

# Shell 函数

主讲人：杨老师

## Shell 函数

linux shell 可以用户定义函数，然后在shell脚本中可以随便调用。

shell中函数的定义格式如下：

```
[ function ] funname [()]  
{  
    action;  
  
    [return int;]  
}
```

说明：

- 1、可以带function fun() 定义，也可以直接fun() 定义,不带任何参数。
- 2、参数返回，可以显示加：return 返回，如果不加，将以最后一条命令运行结果，作为返回值。

return后跟数值n(0-255 )

## Shell 函数---实例1

下面的例子定义了一个函数并进行调用：

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

```
# author:dajiangtai
```

```
# url:http://www.dajiangtai.com/
```

```
myFun(){
```

```
    echo "这是我的第一个 shell 函数!"
```

```
}
```

```
echo "-----函数开始执行-----"
```

```
myFun
```

```
echo "-----函数执行完毕-----"
```

## Shell 函数---实例2

下面定义一个带有return语句的函数：

```
funWithReturn(){  
    echo "对两个整数进行相加运算..."  
    echo "输入第一个整数: "  
    read m  
    echo "输入第二个整数: "  
    read n  
    echo "两个整数分别为 $m 和 $n !"  
    return $(( $m + $n ))  
}  
funWithReturn  
echo "输入的参数之和为 $? !"
```

函数返回值在调用该函数后通过 \$? 来获得。

注意：所有函数在使用前必须定义。这意味着必须将函数放在脚本开始部分，直至shell解释器首次发现它时，才可以使用。调用函数仅使用其函数名即可。

## Shell 函数参数

在Shell中，调用函数时可以向其传递参数。在函数体内部，通过 `$n` 的形式来获取参数的值，例如，  
`$1`表示第一个参数，`$2`表示第二个参数...

带参数的函数示例：

```
functionWithParams(){  
    echo "第1个参数为 $1 !"  
    echo "第2个参数为 $2 !"  
    echo "第3个参数为 $3 !"  
    echo "第9个参数为 $9 !"  
    echo "第10个参数为 ${10} !"  
    echo "参数总数有 $# 个!"  
    echo "按字符串输出所有参数 $* !"  
}
```

```
functionWithParams 1 2 3 4 5 6 7 8 9 10
```

注意，`$10` 不能获取第十个参数，获取第十个参数需要`${10}`。当`n >= 10`时，需要使用`${n}`来获取参数。

## Shell 函数参数

另外，还有几个特殊字符用来处理参数：

参数处理	说明
<code>\$#</code>	传递到脚本的参数个数
<code>\$*</code>	以一个单字符串显示所有向脚本传递的参数
<code>\$\$</code>	脚本运行的当前进程ID号
<code>\$_</code>	后台运行的最后一个进程的ID号
<code>\$@</code>	与 <code>\$*</code> 相同，但是使用时加引号，并在引号中返回每个参数。
<code>\$-</code>	显示Shell使用的当前选项，与set命令功能相同。
<code>\$?</code>	显示最后命令的退出状态。0表示没有错误，其他任何值表明有错误。



**THANKS**

**黄丽老师 : 3354223855**

**小夏老师 : 972628726**