

## 3.2 shell 功能性语句\_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 3.2 shell 功能性语句涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

### 一. 功能性语句

- 输入功能——read

格式: read 变量1 变量2 变量3 。  
功能: 从键盘标准读入一行，并赋值给后面的变量

示例:  
read var1 var2

特点:  
shell编程中若是利用read函数读取参数的时候，若是输入>大于当前参数，则当前参数前面依次对齐，最后一个把所有输入之后的参  
若是输入=当前参数，正常输入  
若是输入<当前参数，取输入的个数与当前参数对齐，不足者补空格。

示例代码:

test-01.sh

```
#!/bin/bash
#
#

echo -n "Input var : "
read VAR1 VAR2

echo VAR1 : $VAR1
echo VAR2 : $VAR2
```

- 算数计算——expr

格式1:  
expr 第一个操作数 运算符 第二个操作数

格式2:  
((C语言语句))

示例:  
var=`expr 1 + 3`  
echo "var = \$var"

((var=1 + 3))  
echo "var = \$var"

代码用法:

expr.sh

```
#!/bin/bash

echo -n "Input two num : "
read DAT1 DAT2
```

```

RES=`expr $DAT1 + $DAT2`
echo "$DAT1 + $DAT2 = $RES"

RES=`expr $DAT1 - $DAT2`
echo "$DAT1 - $DAT2 = $RES"

RES=`expr $DAT1 \* $DAT2`
echo "$DAT1 * $DAT2 = $RES"

RES=`expr $DAT1 / $DAT2`
echo "$DAT1 / $DAT2 = $RES"

```

运行结果:

```

Input two num : 12 4
12 + 4 = 16
12 - 4 = 8
12 * 4 = 48
12 / 4 = 3

```

- 测试——test

- 注：test 测试的时候，若是用到 =, = 两边要有空格

- 字符串

- = 测试两个字符串是否相等
- != 测试两个字符串是否不相等
- -z 测试两个字符串长度是否为0
- -n 测试两个字符串是否不为0

示例：

```
test 123 = 321
echo $?
```

```
test -z 456
echo $?
```

```
STR=123
test $STR = 123
echo $?
```

- 示例代码：

- # -n 代表输出不换行
- echo -n "Input str1 str2 : "
- read STR1 STR2
- #test测试命令字符串的时候，建议被子字符串用""引用起来
- test "\$STR1" = "\$STR2"
- echo "|\$STR1| = |\$STR2|" : \$?
- test "STR1" != "STR2"
- test -n "\$STR1"
- echo "|\$STR1| len is no zeor : \$?"
- test -z "\$STR2"
- echo "|\$STR2| len is zero : \$?"

- 整数

- -eq 等于
- -ne 不等于
- -ge 大于等于
- -le 小于等于
- -gt 大于
- -lt 小于

- 示例代码：

- echo -n "please input two int data : "
- read VAR1 VAR2
- test \$VAR1 -eq \$VAR2
- echo "\$VAR1 -eq \$VAR2 : \$?"
- test \$VAR1 -ne \$VAR2

```
echo "$VAR1 -ne $VAR2 : $?"

test $VAR1 -gt $VAR2
echo "$VAR1 -gt $VAR2 : $?"

test $VAR1 -ge $VAR2
echo "$VAR1 -ge $VAR2 : $?"

test $VAR1 -lt $VAR2
echo "$VAR1 -lt $VAR2 : $?"

test $VAR1 -le $VAR2
echo "$VAR1 -le $VAR2 : $?"
```

- 且与或

- -a 且的关系连接多个命令
- -o 或的关系连接多个命令

- 示例代码：

- ```
echo -n "Input on number : "
read x

#(( x > 8 && x < 100))
test $x -gt 8 -a $x -lt 100
echo $?
```

- 文件

```
-d : 测试是否是一个目录文件
-f : 测试是否是一个普通文件
-w : 测试是否可写
-r : 测试是否可读
-x : 测试是否可执行
echo -n "please input a filename : "
read filename

#test -f $filename
[ -f $filename ]
echo "$filename : $?"

#test -d /home/linux
[ -d /home/linux ]
echo "/home/linux : $?"
```

## 二. 课后练习

### 练习 1:

写一个 shell 脚本，要求用户从键盘输入年，月，日，输出的时候格式为 XX-XX-XX

例如：

输入 2022 9 1 , 输出 2022-9-1

### 练习 2:

写一个 shell 脚本，要求从键盘输入 2 个数，输出这两个数的 +,-,\*,/,%

### 练习 3:

写一个 shell 脚本，要求用户从键盘输入一个数。若是该数在 [0~60) 之间输出 0，  
否则输出 1.

- 划线
- 写笔记

学习要认真，笔记应当先



公开笔记 0/1000 提交



Sunny\_SunshineX

删除 编辑

全文完

本文由 简悦 SimpRead 优化，用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta，点击查看详细说明

