

函数的定义、调用和返回值

shell中的函数类似于C/C++中的函数，但return的返回值与C/C++不同，返回的是exit code，取值为0-255，0表示正常结束。

如果想获取函数的输出结果，可以通过echo输出到stdout中，然后通过\$(function_name)来获取stdout中的结果。

函数的return值可以通过\$?来获取。

命令格式：

```
1 [function] func_name() { # function关键字可以省略
2     语句1
3     语句2
4     ...
5 }
6 f(){
7     echo "打开试卷"
8     echo "填写姓名"
9     echo "开始答题"
10
11     return 123
12 }
13
14 f
15
16 echo $?
17 echo $(f)
18 hioier@yunpc:~$ ./test.sh
19 打开试卷
20 填写姓名
21 开始答题
22 123
23 打开试卷 填写姓名 开始答题
```

函数参数

函数的输入参数

在函数内，\$1 表示第一个输入参数，\$2 表示第二个输入参数，依此类推。

注意：函数内的\$0 仍然是文件名，而不是函数名。

```
1 add(){
2     a=$1
3     b=$2
4     echo `expr $a + $b`
5 }
6
7 echo $(add 1 2)
```

递归求和

```
1 sum(){
2     if [ $1 -le 0 ]
3     then
4         echo 0
5         return 0
6     fi
7
8     ret=$(sum `expr $1 - 1`)
9     echo `expr $ret + $1`
10 }
11
12 echo $(sum 100)
```

exit命令

exit命令用来退出当前shell进程，并返回一个退出状态；使用\$?可以接收这个退出状态。

exit命令可以接受一个整数值作为参数，代表退出状态。如果不指定，默认状态值是 0。

exit退出状态只能是一个介于 0~255 之间的整数，其中只有 0 表示成功，其它值都表示失败。

```
1 if [ $# -eq 1 ]
2 then
3     echo "yes"
4     exit 0
5 else
6     echo "no"
7     exit 1
8 fi
9 hioier@yunpc:~$ ./test.sh 0
10 yes
11 hioier@yunpc:~$ echo $?
12 0
13 hioier@yunpc:~$ ./test.sh
14 no
15 hioier@yunpc:~$ echo $?
16 1
17 hioier@yunpc:~$ cat test.sh
```

文件输入输出重定向

```
1 add(){
2     a=$1
3     b=$2
4     echo `expr $a + $b`
5 }
6
7 read -p "a:" a
8 read -p "b:" b
9
10 echo $(add $a $b)
11 hioier@yunpc:~$ ./test.sh < data.in > data.out
12 hioier@yunpc:~$ cat data.out
13 7
```