[root@localhost Music]# vim sum.sh

[root@localhost Music]# sh sum.sh

Please input the number1: 1

Please input the number2: 2

0

1

3

#!/bin/bash

# 创建函数

function sum()

{

s=0

# 多个echo语句

echo "0"

s=$[ $1 + $2 ]

echo "1"

echo "$s"

}

read -p "Please input the number1: " n1

read -p "Please input the number2: " n2

result=$(sum $n1 $n2) # 向函数传递参数

echo "$result" # 返回值为多个echo输出的集合

[root@localhost Music]# vim fun\_list.sh

[root@localhost Music]# sh fun\_list.sh

The original array is: 1 2 3 4 5

The result is 15

#!/bin/bash

# adding values in an array

# 向函数传递数组参数

function addarray()

{

local sum=0

for value in $@

do

sum=$[ $sum + $value ]

done

echo $sum

}

myarray=(1 2 3 4 5) # 数组

echo "The original array is: ${myarray[\*]}"

arg1=${myarray[\*]} # 1.分解成单个的值,然后将这些值作为函数参数使用

result=$(addarray $arg1)

echo "The result is $result"

#!/bin/bash

# returning an array value

# 从函数返回数组

function arraydblr()

{

local origarray

local newarray

local elements

origarray=($(echo "$@")) # 重新包装为数组

newarray=($(echo "$@")) # 重新包装为数组

elements=$[ $# - 1 ] # 输入参数个数-1

for (( i = 0; i <= $elements; i++ )) # 类c for循环

{

newarray[$i]=$[ ${origarray[$i]} \* 2 ] # 修改数组元素的值

}

echo ${newarray[\*]} # 数组中所有元素(三个字显示数组的内容)

}

myarray=(1 2 3 4 5)

echo "The original array is: ${myarray[\*]}"

arg1=${myarray[\*]}

result=$(arraydblr $arg1) # 将函数的返回值包装为数组

echo "The new array is: ${result[\*]}"

[root@localhost Music]# vim fun\_return\_list.sh

[root@localhost Music]# sh fun\_return\_list.sh

The original array is: 1 2 3 4 5

The new array is: 2 4 6 8 10