

## 作业 2: 脚本中使用函数

作者: 吴声炜

专业: 数学与应用数学 学号:3210102945

2022 年 6 月 28 日

让我们先来看一个简单的函数示例`example1.sh`

```
#!/bin/sh
foo(){
    echo "Function foo is executing"
}
echo "script starting"3
foo
echo "script ended"
exit 0
```

运行此脚本将会得到以下结果

```
script starting
Function foo is executing
script ending
```

从结果可以知道函数`foo`的作用就是在脚本运行到`foo`时, 转去执行`foo`中的命令, 等到`foo`执行完毕后再执行下一行命令.

显然的, 我们必须在使用函数前就已经声明了它, 才能够正常使用它. 事实上, 只要函数声明在使用之前即可, 但是为了美观, 我们通常会将所有函数声明写在第一个命令执行之前. 比如`example2.sh`, 它的结果与`example1.sh`是相同的.

```
#!/bin/sh
```

```

echo "script starting"
foo(){
    echo "Function foo is executing"
}
foo
echo "script ended"
exit 0

```

在调用函数时, 需要注意的是, 脚本的位置参数`$*`, `$@`, `$1`, `$2` 等都将被函数的参数替代. 当函数调用结束时, 它们又将恢复到以前的值. 比如`example3.sh`

```

#!/bin/sh
foo(){
    echo $1
}
echo $1
foo 2
echo $1
exit 0

```

运行`./example3.sh 1`, 其输出如下

```

1
2
1

```

此外, 还可以使用`local` 关键字在`shell` 函数中声明局部变量. 当局部变量与全局变量有相同名称时, 它将在函数内部覆盖该变量. 例如`example4.sh`

```

#!/bin/sh
sample_text="global variable"
foo(){
    local sample_text="local variable"
    echo "Function foo is executing"
    echo $sample_text
}

```

```
echo "script starting"  
echo $sample_text  
foo  
echo "script ended"  
echo $sample_text  
exit 0
```

会得到以下结果

```
script starting  
global variable  
Function foo is executing  
local variable  
script ended
```