# Shell 变量字符串替换,剪切

## 替换语法

${a/123/321};将${a}里的第一个123替换为321

${a//123/321};将${a}里的所有123替换为321

例子：

FULLPATH="$1"

RENAMEPATH=${FULLPATH//\ /\_}

SUBPATH=${RENAMEPATH%.\*}

FILENAME=${SUBPATH##\*/}

SUBPATH=${SUBPATH%/\*}

假設我們定義了一個變量為：  
file=/dir1/dir2/dir3/my.file.txt  
我們可以用 ${ } 分別替換獲得不同的值：  
${file#\*/}：拿掉第一條 / 及其左邊的字串：dir1/dir2/dir3/my.file.txt  
${file##\*/}：拿掉最後一條 / 及其左邊的字串：my.file.txt  
${file#\*.}：拿掉第一個 . 及其左邊的字串：file.txt  
${file##\*.}：拿掉最後一個 . 及其左邊的字串：txt  
${file%/\*}：拿掉最後條 / 及其右邊的字串：/dir1/dir2/dir3  
${file%%/\*}：拿掉第一條 / 及其右邊的字串：(空值)  
${file%.\*}：拿掉最後一個 . 及其右邊的字串：/dir1/dir2/dir3/my.file  
${file%%.\*}：拿掉第一個 . 及其右邊的字串：/dir1/dir2/dir3/my  
記憶的方法為：

# 是去掉左邊(在鑑盤上 # 在 $ 之左邊)  
% 是去掉右邊(在鑑盤上 % 在 $ 之右邊)  
單一符號是最小匹配﹔兩個符號是最大匹配。

${file:0:5}：提取最左邊的 5 個字節：/dir1  
${file:5:5}：提取第 5 個字節右邊的連續 5 個字節：/dir2  
  
我們也可以對變量值裡的字串作替換：  
${file/dir/path}：將第一個 dir 提換為 path：/path1/dir2/dir3/my.file.txt  
${file//dir/path}：將全部 dir 提換為 path：/path1/path2/path3/my.file.txt  
  
利用 ${ } 還可針對不同的變數狀態賦值(沒設定、空值、非空值)：   
${file-my.file.txt} ：假如 $file 沒有設定，則使用 my.file.txt 作傳回值。(空值及非空值時不作處理)   
${file:-my.file.txt} ：假如 $file 沒有設定或為空值，則使用 my.file.txt 作傳回值。 (非空值時不作處理)  
${file+my.file.txt} ：假如 $file 設為空值或非空值，均使用 my.file.txt 作傳回值。(沒設定時不作處理)  
${file:+my.file.txt} ：若 $file 為非空值，則使用 my.file.txt 作傳回值。 (沒設定及空值時不作處理)  
${file=my.file.txt} ：若 $file 沒設定，則使用 my.file.txt 作傳回值，同時將 $file 賦值為 my.file.txt 。 (空值及非空值時不作處理)  
${file:=my.file.txt} ：若 $file 沒設定或為空值，則使用 my.file.txt 作傳回值，同時將 $file 賦值為 my.file.txt 。 (非空值時不作處理)  
${file?my.file.txt} ：若 $file 沒設定，則將 my.file.txt 輸出至 STDERR。 (空值及非空值時不作處理)  
${file:?my.file.txt} ：若 $file 沒設定或為空值，則將 my.file.txt 輸出至 STDERR。 (非空值時不作處理)  
  
tips:  
以上的理解在於, 你一定要分清楚 unset 與 null 及 non-null 這三種賦值狀態.  
一般而言, : 與 null 有關, 若不帶 : 的話, null 不受影響, 若帶 : 則連 null 也受影響.  
  
  
還有哦，${#var} 可計算出變量值的長度：  
${#file} 可得到 27 ，因為 /dir1/dir2/dir3/my.file.txt 剛好是 27 個字節...