链滴

shell变量

作者：wangsch

原文链接：https://ld246.com/article/1464770137409

来源网站：链滴

<h1>一、基本操作</h1>
<h2>变量赋值</h2>
直接用等号赋值，中间不要加任何空格。如：name=zhangsan

<h2>获取变量的值</h2>
<h3>$VAR_NAME</h3>
如：echo $name
如果需要在变量基础上累加内容，尽量用双引号将变量括起以免和后面的字符连起来造成混淆。
如：PATH="$PATH":/home/bin

加双引号会有几点不同
* 作为整体“保护”起来，不会引起混淆
* 不会忽略换行符
<pre class="brush: bash">wangsch@wangsch-pc:/tmp/test$ touch a b c

wangsch@wangsch-pc:/tmp/test$ files=`ls -l`
wangsch@wangsch-pc:/tmp/test$ echo $files
总用量 0 -rw-rw-r-- 1 wangsch wangsch 0 6月 1 16:32 a -rw-rw-r-- 1 wangsch wangsch 0 6月 1 16:32 b -rw-rw-r-- 1 wangsch wangsch 0 6月 1 16:32 c

wangsch@wangsch-pc:/tmp/test echo "files"
总用量 0
-rw-rw-r-- 1 wangsch wangsch 0 6月 1 16:32 a
-rw-rw-r-- 1 wangsch wangsch 0 6月 1 16:32 b
-rw-rw-r-- 1 wangsch wangsch 0 6月 1 16:32 c

wangsch@wangsch-pc:/tmp/test echo {files}
总用量 0 -rw-rw-r-- 1 wangsch wangsch 0 6月 1 16:32 a -rw-rw-r-- 1 wangsch wangsch 0 6月 1 16:32 b -rw-rw-r-- 1 wangsch wangsch 0 6月 1 16:32 c

wangsch@wangsch-pc:/tmp/test echo "{files}"
总用量 0
-rw-rw-r-- 1 wangsch wangsch 0 6月 1 16:32 a
-rw-rw-r-- 1 wangsch wangsch 0 6月 1 16:32 b
-rw-rw-r-- 1 wangsch wangsch 0 6月 1 16:32 c</pre>

不加双引号，会显示成一行（上面可能看到不是一行，这只是被博客的编辑器强硬的换行了！）加了双引号，就按原样显示了
* 如果变量出现在判断条件中，尽量用双引号括起来（尤其变量是个字符串的时候，可能包含空）否则可能出错

<h3>${VAR_NAME}</h3>
上面$VAR_NAME的方式可以替换为${VAR_NAME}，需要注意如果不想忽略换行，同样需要用引号括起来。实际上，${VAR_NAME}的写法不是为了简单的获取变量值的，还能做字符串替换、删等操作，后面会有说明。

算数运算
let、expr、((i=$j+$k))、bc

`pwd`等价于$(pwd)

source 等价于 .，在当前（而不是另启动一个子shell）shell执行脚本或命令。
bash命令将开启子shell，echo $SHLVL查看当前所属的子shell层次，从1开始，依次递增

直接使用env和export，显示环境变量
$RANDOM是随机数，介于 0~32767 之间

直接使用declare和set，显示所有变量（含环境变量与自定义变量）

自定义变量和环境变量

export输出变量到子shell中（相反的，子shell不能导出到父shell）

数组元素要用${}引用：
wangsch@wangsch-pc:~$ arr[0]=hello
wangsch@wangsch-pc:~$ echo $arr[0]
hello[0]
wangsch@wangsch-pc:~$ echo ${arr[0]}
hello

使用${}实现
1. 变量内容的删除、取代与替换
<table width="95%" border="1" cellspacing="0" cellpadding="3" bgcolor="lightyellow">
<tbody>
<tr align="center" bgcolor="#182448">
<td width="180">变量配置方式</td>
<td>说明</td>
</tr>
<tr>
<td style="font-family: '细明体';">${变量<pan style="text-decoration: underline;">#关键词}
${变量##关键词}</san></td>
<td style="font-size: 10pt;">若变量内容从头开始的数据符合『关键词』，则将符合的最短数据删
 若变量内容从头开始的数据符合『关键词』，则将符合的最长数据删除</td>
</tr>
<tr>
<td style="font-family: '细明体';">${变量<pan style="text-decoration: underline;">%关键词}
${变量%%关键词}</pan></td>
<td style="font-size: 10pt;">若变量内容从尾向前的数据符合『关键词』，则将符合的最短数据删
 若变量内容从尾向前的数据符合『关键词』，则将符合的最长数据删除</td>
</tr>
<tr>
<td style="font-family: '细明体';">${变量<pan style="text-decoration: underline;">/旧字符串/新字符串}
<span stye="text-decoration: underline;">${变量//旧字符串/新字符串}</td>
<td style="font-size: 10pt;">若变量内容符合『旧字符串』则『第一个旧字符串会被新字符串取代

 若变量内容符合『旧字符串』则『全部的旧字符串会被新字符串取代』</td>
</tr>
</tbody>
</table>

2. 变量的测试与内容替换

<h1>参考</h1>
http://vbird.dic.ksu.edu.tw/linux_basic/0320bash_2.php
