

1. BASH 特殊符号意义

“ ”	双引号，特殊字符脱意符，但能保持变量特性。
‘ ’	单引号，特殊字符脱意符，引号内被作为一般字符。
``	反引号，引用命令执行的结果。
\	反斜杠，特殊字符脱意符。
~	波浪号，表示用户家目录。
-	短横杠，多种意义。上一次工作目录；用于重定向的标准输出和输入。
!	感叹号，多种意义。如历史命令调用；取反。
\$?	上一个命令执行的结果返回值。（true 为0，false 为非0）。
\$A	取出 A 的变量值。
\$()	引用，与反引号相同。可以嵌套。
\$[]	数学运算。（+加-减*乘/除%取余**幂=等于）
\${}	取出变量值，花括号起控制变量名长度的作用。
@	邮件分隔符。
#	注释符。
%	作业控制。
&	作业后台运行。
&&	命令逻辑与连接。只有当前面的命令返回值为真时才会执行后面的命令。
	命令逻辑或连接。只有当前面的命令返回值为假时才会执行后面的命令。
*	通配符，匹配零个或多个任意字符。
?	通配符，匹配一个任意字符。
()	组合命令，改变命令执行的顺序。
(())	数学运算。
[]	两边有空格时，与 test 命令相同，表示判断。
{}	多种意义。枚举；脚本中使用代码块；find 命令中表示搜索到的结果引用。
	管道功能。
:	冒号，内部命令，表示一个返回值为真的空命令。
;	分号，分隔多个连续执行的命令。
<	包括>,>>,<<,2>,&>，输入输出重定向，参照第三章。

.	圆点，有多种意义。第价于 source 命令；隐藏文件。
/	文件名路径分隔符。
=	字符串比较。

2. POSIX 字符类

特殊符号	代表意义
[:alnum:]	代表英文大小写字符及数字，亦即 0-9, A-Z, a-z
[:alpha:]	代表任何英文大小写字符，亦即 A-Z, a-z
[:blank:]	代表空白键与 [Tab] 按键两者
[:cntrl:]	代表键盘上面的控制按键，亦即包括 CR, LF, Tab, Del.. 等等
[:digit:]	代表数字而已，亦即 0-9
[:graph:]	除了空格符（空白键与 [Tab] 按键）外的其他所有按键
[:lower:]	代表小写字符，亦即 a-z
[:print:]	代表任何可以被打印出来的字符
[:punct:]	代表标点符号 (punctuation symbol)，亦即： " ' ? ! ; : # \$...
[:upper:]	代表大写字符，亦即 A-Z
[:space:]	任何会产生空白的字符，包括空白键，[Tab]，CR 等等
[:xdigit:]	代表 16 进位的数字类型，因此包括： 0-9, A-F, a-f 的数字与字符

3. 元字符

元字符	描述	示例
\b	单词边界	\bcool\b 匹配 cool，但不匹配 coolant
\B	非单词边界	cool\B 匹配 coolant,但不匹配 cool
\d	单个数字字符	b\db 匹配 b2b，但不匹配 bcb
\D	单个非数字字符	b\Db 匹配 bcb, 但不匹配 b2b
\w	单个单词字符（字母、数字与_）	\w 匹配 l或 a，但不匹配&
\W	单个非单词字符	\W 匹配&,但不匹配 l或 a
\n	换行符	\n 匹配一个新行
\s	单个空白字符	x\sx 匹配 x x ，但不匹配 xx
\S	单个非空白字符	x\Sx 匹配 xkx，但不匹配 xx
\r	回车	\r 匹配回车

