

# 06.Shell函数应用

- 06.Shell函数应用
  - 1.定义函数
  - 2.调用函数
  - 3.函数返回
  - 4.函数案例

徐亮伟, 江湖人称标杆徐。多年互联网运维工作经验, 曾负责过大规模集群架构自动化运维管理工作。擅长Web集群架构与自动化运维, 曾负责国内某大型电商运维工作。

个人博客"[徐亮伟架构师之路](#)"累计受益数万人。

笔者Q:552408925、572891887

架构师群:471443208

## 函数的作用

- 1.命令合集, 完成特定功能的代码块
- 2.函数必须先定义才可以使用, 如果定义不调用则不被执行
- 3.使代码模块化, 避免重复代码, 加强可读性, 便于复用

## 1.定义函数

定义函数的方法如下

```
//方式一
函数名() {
    函数要实现的功能代码
}

//方式二
function 函数名 {
    函数要实现的功能代码
}
```

## 2.调用函数

调用函数非常简单, 只需要在脚本中写入函数的名称即可

//调用函数

function\_name 即可调用函数

//函数传参

**fun** \$1 传入脚本后第一个参数

**fun** \$\* 接收所有参数的传递

## 3.函数返回

```
#!/bin/bash
fun2() {
  read -p "enter num: " num
  return ${2*$num}
}

fun2
echo "fun2 return value: $?"
```

### 位置参数

```
#!/usr/bin/bash
if [ $# -ne 3 ];then
  echo "usage: `basename $0` par1 par2 par3"
  exit
fi

fun3() {
  echo "${($1 * $2 * $3)}"
}

result=`fun3 $1 $2 $3`
echo "result is: $result"
```

### 数组传参

```
#!/usr/bin/bash
num=(1 2 3)
num2=(2 3 4)

array() {
  local factorial=1
  for i in $*
```

```
do
    factorial=$((factorial * $i))
done
echo "$factorial"
}
```

  

```
#array ${num[@]}
array ${num[*]}      #数组所有元数值
array ${num2[*]}     #数组所有元数值
```

## 4.函数案例

---

- 1.所有脚本必须有选择菜单
- 2.所有的危险操作均需给出提示

### 1.编写系统初始化脚本

```
配置 YUM
安装 Nginx
q 键退出程序
```

### 2.编写虚拟机管理脚本

```
安装虚拟机
关闭虚拟机
打开虚拟机
克隆虚拟机（可以指定克隆的数量）
q键退出程序
```

### 3.编写系统管理工具箱

```
查看内存的使用情况
查看磁盘的使用情况
查看系统的负载
q 键退出程序
```