1：函数定义

tip：Shell 函数必须先定义后使用。

格式如下：

```
[function]  name () {
    list of commands
    [ return value ]
}
```

tip:括号里面没有参数,function 可加可不加，为了提高速率，一般不写

函数返回值，可以显式增加return语句；如果不加，会将最后一条命令运行结果作为返回值

Shell 函数返回值只能是整数，一般用来表示函数执行成功与否，0表示成功，其他值表示失败。如果 return 其他数据，比如一个字符串，往往会得到错误提示：“numeric argument required”。

如果一定要让函数返回字符串，那么可以先定义一个变量，用来接收函数的计算结果，脚本在需要的时候访问这个变量来获得函数返回值。

2：函数使用

先来看一个例子：

```
1. #!/bin/bash
2. # Define your function here
3. Hello () {
4.    echo "Url is http://see.xidian.edu.cn/cpp/shell/"
5. }
6. # Invoke your function
7. Hello
```

运行结果：

```
$./test.sh
Hello World
$
```

调用函数只需要给出函数名，不需要加括号。

再来看一个带有return语句的函数：

```
#!/bin/bash
funWithReturn(){
    echo "The function is to get the sum of two numbers..."
    echo -n "Input first number: "
    read aNum
    echo -n "Input another number: "
    read anotherNum
    echo "The two numbers are $aNum and $anotherNum !"
    return $(($aNum+$anotherNum))
}
funWithReturn
# Capture value returnd by last command
ret=$?
echo "The sum of two numbers is $ret !"
```

运行结果：

```
The function is to get the sum of two numbers...
Input first number: 25
Input another number: 50
The two numbers are 25 and 50 !
The sum of two numbers is 75 !
```

函数返回值在调用该函数后通过 $? 来获得。

再来看一个函数嵌套的例子：

```
#!/bin/bash
# Calling one function from another
number_one () {
    echo "Url_1 is http://see.xidian.edu.cn/cpp/shell/"
    number_two
}
number_two () {
    echo "Url_2 is http://see.xidian.edu.cn/cpp/u/xitong/"
}
number_one
```

运行结果：

```
Url_1 is http://see.xidian.edu.cn/cpp/shell/
Url_2 is http://see.xidian.edu.cn/cpp/u/xitong/
```

像删除变量一样，删除函数也可以使用 unset 命令，不过要加上 .f 选项，如下所示：

```
$unset .f function_name
```

如果你希望直接从终端调用函数，可以将函数定义在主目录下的 .profile 文件，这样每次登录后，在命令提示符后面输入函数名字就可以立即调用。

给函数传参数

例如：

```
function_name 1 2 3 4 5 6 7 8 9 34 73
```

函数获取传入的参数

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：

```
#!/bin/bash
funWithParam(){
    echo "The value of the first parameter is $1 !"
    echo "The value of the second parameter is $2 !"
    echo "The value of the tenth parameter is $10 !"
    echo "The value of the tenth parameter is ${10} !"
    echo "The value of the eleventh parameter is ${11} !"
    echo "The amount of the parameters is $# !"  # 参数个数
    echo "The string of the parameters is $* !"  # 传递给函数的所有参数
}
funWithParam 1 2 3 4 5 6 7 8 9 34 73
```

运行脚本：

```
The value of the first parameter is 1 !
The value of the second parameter is 2 !
The value of the tenth parameter is 10 !
The value of the tenth parameter is 34 !
The value of the eleventh parameter is 73 !
The amount of the parameters is 12 !
The string of the parameters is 1 2 3 4 5 6 7 8 9 34 73 !"
```

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。

另外，还有几个特殊变量用来处理参数，前面已经提到：

特殊变量	说明
$#	传递给函数的参数个数。
$*	显示所有传递给函数的参数。
$@	与$*相同，但是略有区别。
$?	函数的返回值。