4. 常规正则表达式

RE 字符	意义与范例
-------	-------

<code>^word</code>	<p>意义: 待搜寻的字符串(word)在行首!</p> <p>范例: 搜寻行首为 # 开始的那一行, 并列出行号</p> <pre>grep -n '^#' regular_express.txt</pre>
<code>word\$</code>	<p>意义: 待搜寻的字符串(word)在行尾!</p> <p>范例: 将行尾为 ! 的那一行打印出来, 并列出行号</p> <pre>grep -n '!\$' regular_express.txt</pre>
<code>.</code>	<p>意义: 代表『一定有一个任意字符』的字符!</p> <p>范例: 搜寻的字符串可以是 (eve) (eae) (eee) (e e), 但不能仅有 (ee) ! 亦即 e 与 e 中间『一定』仅有一个字符, 而空格符也是字符!</p> <pre>grep -n 'e.e' regular_express.txt</pre>
<code>\</code>	<p>意义: 跳脱字符, 将特殊符号的特殊意义去除!</p> <p>范例: 搜寻含有单引号 ' 的那一行!</p> <pre>grep -n \' regular_express.txt</pre>
<code>*</code>	<p>意义: 重复零个到无穷多个的前一个 RE 字符</p> <p>范例: 找出含有 (es) (ess) (esss) 等等的字符串, 注意, 因为 * 可以是 0 个, 所以 es 也是符合带搜寻字符串。另外, 因为 * 为重复『前一个 RE 字符』的符号, 因此, 在 * 之前必须要紧接着一个 RE 字符喔! 例如任意字符则为 [.*] !</p> <pre>grep -n 'ess*' regular_express.txt</pre>
<code>[list]</code>	<p>意义: 字符集合的 RE 字符, 里面列出想要撷取的字符!</p> <p>范例: 搜寻含有 (gl) 或 (gd) 的那一行, 需要特别留意的是, 在 [] 当中『谨代表一个待搜寻的字符』, 例如 [a[af]y] 代表搜寻的字符串可以是 aay, afy, aly 即 [af] 代表 a 或 f 或 l 的意思!</p> <pre>grep -n 'g[ld]' regular_express.txt</pre>
<code>[n1-n2]</code>	<p>意义: 字符集合的 RE 字符, 里面列出想要撷取的字符范围!</p> <p>范例: 搜寻含有任意数字的那一行! 需特别留意, 在字符集合 [] 中的减号 - 是有特殊意义的, 他代表两个字符之间的所有连续字符! 但这个连续与否与 ASCII 编码有关, 因此, 你的编码需要设定正确(在 bash 当中, 需要确定 LANG 与 LANGUAGE 的变量是否正确!) 例如所有大写字符则为 [A-Z]</p> <pre>grep -n '[A-Z]' regular_express.txt</pre>
<code>[^list]</code>	<p>意义: 字符集合的 RE 字符, 里面列出不要的字符串或范围!</p> <p>范例: 搜寻的字符串可以是 (oog) (ood) 但不能是 (oot), 那个 ^ 在 [] 内时, 代表的意义是『反向选择』的意思。例如, 我不要大写字符, 则为 [^A-Z]。但是, 需要特别注意的是, 如果以 <code>grep -n [^A-Z] regular_express.txt</code> 来搜寻, 却发现该档案内的所有行都被列出, 为什么? 因为这个 [^A-Z] 是『非大写字符』的意思, 因为每一行均有非大写字符, 例如第一行的 "Open Source" 就有 p, e, n, o... 等等的小写字</p> <pre>grep -n 'oo[^t]' regular_express.txt</pre>
<code>\{n,m\}</code>	<p>意义: 连续 n 到 m 个的『前一个 RE 字符』</p> <p>意义: 若为 \{n\} 则是连续 n 个的前一个 RE 字符,</p> <p>意义: 若是 \{n,\} 则是连续 n 个以上的前一个 RE 字符! 范例: 在 g 与 g 之间有 2 个到 3 个的 o 存在的字符串, 亦即 (goog) (gooog)</p> <pre>grep -n 'go\{2,3\}g' regular_express.txt</pre>

5. 扩展正则表达式

RE 字符	意义与范例
+	<p>意义: 重复『一个或一个以上』的前一个 RE 字符</p> <p>范例: 搜寻 (god) (good) (goood)... 等等的字符串。那个 o+ 代表『一个以上的 o』所以, 底下的执行成果会将第 1, 9, 13 行列出来。</p> <p>egrep -n 'go+d' regular_express.txt</p>
?	<p>意义: 『零个或一个』的前一个 RE 字符</p> <p>范例: 搜寻 (gd) (god) 这两个字符串。那个 o? 代表『空的或 1 个 o』所以, 上面的执行成果会将第 13, 14 行列出来。有没有发现到, 这两个案例('go+d' 与 'go?d')的结果集合与 'go*d' 相同? 想想看, 这是为什么喔! ^_^</p> <p>egrep -n 'go?d' regular_express.txt</p>
	<p>意义: 用或(or)的方式找出数个字符串</p> <p>范例: 搜寻 gd 或 good 这两个字符串, 注意, 是『或』! 所以, 第 1, 9, 14 这三行都可以被打印出来喔! 那如果还想要找出 dog 呢?</p> <p>egrep -n 'gd good' regular_express.txt egrep -n 'gd good dog' regular_express.txt</p>
()	<p>意义: 找出『群组』字符串</p> <p>范例: 搜寻 (glad) 或 (good) 这两个字符串, 因为 g 与 d 是重复的, 所以, 我就可以将 la 与 oo 列于 () 当中, 并以 来分隔开来, 就可以啦!</p> <p>egrep -n 'g(la oo)d' regular_express.txt</p>
()+	<p>意义: 多个重复群组的判别</p> <p>范例: 将『AxyzxyzxyzxyzC』用 echo 叫出, 然后再使用如下的方法搜寻一下!</p> <p>echo 'AxyzxyzxyzxyzC' egrep 'A(xyz)+C'</p> <p>上面的例子意思是说, 我要找开头是 A 结尾是 C, 中间有一个以上的 "xyz" 字符串的意思~</p>