

## Linux Shell 快速入门

### 1. 变量的定义:

<1> Shell 变量没有类型限制, 使用=为变量赋值, 使用\$引用变量.

```
var=abc  
echo $avr
```

### <2> 特殊变量:

```
$# 命令行参数个数  
$? 保存前一个命令的返回码 (0 表示成功)  
$0 保存程序名  
$@ 保存所有命令行参数  
$* 保存所有命令行参数  
$1, $2, $3, ... 保存程序的命令行参数 (特殊的位置变量表示命令行参数)
```

### 2. 反斜杠 (\), 单引号 ('), 双引号 ("), 反引号 (`) 的区别:

<1> 反斜杠用于转意,

<2> 单引号不能作变量的内插

<3> 双引号可以作变量的内插

<4> 反引号可以做变量内查, 把整个反引号的内容当作一条命令执行, 返回结果

<5> 例子:

```
var=abc  
var1 = " \ $var " → $var  
var2 = ' $var ' → $var  
var3 = " $var " → abc  
var4 = `ls -l $var`
```

### 3. If 语句的格式:

```
<1> if [ expression1 ]      // 要注意: 条件大括号的两边一定要有空格  
then  
    command1  
elif [ expression2 ]  
    command2  
else  
    command3  
fi
```

或者

if [ expression ]; then // 要特别注意, then 前面不要加分号, 表示一条语句结束.

command1

else

command2

fi

<2> 例子:

var=abc

if [ "\$var" = "abc" ];then

echo "true"

else

echo "false"

fi

#### 4. case 语句格式

<1> case string1 in

str1) command1;;

str2) command2;;

str3) command3;;

esac

将条件 string1 和 str1,str2,str3 比较, 如果相等, 则执行对应的 command 命令

注意: 每个分支命令都是用;;两个分号结尾的.

<2> 例子:

case 1 in

1) echo "this is 1";;

2) echo "this is 2";;

3) echo "this is 3";;

esac

#### 5. for 语句格式

<1> for var in list

do

command

done

<2> 例子

```
for var in "a" "b" "c" "d" // 使用空格分隔 list
do
    echo $var
done
```

6. while 语句

<1> while [ Expression ] ; do  
 command  
done

<2> 例子:

```
#!/bin/sh
var=19
while [ $var -ge 9 ]; do
    echo $var
    var=`expr $var - 1` // 注意减号两面的空格，这是 sh 下获取运算结果的方式
done
```

7. 条件判断:

<1> string1 = string      若两个字符串相等，则为真  
    string1 != string2    若两个字符串不相等，则为真

int1 -eq int2	若 int1 等于 int2，则为真
int1 -ne int2	若 int1 不等于 int2，则为真
int1 -lt int2	若 int1 小于 int2，则为真
int1 -le int2	若 int1 小于等于 int2，则为真
int1 -gt int2	若 int1 大于 int2，则为真
int1 -ge int2	若 int1 大于等于 int2，则为真