

吴 Z 静

永久域名 <http://wuzijingaip.iteye.com>



wuzijingaip

浏览: 59016 次

性别: ♀

来自: 广州

我现在离线

[详细资料](#) [留言簿](#)

搜索本博客

最近访客 [>>更多访客](#)



[gao911111](#)



[tianshiyeben](#)



[褚晓娜](#)
(0511)



[wss71104307](#)

博客分类

- [全部博客 \(194\)](#)

我的相册



4

[共 6 张](#)

我的留言簿 [>>更多留言](#)

其他分类

- [我的收藏](#) (32)
- [我的代码](#) (0)
- [我的论坛主题帖](#) (0)

2009-11-05

◀ [建立 U 盘 \(USB\) 启动的 Linux - SLAX](#) | [java JFreeChart 中文文档](#) ▶

[mysql 正则表达式](#)

[正则表达式](#) [MySQL](#) [C](#) [C++](#) [C#](#)

正则表达式 (regex)是定义复杂查询的一个强有力的工具。

这里是一个简单的资料，它忽略了一些详细的信息。

正则表达式定义了一个字符串的规则。最简单的正则表达式不包含任何保留字。例如，正则表达式hello只和字符串“hello”匹配。

一般的正则表达式使用了某些特殊的结构，所以它能匹配更多的字符串。例如，正则表达式hello|word既能匹配字符串“hello”也能匹配字符串 “word”。举一个更复杂一点的例子，正则表达式B[an]*s可以匹配字符串“Bananas”、“Baaaaas”、“Bs”以及其他任何以B开头以s结尾的字符串，中间可以包括任意个a和任意个n的组合。

一个正则表达式中的可以使用以下保留字

^

所匹配的字符串以后面的字符串开头

mysql> select "fonfo" REGEXP "^fo\$"; -> 0 (表示不匹配)

mysql> select "fofo" REGEXP "fo"; -> 1 (表示匹配)

\$

所匹配的字符串以前面的字符串结尾

mysql> select "fono" REGEXP "fono\$"; -> 1 (表示匹配)

mysql> select "fono" REGEXP "fo\$"; -> 0 (表示不匹配)

.

匹配任何字符 (包括新行)

mysql> select "fofo" REGEXP "f.*"; -> 1 (表示匹配)

[我的所有论坛帖](#) (0)

- [我的精华良好帖](#) (0)

最近加入群组

- [EXT](#)

存档

- [2011-09](#) (3)
- [2011-08](#) (2)
- [2011-06](#) (5)
- [更多存档...](#)

评论排行榜

- [tomcat 中运行php](#)
- [php 网站访问量统计](#)



mysql> select "fonfo" REGEXP "^f.*"; -> 1 (表示匹配)

a*

匹配任意多个a (包括空串)

mysql> select "Ban" REGEXP "^Ba*n"; -> 1 (表示匹配)

mysql> select "Baaan" REGEXP "^Ba*n"; -> 1 (表示匹配)

mysql> select "Bn" REGEXP "^Ba*n"; -> 1 (表示匹配)

a+

匹配任意多个a (不包括空串)

mysql> select "Ban" REGEXP "^Ba+n"; -> 1 (表示匹配)

mysql> select "Bn" REGEXP "^Ba+n"; -> 0 (表示不匹配)

a?

匹配一个或零个a

mysql> select "Bn" REGEXP "^Ba?n"; -> 1 (表示匹配)

mysql> select "Ban" REGEXP "^Ba?n"; -> 1 (表示匹配)

mysql> select "Baan" REGEXP "^Ba?n"; -> 0 (表示不匹配)

de|abc

匹配de或abc

mysql> select "pi" REGEXP "pi|apa"; -> 1 (表示匹配)

mysql> select "axe" REGEXP "pi|apa"; -> 0 (表示不匹配)

mysql> select "apa" REGEXP "pi|apa"; -> 1 (表示匹配)

mysql> select "apa" REGEXP "^ (pi|apa)\$"; -> 1 (表示匹配)

mysql> select "pi" REGEXP "^ (pi|apa)\$"; -> 1 (表示匹配)

mysql> select "pix" REGEXP "^ (pi|apa)\$"; -> 0 (表示不匹配)

