:(冒号)命令，即空命令的用法

:等价于"NOP" (no op, 一个什么也不干的命令)，也可以被认为与shell的内建命令true作用相同。":"命令是一个bash的内建命令, 它的退出码(exit status)是"true"(0)

|  |
| --- |
| :  echo $? #输出为 0 |

**变量长度/子串删除**

${#var}

字符串长度(变量$var得字符个数). 对于[array](mk:@MSITStore:C:\\DOCUME~1\\ADMINI~1\\LOCALS~1\\Temp\\Rar$DI00.296\\高级bash脚本编程指南.CHM::/arrays.html" \l "ARRAYREF)来说, **${#array}**表示的是数组中第一个元素的长度.

|  |  |
| --- | --- |
|  | 例外情况:   * **${#\*}**和**${#@}**表示*位置参数的个数*. * 对于数组来说, **${#array[\*]}**和**${#array[@]}**表示数组中元素的个数. |

**例子 9-17. 变量长度**

|  |
| --- |
| 1 #!/bin/bash  2 # length.sh  3  4 E\_NO\_ARGS=65  5  6 if [ $# -eq 0 ] # 这个演示脚本必须有命令行参数.  7 then  8  echo "Please invoke this script with one or more command-line arguments."  9  exit $E\_NO\_ARGS  10 fi  11  12 var01=abcdEFGH28ij  13 echo "var01 = ${var01}"  14 echo "Length of var01 = ${#var01}"  15 # 现在, 让我们试试在变量中嵌入一个空格.  16 var02="abcd EFGH28ij"  17 echo "var02 = ${var02}"  18 echo "Length of var02 = ${#var02}"  19  20 echo "Number of command-line arguments passed to script = ${#@}"  21 echo "Number of command-line arguments passed to script = ${#\*}"  22  23 exit 0 |

${var#Pattern}, ${var##Pattern}

从变量$var的*开头*删除最短或最长匹配$Pattern的子串. (译者注: 这是一个很常见的用法, 请读者牢记, 一个"#"表示匹配最短, "##"表示匹配最长.)

[例子 A-7](mk:@MSITStore:C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\Rar$DI00.296\高级bash脚本编程指南.CHM::/contributed-scripts.html#DAYSBETWEEN)中的一个用法示例:

|  |
| --- |
| 1 # 摘自例子"days-between.sh"的一个函数.  2 # 去掉传递进来参数开头的0.  3  4 strip\_leading\_zero () # 去掉从参数中传递进来的,  5 { #+ 可能存在的开头的0(也可能有多个0).  6  return=${1#0} # "1"表示的是"$1" -- 传递进来的参数.  7 } # "0"就是我们想从"$1"中删除的子串 -- 去掉零. |

下边是Manfred Schwarb给出的一个更加详细的例子:

|  |
| --- |
| 1 strip\_leading\_zero2 () # 去掉开头可能存在的0(也可能有多个0), 因为如果不取掉的话,  2 { # Bash就会把这个值当作8进制的值来解释.  3  shopt -s extglob # 打开扩展的通配(globbing).  4  local val=${1##+(0)} # 使用局部变量, 匹配最长连续的一个或多个0.  5  shopt -u extglob # 关闭扩展的通配(globbing).  6  \_strip\_leading\_zero2=${val:-0}  7  # 如果输入为0, 那么返回0来代替"".  8 } |

另一个用法示例:

|  |
| --- |
| 1 echo `basename $PWD` # 当前工作目录的basename(就是去掉目录名).  2 echo "${PWD##\*/}" # 当前工作目录的basename(就是去掉目录名).  3 echo  4 echo `basename $0` # 脚本名字.  5 echo $0 # 脚本名字.  6 echo "${0##\*/}" # 脚本名字.  7 echo  8 filename=test.data  9 echo "${filename##\*.}" # data  10  # 文件扩展名. |

${var%Pattern}, ${var%%Pattern}

从变量$var的*结尾*删除最短或最长匹配$Pattern的子串. (译者注: 这是一个很常见的用法, 请读者牢记, 一个"%"表示匹配最短, "%%"表示匹配最长.)

Bash的[版本2](mk:@MSITStore:C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\Rar$DI00.296\高级bash脚本编程指南.CHM::/bashver2.html#BASH2REF)添加了一些额外选项.