

3.7 Shell 中的函数_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 3.7 Shell 中的函数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。

我们常常把完成固定功能、且多次使用的一组命令（语句）封装在一个特定的结构中
这种结构，我们就叫做函数。

```
方法1:
function 函数名()
{
    命令1
    命令2
    ...
}
```

```
方法2:
函数名()
{
    命令1
    命令2
    ...
}
```

示例用法:

```
#!/bin/bash

fun_test1()
{
    cd /home/linux
    mkdir new_dir
}

function fun_test2()
{
    cd /home/linux
    mkdir new_dir
}
```

方法1: 通过\$?获得函数的返回值

函数名 参数1 参数2

\$? 获得return返回值

方法2: 通过echo来获得函数的结果

变量=`函数名 参数1 参数2`
注意：此种方法是函数中的内容是通过echo输出，把echo输出的内容保存在变量中。

示例代码 1:

```
#!/bin/bash

fun_test1()
{
    echo "fun_test1 is call"
    value = 20
    return $value
}
```

```

fun_test2()
{
    echo "fun_test2 is call"
    echo '$#' : $#          # 获得函数参数的个数
    echo '$1' : $1          # 获得第一个参数
    echo '$2' : $2          # 获得第二个参数
    echo '$0' : $0          # $0获得脚本名

    return `expr $1 + $2`
}

echo $0 start ...
echo "===== "
#调用无参的函数
fun_test1
ret=$?
echo $ret

echo "===== "

#调用有参函数
fun_test2 10 20
ret=$?
echo $ret

echo "===== "

```

示例代码 2:

```

#!/bin/bash

fun_test1()
{
    VAR="hello world"
    echo $VAR
}

echo $0 start ...

echo "===== "
echo "first VAR : $VAR"

#开始调用函数, 不传参
fun_test1

echo "===== "
#shell中变量默认为全局变量, 当shell函数调用后生效
echo "second VAR : $VAR"

str=`fun_test1`
echo $str

```

1. 设计一个加法函数 add(), 给函数传参 10,20, 计算 10 + 20 的和
2. 设计一个函数 output_string(), 返回 "hello word" 字符串

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