(abc)*

匹配任意多个abc（包括空串）

mysql> select "pi" REGEXP "^(pi)*\$"; -> 1（表示匹配）

mysql> select "pip" REGEXP "^(pi)*\$"; -> 0（表示不匹配）

mysql> select "pipi" REGEXP "^(pi)*\$"; -> 1（表示匹配）

{1}

{2,3}

这是一个更全面的方法，它可以实现前面好几种保留字的功能

a*

可以写成**a{0,}**

a+

可以写成**a{1,}**

a?

可以写成**a{0,1}**

在{}内只有一个整型参数i，表示字符只能出现i次；在{}内有一个整型参数i，后面跟一个“，”，表示字符可以出现i次或i次以上；在{}内只有一个整型参数i，后面跟一个“，”，再跟一个整型参数j,表示字符只能出现i次以上，j次以下（包括i次和j次）。其中的整型参数必须大于等于0，小于等于 RE_DUP_MAX（默认是255）。如果有两个参数，第二个必须大于等于第一个

[a-dX]

匹配“a”、“b”、“c”、“d”或“X”

[^a-dX]

匹配除“a”、“b”、“c”、“d”、“X”以外的任何字符。“[”、“]”必须成对使用

mysql> select "aXbc" REGEXP "[a-dXYZ]"; -> 1（表示匹配）

mysql> select "aXbc" REGEXP "^[a-dXYZ]\$"; -> 0（表示不匹配）

mysql> select "aXbc" REGEXP "^[a-dXYZ]+\$"; -> 1（表示匹配）

mysql> select "aXbc" REGEXP "^a-dXYZ]+\$"; -> 0 (表示不匹配)

mysql> select "gheis" REGEXP "^a-dXYZ]+\$"; -> 1 (表示匹配)

mysql> select "gheisa" REGEXP "^a-dXYZ]+\$"; -> 0 (表示不匹配)

[[.characters.]]

表示比较元素的顺序。在括号内的字符顺序是唯一的。但是括号中可以包含通配符, 所以他能匹配更多的字符。举例来说: 正则表达式[[.ch.]]*c匹配chchccc的前五个字符。

[=character_class=]

表示相等的类, 可以代替类中其他相等的元素, 包括它自己。例如, 如果o和(+)是一个相等的类的成员, 那么[[=o=]]、[[=(+)=]]和[o(+)]是完全等价的。

[:character_class:]

在括号里面, 在[:和:]中间是字符类的名字, 可以代表属于这个类的所有字符。

字符类的名字有: alnum、digit、punct、alpha、graph、space、blank、lower、upper、cntrl、print和xdigit

mysql> select "justalnums" REGEXP "[[:alnum:]]+"; -> 1 (表示匹配)

mysql> select "!!" REGEXP "[[:alnum:]]+"; -> 0 (表示不匹配)

[::]

分别匹配一个单词开头和结尾的空字符串, 这个单词开头和结尾都不是包含在alnum中的字符也不能是下划线。

mysql> select "a word a" REGEXP "[[::]]"; -> 1 (表示匹配)

mysql> select "a xword a" REGEXP "[[::]]"; -> 0 (表示不匹配)

mysql> select "weeknights" REGEXP "^(wee|week)(knights|nights)\$"; -> 1 (表示匹配)

分享到:



◀ [建立 U 盘 \(USB\) 启动的 Linux - SLAX](#) | [java JFreeChart 中文文档](#) ▶

17:17 | [评论 / 浏览 \(0 / 102\)](#) | [相关推荐](#) [▶ MORE](#)

评论

发表评论



[您还没有登录,请您登录后再发表评论](#)

声明：ITeye 文章版权属于作者，受法律保护。没有作者书面许可不得转载。若作者同意转载，必须以超链接形式标明文章原始出处和作者。
© 2003-2011 ITeye.com. All rights reserved. [京ICP证110151号]