



链滴

shell 脚本从零实战——终端显示

作者: [Smiteli](#)

原文链接: <https://ld246.com/article/1563324041405>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

1.用 vim 编写第一个脚本:

```
vim helloworld.sh
```

脚本内容如下:

```
#!/bin/bash
```

```
# 第一行被称作 shebang , 脚本在运行时, 内核会读取首行, 识别出 /bin/bash
```

```
# 然后执行脚本
```

```
echo "hello world!"
```

2.执行脚本

2.1 `bash helloworld.sh`

2.2 授予脚本执行权限, 然后执行:

```
chmod 755 helloworld.sh
```

```
./helloworld.sh 这里的 . 表示当前目录
```

```
或者用全路径来执行: /home/tom/helloworld.sh
```

3.登录linux时, 会启动一个 shell 窗口给用户进行交互式输入。在启动一个shell的过程中, 会执行当用户目录中的`.bashrc`文件, 或者是`.bash_profile`文件。所以, 对于一些需要提前设置的命令可以放在`.ashrc`文件, 或者是`.bash_profile`文件中, 如修改环境变量 PATH: `echo "PATH=$PATH:~/bin" >> ~/.bashrc`

```
~/.bash_history 保存了用户运行过的命令
```

4.一次运行多次命令

```
mkdir test;cd test
```

上面的命令等同于下面:

```
mkdir test
```

```
cd test
```

5.再次探讨`echo`命令:

5.1 不用单/双引号:

```
[root@localhost ~]# echo welcome to bash
```

```
welcome to bash
```

```
[root@localhost ~]# echo "welcome to bash"
```

```
welcome to bash
```

```
[root@localhost ~]# echo 'welcome to bash'
```

```
welcome to bash
```

上面的命令执行都没有问题。下面我们再执行一条:

```
[root@localhost ~]# echo welcome to bash!
```

```
welcome to bash!
```

```
[root@localhost ~]# echo 'welcome to bash!'
welcome to bash!
[root@localhost ~]# echo "welcome to bash!"
-bash: !": event not found
```

可以看到，字符串中用了!时会报错。这是因为像!,;等这种符合在shell中都是特殊字符，有着特殊含义在使用 echo 命令时，如果使用了双引号，那么就要对特殊字符进行转义，如：

```
[root@localhost ~]# echo "welcome to bash\!"
welcome to bash\!
```

在 shell 中，

1. ! 号具有执行某条历史命令的作用，如：

! 59 表示执行第 59 条命令

!! 表示执行上一条命令

2. ; 号表示两条命令的分隔，如上面第4点提到的

6.echo 命令的一些说明

6.1 默认情况下，echo 会在输出文本的尾部追加一个换行符，可以使用 -n 来禁止这种行为：

```
[root@localhost ~]# echo test
test
[root@localhost ~]# echo -n test
test[root@localhost ~]#
```

6.2当双引号中的字符串需要转义时，如何使用？使用 -e 选项：

```
[root@localhost ~]# echo -e "1\t2\t3"
1 2 3
[root@localhost ~]#
```

7.另一个终端打印命令：printf

编写脚本如下：vim printf.sh

```
#!/bin/bash

printf "%-5s %-10s %-4s\n" No Name Mark
printf "%-5s %-10s %-4.2f\n" 1 Apple 90.555
printf "%-5s %-10s %-4.2f\n" 2 Boy 84.33
printf "%-5s %-10s %-4.2f\n" 3 Cat 79.66
```

执行脚本：

```
[root@localhost ~]# vi printf.sh
[root@localhost ~]# bash printf.sh
No Name Mark
1 Apple 90.56
```

```
2 Boy 84.33
3 Cat 79.66
```

脚本说明：

1. %s,%c,%d,%f都是格式替换符，定义了该如何打印后续的参数。
2. %-5s 指明了一个格式为左对齐且字符串宽度为5的字符替换（- 表示左对齐）。如果不用 - 就表示对齐。宽度指定了某个字符串的字符数量。脚本中，对Name而言，其宽度为10，超过10个则取前1个，不足10个，空格补足
3. 对Mark字段，我们设置为 %-4.2f，其中 .2 指定保留两位小数。
4. 每一行的字符串后面都跟了一个换行符（\n）
5. 如果你学过c语言，你肯定发现了 printf命令 的使用和 c 语言的 printf 函数的使用是一样的