



Shell的语法介绍

计算机学院 盛剑会



Shell的语法介绍

使用变量：在Shell程序中**所有变量中保存的值**都是**字符串**。

变量类型	说 明
环境变量	<p>与Shell的执行环境相关的一些变量。如PATH，HOME等，用户可重新定义。</p> <p>一、环境变量的创建：export， export ABCD=2</p> <p>二、环境变量的查看</p> <p>1 使用echo命令查看单个环境变量。如： echo \$PATH</p> <p>2 使用printenv查看所有环境变量。</p> <p>3 使用set查看所有本地定义的环境变量。</p>
预定义变量	用户不能修改，只能引用的变量。由“\$”+其他符号组成。
位置变量	在执行脚本时，传入到脚本中对应脚本位置的变量
标准变量	也是环境变量，在bash环境建立时生成。可使用printenv命令查看
自定义变量	变量名=变量值（变量名前不需加\$，等号两边不能有空格）

常用的预定义变量

预定义变量	说 明
\$#	传递到脚本或函数的参数数量
\$*	传递到脚本或函数的全部参数
\$?	前一个命令执行情况，返回0表示成功，其他值表示失败
\$\$	当前进程的ID（PID）
\$_	后台运行的最后一个进程的进程ID（PID）
\$0	当前脚本的名称。

预定义变量的使用

预定义脚本variable.sh

```
1 #!/bin/bash
2 echo "The letters you like is: $1 $2 $3"    #输出位置变量的参数
3 echo "本脚本名：" $0
4 echo "本脚本中参数的个数：" $#
5 echo "本脚本的所有参数：" $*
6 echo "上一条命令的返回值：" $?    #0, 成功；其他值失败
7 cd /hom          #切换至目录/home，此处故意输错，使命令不生效
8 echo $?          #上一条命令 (cd /hom) 执行失败，返回非0
9 cd /home         #此处输入正确
10 echo "上一条命令的返回值" $?
11 echo "当前进程的pid进程号：" $$
```

运行结果

```
[root@localhost ~]# sh variable.sh SJH DZY LXN
The letters you like is: SJH DZY LXN
本脚本名： variable.sh
本脚本中参数的个数： 3
本脚本的所有参数： SJH DZY LXN
上一条命令的返回值： 0
variable.sh: 第 7 行:cd: /hom: 没有那个文件或目录
上一条命令'cd /hom'的返回值： 1
上一条命令的'cd /home'的返回值： 0
当前进程的pid进程号： 13269
```

Shell的语法介绍

输入/输出内部命令

1 . 输出命令 : **echo**

2 . 输入命令 : **read**

举例3 : 编写一个Shell程序contact.sh , 用于接收用户输入的姓名和电话号码 , 并显示在终端上 , 和保存到contact.txt文件中。

contact.sh

```
1 # ! /bin/bash
2 # 参数-p把键盘输入的数据存入变NAME,PHONE
3 read -p "Enter your name phone:" NAME PHONE
4 echo $NAME,$PHONE
5 echo $NAME,$PHONE>contact.txt
```

Shell的语法介绍

条件测试

1. 测试命令执行结果

执行任何Linux命令都可能存在两种状态：正确或错误。若前一条命令正确执行，则返回状态值为0，返回值为任何非0值都表示执行命令出错。

使用预定义变量 “\$?” 可返回命令执行后的状态，在程序中可根据预定义变量 “\$?” 的值来判断前一条命令的执行是否正确。

Shell的语法介绍

条件测试

2. 测试文件状态：[测试表达式]

注意：中括号 “[]” 与测试表达式之间必须有一个空格来分隔。

测试文件状态可使用以下参数：

- e 文件名：若文件存在，返回真。
 - r 文件名：若文件可读，返回真。
 - w 文件名：若文件可写，返回真。
 - x 文件名：若文件可执行，返回真。
-

Shell的语法介绍

条件测试

2. 测试文件状态：[测试表达式]

注意：中括号 “[]” 与测试表达式之间必须有一个空格来分隔。

测试文件状态可使用以下参数：

- s 文件名：若文件类型为套接字文件，返回真。
- d 文件名：若文件类型为目录文件，返回真。
- f 文件名：若文件类型为普通文件，返回真。
- c 文件名：若文件类型为字符设备文件，返回真。
- b 文件名：若文件类型为块设备文件，返回真。

举例4：isdir

举例5：compare1

Shell的语法介绍

条件测试

3. 测试数据值

- eq:判断是否相等
- ne:判断是否不相等
- gt:判断是否大于
- ge:判断是否大于等于
- lt:判断是否小于
- le:判断是否小于等于

举例6 : compare2

Shell的语法介绍

条件测试

4 . 测试字符串：

= != -z -n

举例7：compare3

5 . 逻辑操作符：

! -a -o

Shell的语法介绍

流程控制语句

1. if条件语句

```
if  条件测试命令串
then
    条件为真时执行的命令
else
    条件为假时执行的命令
fi
```

举例8 : ifyuju

Shell的语法介绍

流程控制语句

2 . case条件语句

```
case 条件测试命令串 in
表达式1)
    分支1执行的命令
    ;;
表达式2)
    分支2执行的命令
    ;;
... ..
*)
    其他命令
esac
```

举例9 : caseyuju

Shell的语法介绍

流程控制语句

3 . for循环语句

```
for 变量名  
    [in 数值列表]  
do  
    循环执行的命令串  
done
```

举例10 : fortest

举例11 : loop1

Shell的语法介绍

流程控制语句

4 . while和until循环语句

举例12 : whilettest

举例13 : loop2

举例14 : untiltest

举例15 : loop3

```
while 循环条件命令串
do
    循环命令串
done
```

```
until 循环条件命令串
do
    循环命令串
done
```

Shell的语法介绍

函数

```
函数名()  
{  
    命令串  
}
```

举例16 : hanshu

举例17 : func1

Shell的语法介绍

处理参数

1. 位置参数

当调用Shell程序的命令行时，除了输入Shell程序名之外，还可以在后面跟上多个参数，这些参数与在命令行中的位置有关，因此称为**位置参数**。各位置参数之间用空格分隔，用**\$1**表示第1个参数，**\$2**表示第2个参数，依次类推。而**\$0**是一个特殊的变量，其内容是**当前Shell程序的文件名**。

举例18：parameter1

Shell的语法介绍

处理参数

2. 函数参数

向函数传递参数与向一般脚本传递参数一样，应使用预定义变量**\$1~\$9**来传递。

函数取得所传参数后，将原始参数传回Shell脚本。

举例19：sum1
