字 (0-7)，则 n 为一个八进制转义值。

```
1 \nm 标识一个八进制转义值或一个后向引用。如果 \nm 之前至少有is preceded by at least
2
3 \nml 如果 n 为八进制数字 (0-3)，且 m 和 l 均为八进制数字 (0-7)，则匹配八进制转义值 r
4
5 \un 匹配 n，其中 n 是一个用四个十六进制数字表示的Unicode字符。
6
7 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]

```
1 匹配空行的正则表达式： n[s| ]*r
2
3 匹配HTML标记的正则表达式： /<(.*?)>.*</1>|<(.*?) />/
```

匹配首尾空格的正则表达式： (^s)/(\$s)

```
1 匹配Email地址的正则表达式： w+([-+.]w+)*@w+([-+.]w+)*.w+([-+.]w+)*
```

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, ''

用正则表达式限制只能输入全角字符： onkeyup="value=value.replace(/[^\uFF00-\uFFFF]

用正则表达式限制只能输入数字：onkeyup="value=value.replace(/[^\d]/g, '') "onbefo:

用正则表达式限制只能输入数字和英文：onkeyup="value=value.replace(/[W]/g, '') "onl

=====常用正则式

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： /<(.*>.*<\/\1>|<(.*) \/>/

匹配首尾空格的正则表达式： (^\s*) | (\s*\$)

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

匹配Email地址的正则表达式： \w+([-+.]\w+)*@\w+([-.\]\w+)*\.\w+([-.\]\w+)*

匹配网址URL的正则表达式： http://([\w-]+\.)+[\w-]+(/[\w- ./?%&=]*)?

sql语句： ^(select|drop|delete|create|update|insert).*

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： ^(-(正浮点数正则式))\$

10、英文字符串： ^[A-Za-z]+\$

11、英文大写串： ^[A-Z]+\$

12、英文小写串： ^[a-z]+\$

13、英文字符数字串： ^[A-Za-z0-9]+\$

14、英数字加下划线串： ^\w+\$

15、E-mail地址： ^[\w-]+(\. [\w-]+)*@[\w-]+(\. [\w-]+)+

输入关键字搜索

17、邮政编码: `^[1-9]\d{5}$`

1 18、中文: `^[\u0391-\uFFFE5]+$`

2

3 19、电话号码: `^((\(\d{2,3}\)|(\d{3}-))?(0\d{2,3}|0\d{2,3}-)?[1-9]\d{6,`

20、手机号码: `^(((\d{2,3})|(\d{3}-))?)13\d{9}$`

1 21、双字节字符 (包括汉字在内): `^\x00-\xff`

2

3 22、匹配首尾空格: `(^\s*)|(\s*$)` (像vbscript那样的trim函数)

4

5 23、匹配HTML标记: `<(.*?)>.*</\1>|<(.*?) \/>`

6

7 24、匹配空行: `\n[\s| ]*\r`

8

9 25、提取信息中的网络链接: `(h|H)(r|R)(e|E)(f|F) *= *('|")?(\w|\||\|/|\.)+('|"|`

10

11 26、提取信息中的邮件地址: `\w+([-+.] \w+)*@\w+([-.] \w+)*\.\w+([-.] \w+)*`

12

13 27、提取信息中的图片链接: `(s|S)(r|R)(c|C) *= *('|")?(\w|\||\|/|\.)+('|"|`

14

15 28、提取信息中的IP地址: `(\d+)\.(\d+)\.(\d+)\.(\d+)`

16

17 29、提取信息中的中国手机号码: `(86)*0*13\d{9}`

18

19 30、提取信息中的中国固定电话号码: `(\(\d{3,4}\)|\d{3,4}-|\s)?\d{8}`

20

21 31、提取信息中的中国电话号码 (包括移动和固定电话): `(\(\d{3,4}\)|\d{3,4}-|\s)?\d{7`

22

23 32、提取信息中的中国邮政编码: `[1-9]{1}(\d+){5}`

24

25 33、提取信息中的浮点数 (即小数): `(-?\d+)\.?\d+`

26

27 34、提取信息中的任何数字: `(-?\d+)(\.\d+)?`

28

29 35、IP: `(\d+)\.(\d+)\.(\d+)\.(\d+)`

30

31 36、电话区号: `/^0\d{2,3}$/`

32

33 37、腾讯QQ号: `^[1-9]*[1-9][0-9]*$`

34

35 38、帐号 (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}`

36

37 39、中文、英文、数字及下划线: `^[\u4e00-\u9fa5_a-zA-Z0-9]+$`

❤ 赞 | 1

赏

标签: [php](#) [正则表达式](#) [php正则表达式](#)

2 条评论

✎ 添加新评论



Arrrrrrr November 18th, 2015 at 02:08 pm

回复

高大上...



wujunze 博主



November 19th, 2015 at 10:40 am

输入关键字搜索