

Shell编程

bash c shell k shell

默认情况下Linux使用的是bash

变量

存储数据的内存空间

注意:变量的命名

变量名 (可以自定义)= "Value"

1abc="11" 不对

_abc="11" 可以

\$abc="11" 不对

abc="11" 可以

abc_efg="11" 可以

readonly 变量名 (设置变量只读)

unset 变量名 (删除变量对应内存上面的值)

数组(集合)-- 数组也是一个变量

存储多个变量的集合

注意:变量的命名

数组变量名=(value1 value2 value3)

运算符(+ - * / %)Linux Shell没有给我提供算术运算，

如果要使用算术运算可以采用下面两种方式：(evaluate expressions)

1. 结果变量名=`expr \$变量名1 + \$变量名2` (\$? : 获得上一个命令的结果)
2. 结果变量名=\$((\$变量名1 + \$变量名2))

基本算术运算符

a=1

b=2

#c=`expr \$a + \$b` 第一种计算

```
c=$(( $a + $b ))
```

参数传递

1. 第一种：给脚本传值，在执行脚本的时候传递参数

传值语法结构

```
./demo.sh 1 2 3 4 5 5
```

2. 第二种：给函数传值，在调用函数的时候传递参数

传值的语法结构

```
sum 1 2 3 4
```

获取参数：

1. `$*` 将参数作为一个字符串进行传递
2. `$@` 将参数作为一个数组集合进行传递
3. `$数字` (大于等于0的整数) 注意：`$0`是获取执行的脚本文件名称

流程控制：

1. if-elif-else-fi (判断语句结构 elif: else if) **case in 选学**

判断学生成绩，给学生成绩定等级

```
score=90
```

-ge (greater equal) 大于等于

-le (less equal) 小于等于

-eq (equal) 等于

-gt (greater than) 大于

-lt (less than) 小于

-ne (not equal) 不等于

```
if [ $score -ge 90 -a $score -le 100 ]
```

```
then
```

```
    echo "该学生成绩优秀"
elif [ $score -ge 80 -a $score -lt 90 ]
then
    echo "该学生成绩良好"
else
    echo "该学生成绩有待提高"
fi
```

循环语句

1. for循环

```
for 变量名 in 数组集合;
do
    echo $变量名
done
```

- 遍历打印/training/demo01下面的所有文件和目录的名称
- 遍历一个数组(zhangsan lis wangwu)

1. while do循环

```
while()
do
    echo "hello"
done
```

- 计算1~100的和

break: 终止整个循环体，结束所有循环执行，执行循环外面的脚本语句

continue : 终止当前循环，进入下一次循环,continue语句下面的所有循环体内部的语句都不会执行

```

#保存计算的结果
sum=0
while(($i<=100))
do
    if [ $i -lt 50 ]
    then
        let "i++"
        continue
    else
        sum=`expr $sum + $i`
        let "i++"
    fi
    echo "hahaha"$i
#   sum=`expr $sum + $i`
#   let "i++"
done
echo "1~100的和为: "$sum

```

函数

函数的定义：

函数名的命名规则：

如果说是由多个单词组成，第一个单词全部小写，后面所有单词第一个字母大写

function 函数名()

{

函数体

}

参数传递

函数名 参数列表 (sum 1 2 3 4 5。。。)