shell脚本

第一行：#!/bin/bash

输出命令 echo

echo "output one cmd"

输入命令read

read cmd

调用变量

$cmd

数值计算

read -p "input twi int num:" first second

total=$(($first+$second))

test命令

用于检查某个条件是否成立，它可以进行数值、字符和文件三个方面的测试

&&和||命令

&&左边的命令（命令1）返回真(即返回0，成功被执行）后，&&右边的命令（命令2）才能够被执行

||则与&&相反。如果||左边的命令（命令1）未执行成功，那么就执行||右边的命令（命令2）

中括号[]判断符

echo "please input two string"

read -p "string1:" string1

read -p "string2:" string2

[ "$string1" == "$string2" ] && echo "string1" == "string2" || echo "string1" != "string2"

变量默认

$0~$n,表示shell脚本的参数

$0:shell脚本命令本身

$#:表示最后一个参数

$@:表示除了$0其他的所有参数

if语句

if [ "$string1" == "$string2" ]; then

echo "string1" == "string2"

fi

if [ "$string1" == "y" ] || [ "$string1" == "Y" ]; then

echo "input Y"

else

echo "input n"

fi

if [ "$string1" == "y" ] || [ "$string1" == "Y" ]; then

echo "input Y"

elif [ "$string1" == "n" ] || [ "$string1" == "N" ]; then

echo "input n"

else

echo "cmd err"

fi

case语句

case $1 in

"a")

echo "parm is:a"

;;

"b")

echo "parm is:b"

;;

\*)

echo "can't identify!!"

;;

esac

函数

function help(){

echo "this is help cmd"

}

case $1 in

"-h")

help

;;

\*)

echo "can't identify!!"

;;

esac

print(){

echo "param 1:$1"

echo "param 2:$2"

}

print a b

循环语句

while

while condition

do

command

done

for

for var in item1 item2 ... itemN

do

command1

command2

...

commandN

done

for ((i=1; i<=100; i++))

do

echo $i

done

until

与while相反 条件不成立则循环

例子

#!/bin/bash

echo "please input file name"

read -p "file name:" filename

test -e $filename && echo "$filename exist" || echo "$filename no exist"