

shell直接提供无需定义的变量

\$#: 传给 shell 脚本参数的数量

\$*: 传给 shell 脚本参数的内容

\$1、\$2、\$3、...、\$9: 运行脚本时传递给其的参数，用空格隔开

\$?: 命令执行后返回的状态

"\$?"用于检查上一个命令执行是否正确(在 Linux 中，命令退出状态为 0 表示该命令正确执行，任何非 0 值表示命令出错)。

\$0: 当前执行的进程名

\$\$: 当前进程的进程号

"\$\$"变量最常见的用途是用作临时文件的名字以保证临时文件不会重复

编写文件

```
1 vim shellvar.sh
```

```
1 #!/bin/bash
2 echo "参数的个数=$#"
3 echo "参数的内容=$*"
4 echo "第一个参数:$1"
5 echo "第二个参数:$2"
6 echo "第三个参数:$3"
7 readonly data=10
8 data=250
9 echo "data=250的结果:$?"
10 echo "进程名:$0"
11 echo "进程号:$$"
```

结果

```
1 [root@iZuf6gy6qwwk16k8vp21vrZ script]# ./shellvar.sh a b c
2 参数的个数=3
3 参数的内容=a b c
4 第一个参数:a
5 第二个参数:b
6 第三个参数:c
7 ./shellvar.sh: line 8: data: readonly variable
8 data=250的结果:1
9 进程名:./shellvar.sh
10 进程号:14326
```

脚本标量的特殊用法

1.2.2.4 脚本变量的特殊用法: "" `` '\0\0

"" (双引号): 包含的变量会被解释

" (单引号): 包含的变量会当做字符串解释

`` (数字键 1 左面的反引号): 反引号中的内容作为系统命令, 并执行其内容, 可以替换输出为一个变量

\$ echo "today is `date` "

today is 2012 年 07 月 29 日星期日 12:55:21 CST

\ 转义字符:

同 c 语言 \n \t \r \a 等 echo 命令需加-e 转义

(命令序列):

由于 shell 来完成, 不影响当前 shell 中的变量

{ 命令序列 }:

在当前 shell 中执行, 会影响当前变量

https://blog.csdn.net/weixin_4

```
1 #!/bin/sh
2 echo "today is `date` "
3 ls
4 echo "##\n##"
5 echo -e "##\n##"
6 data=10
7 (
8     data=100
9     echo "()里面data = $data"
10 )
11 echo "当前data=$data"
12
13 b=B
14 {
15     b=A
16     echo "{}里面b=$b"
17 }
18 echo "current b is $b"
```

结果:

```
1 [root@iZuf6gy6qwk16k8vp21vrZ script]# chmod +x special_char.sh
2 [root@iZuf6gy6qwk16k8vp21vrZ script]# ./special_char.sh
3 today is Tue Oct 19 03:58:14 PM CST 2021
4 shellvar.sh special_char.sh text01.sh text02.sh
5 ##\n##
6 ##
7 ##
8 ()里面data = 100
9 当前data=10
10 {}里面b=A
11 current b is A
```

