

Shell正则表达式

正则表达式的分类

- 1. 基本的正则表达式 (Basic Regular Expression 又叫Basic RegEx 简称BRES)
- 2. 扩展的正则表达式 (Extended Regular Expression 又叫Extended RegEx 简称EREs)
- 3. Perl的正则表达式 (Perl Regular Expression 又叫Perl RegEx 简称PREs)

基本组成部分

正则表达式的基本组成部分。

正则表达式	描述	示例	Basic RegEx	Extended RegEx	Python RegEx
\	转义符,将特殊字符进行转义,忽略其特殊意义	a\.b匹配a.b,但不能匹配ajb, .被转义为特殊意义	\	\	\
^	匹配行首,awk中,^则是匹配字符串的开始	^tux匹配以tux开头的行	^	^	^
\$	匹配行尾,awk中,\$则是匹配字符串的结尾	tux\$匹配以tux结尾的行	\$	\$	\$
.	匹配除换行符\n之外的任意单个字符,awk则中可以	ab.匹配abc或bad,不可匹配abcd或abde,只能匹配单字符	.	.	.
[]	匹配包含在[字符]之中的任意一个字符	coo[kl]可以匹配cook或cool	[]	[]	[]
[^]	匹配[^字符]之外的任意一个字符	123[^45]不可以匹配1234或1235,1236、1237都可以	[^]	[^]	[^]
[-]	匹配[]中指定范围内的任意一个字符,要写成递增	[0-9]可以匹配1、2或3等其中任意一个数字	[-]	[-]	[-]
?	匹配之前的项1次或者0次	colou?r可以匹配color或者colour,不能匹配colouur	不支持	?	?
+	匹配之前的项1次或者多次	sa-6+匹配sa-6、sa-666,不能匹配sa-	不支持	+	+
*	匹配之前的项0次或者多次	co*1匹配c1、col、cool、coool等	*	*	*
()	匹配表达式,创建一个用于匹配的子串	ma(tri)?匹配max或maxtrix	不支持	()	()
{n}	匹配之前的项n次,n是可以为0的正整数	[0-9]{3}匹配任意一个三位数,可以扩展为[0-9]{0-9}[0-9]{0-9}	不支持	{n}	{n}
{n,}	之前的项至少需要匹配n次	[0-9]{2,}匹配任意一个两位数或更多位数	不支持	{n,}	{n,}
{n,m}	指定之前的项至少匹配n次,最多匹配m次,n<=m	[0-9]{2,5}匹配从两位数到五位数之间的任意一个数字	不支持	{n,m}	{n,m}
	交替匹配 两边的任意一项	ab(c d)匹配abc或abd	不支持		

POSIX字符类

POSIX字符类是一个形如[:...:]的特殊元序列 (meta sequence), 他可以用于匹配特定的字符范围。

正则表达式	描述	示例	Basic RegEx	Extended RegEx	Python RegEx
[:alnum:]	匹配任意一个字母或数字字符	[[[:alnum:]]]+	[:alnum:]	[:alnum:]	[:alnum:]
[:alpha:]	匹配任意一个字母字符 (包括大小写字母)	[[[:alpha:]]]{4}	[:alpha:]	[:alpha:]	[:alpha:]
[:blank:]	空格与制表符 (横向和纵向)	[[[:blank:]]]*	[:blank:]	[:blank:]	[:blank:]
[:digit:]	匹配任意一个数字字符	[[[:digit:]]]?	[:digit:]	[:digit:]	[:digit:]
[:lower:]	匹配小写字母	[[[:lower:]]]{5,}	[:lower:]	[:lower:]	[:lower:]
[:upper:]	匹配大写字母	(([[[:upper:]]]+)?)	[:upper:]	[:upper:]	[:upper:]
[:punct:]	匹配标点符号	[[[:punct:]]]	[:punct:]	[:punct:]	[:punct:]
[:space:]	匹配一个包括换行符、回车等在内的所有空白符	[[[:space:]]]+	[:space:]	[:space:]	[:space:]
[:graph:]	匹配任何一个可以看得见的且可以打印的字符	[[[:graph:]]]	[:graph:]	[:graph:]	[:graph:]
[:xdigit:]	任何一个十六进制数 (即: 0-9, a-f, A-F)	[[[:xdigit:]]]+	[:xdigit:]	[:xdigit:]	[:xdigit:]
[:cntrl:]	任何一个控制字符 (ASCII 字符集中的前32个字符)	[[[:cntrl:]]]	[:cntrl:]	[:cntrl:]	[:cntrl:]
[:print:]	任何一个可以打印的字符	[[[:print:]]]	[:print:]	[:print:]	[:print:]

元字符

元字符 (meta character) 是一种Perl风格的正则表达式, 只有一部分文本处理工具支持它, 并不是所有的文本处理工具都支持。

正则表达式	描述	示例	Basic RegEx	Extended RegEx	Python RegEx
\b	单词边界	\bcool\b 匹配cool, 不匹配coolant	\b	\b	\b
\B	非单词边界	cool\B 匹配coolant, 不匹配cool	\B	\B	\B
\d	单个数字字符	b\d\b 匹配b2b, 不匹配bcb	不支持	不支持	\d
\D	单个非数字字符	b\D\b 匹配bcb, 不匹配b2b	不支持	不支持	\D
\w	单个单词字符 (字母、数字与_)	\w 匹配1或a, 不匹配&	\w	\w	\w
\W	单个非单词字符	\W 匹配&, 不匹配1或a	\W	\W	\W
\n	换行符	\n 匹配一个新行	不支持	不支持	\n

\s	单个空白字符	x\sx 匹配x x，不匹配xx	不支持	不支持	\s
\S	单个非空白字符	x\Sx 匹配xx x，不匹配xx	不支持	不支持	\S
\r	回车	\r 匹配回车	不支持	不支持	\r
\t	横向制表符	\t 匹配一个横向制表符	不支持	不支持	\t
\v	垂直制表符	\v 匹配一个垂直制表符	不支持	不支持	\v
\f	换页符	\f 匹配一个换页符	不支持	不支持	\f

上面的内容只是一些介绍，下面脚本之家小编给大家整理一些文章，大家可以看下实例。

[Shell正则表达式之grep、sed、awk实操笔记](#)

[shell 正则表达式详细整理](#)

[Linux Shell脚本的编程之正则表达式](#)

[详解Linux命令中的正则表达式](#)

[详解Linux--shell脚本之正则表达式](#)

返回[shouce.jb51.net](#)