Linux 单引号+双引号+转义符

【1】Linux 单引号+双引号+转义符

<https://blog.csdn.net/u013985241/article/details/83188308>

【2】Linux shell命令中双引号与单引号嵌套的问题

<https://www.csdn.net/gather_21/MtjaQg1sNTA3MS1ibG9n.html>

一、单引号""

目的: 为了保护文字不被转换.除了他本身. 就是说除去单引号外, 在单引号内的所有文字都是原样输出.

1. [root@jszwl161 SP49EP9]# echo '$\*><!'

$\*><!

2. [root@jszwl161 SP49EP9]# echo 'she is crying: "help"'

she is crying: "help"

3. [root@jszwl161 SP49EP9]# echo '\\\\'

\\\\

4. [root@jszwl161 SP49EP9]# echo 'hah 'test''

hah test #略去了所有'

5. [root@jszwl161 SP49EP9]# echo ' today is `date`'

today is `date` #反引号在此无法实现函数功能.

二、双引号""

目的: 为了包含文字或者函数段. 除了本身,反引号内的函数,$开头的变量和\开头反转换的字符外, 其余都是直接输出.

1. [root@jszwl161 SP49EP9]# echo "today is `date`"

today is Fri Jul 4 08:03:34 GMT 2008

2. [root@jszwl161 SP49EP9]# echo "today is 'date'"

today is 'date'

3. [root@jszwl161 SP49EP9]# echo "\\\\"

\\

4. [root@jszwl161 SP49EP9]# echo "test "test""

test test

三、反引号`

目的是为了在双引号内保持函数转换. 但单引号内其无作用.

1. [root@jszwl161 SP49EP9]# echo "today is `date`"

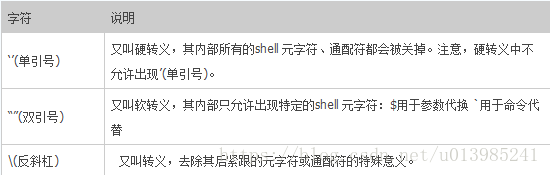
today is Fri Jul 4 08:03:34 GMT 2008 #将函数date转换.

2. [root@jszwl161 SP49EP9]# echo ' today is `date` '

today is `date` #在单引号内无作用.

四、转义符\

有时候，我们想让 通配符，或者元字符变成普通字符，不需要使用它。那么这里我们就需要用到转义符了。 shell提供转义符有三种。



\后跟的是非元字符，与没有加\的效果一样

eg：echo '\A' 和echo "\A" 的输出都是\A 而echo\A 的输出却是A

补充：

（1）linux shell通配符(wildcard)

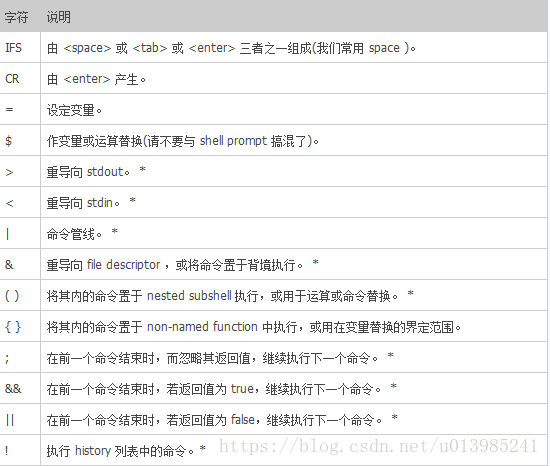
通配符是由shell处理的（不是由所涉及到命令语句处理的，其实我们在shell各个命令中也没有发现有这些通配符介绍）, 它只会出现在命令的“参数”里（它不用在命令名称里， 也不用在 操作符上）。当shell在“参数”中遇到了通配符时，shell会将其当作路径或文件名去在磁盘上搜寻可能的匹配：若符合要求的匹配存在，则进行代换(路径扩展)；否则就将该通配符作为一个普通字符传递给“命令”，然后再由命令进行处理。总之，通配符 实际上就是一种shell实现的路径扩展功能。在通配符被处理后, shell会先完成该命令的重组，然后再继续处理重组后的命令，直至执行该命令。



需要说明的是：通配符看起来有点象正则表达式语句，但是它与正则表达式不同的，不能相互混淆。把通配符理解shell特殊代号字符就可。而且涉及的只有，\*,? [] ,{} 这几种。

（2）shell元字符（特殊字符 Meta）

shell 除了有通配符之外，由shell 负责预先先解析后，将处理结果传给命令行之外，shell还有一系列自己的其他特殊字符。



加入”\*”都是作用在命令名直接。可以看到shell元字符，基本是作用在命令上面，用作多命令分割（或者参数分割）。因此看到与通配符有相同的字符，但是实际上作用范围不同。所以不会出现混淆。

小结：

**单引号：**被单引号用括住的内容，将被视为单一字串。在引号内的代表变数的$符号，没有作用，也就是说，他被视为一般符号处理，防止任何变量替换。

**双引号：**被双引号用括住的内容，将被视为单一字串。它防止通配符扩展，但允许变量扩展。这点与单引数的处理方式不同。

**反引号：**在前面的单双引号，括住的是字串，但如果该字串是一列命令列，会怎样？答案是不会执行。要处理这种情况，我们得用反单引号来做。

fdv=`date +%F`

echo "Today $fdv"

在倒引号内的 date +%F 会被视为指令，执行的结果会带入fdv变量中。