

test命令：用于测试字符串、文件状态和数字

test命令有两种格式：

test condition 或[ condition ]

使用方括号时，要注意在条件两边加上空格。

文件测试

字符串测试

数值测试

符合语句测试

## 文件测试

文件测试：测试文件状态的条件表达式

-e 是否存在      -d 是目录      -f 是文件  
-r 可读            -w 可写            -x 可执行  
-L 符号连接      -c 是否字符设备    -b 是否块设备  
-s 文件非空

[https://blog.csdn.net/weixin\\_43288201](https://blog.csdn.net/weixin_43288201)

```
1 #!/bin/sh
2 read -p "请输入一个文件名" fileName
3 test -e $fileName
4 # [ -e $fileName ]
5 echo $?
```

结果：

```
1 [root@iZuf6gy6qwvk16k8vp21vrZ script]# ./test01.sh
2 请输入一个文件名text01.sh
3 0
4 [root@iZuf6gy6qwvk16k8vp21vrZ script]# ./test01.sh
5 请输入一个文件名s
6 1
```

## 字符串测试

## 字符串测试

```
test str_operator "str"
    test "str1" str_operator "str2"
    [ str_operator "str" ]
    [ "str1" str_operator "str2" ]
```

其中 str\_operator 可以是:

= 两个字符串相等    != 两个字符串不相等  
-z 空串                    -n 非空串

[https://blog.csdn.net/weixin\\_43288201](https://blog.csdn.net/weixin_43288201)

```
1 #!/bin/sh
2 test -z $yn
3 echo $? #0
4 read -p "please input y/n:" yn
5 [ -z $yn ]
6 echo "1:$?"
7
8 [ $yn = "y" ]
9 echo "2:$?"
```

结果:

```
1 [root@iZuf6gy6qwvk16k8vp21vrZ script]# ./test_str.sh
2 0
3 please input y/n:y
4 1:1
5 2:0
```

```
1 #!/bin/bash
2 read -p "请输入第一个字符串:" str1
3 read -p "请输入第二个字符串:" str2
4 # test $str1 = $str2
5 [ $str1 = $str2 ]
6 echo "结果:$?"
7
```

```
OpenSSH SSH client
[day01]. /14 sh. sh
请输入第一个字符串:hehe
请输入第二个字符串:haha
结果:1
[day01]. /14 sh. sh
请输入第一个字符串:hehe
请输入第二个字符串:hehe
结果:0
[day01]_
```

[https://blog.csdn.net/weixin\\_43288201](https://blog.csdn.net/weixin_43288201)

## 数值测试

测试数值格式如下:

test num1 num\_operator num2

[ num1 num\_operator num2 ]

num\_operator 可以是:

-eq 数值相等

-ne 数值不相等

-gt 数 1 大于数 2

-ge 数 1 大于等于数 2

-le 数 1 小于等于数 2

-lt 数 1 小于数 2

[https://blog.csdn.net/weixin\\_43288201](https://blog.csdn.net/weixin_43288201)

输入标题

	英文单词	shell比较符
相等	equal	-eq
不相等	not equal	-ne
大于	greater than	-gt
大于等于	greater equal	-ge
小于等于	less equal	-le
小于	less than	-lt

[https://blog.csdn.net/weixin\\_43288201](https://blog.csdn.net/weixin_43288201)

```
1 [root@iZuf6gy6qvwk16k8vp21vrZ script]# vim compare.sh
2
3 #!/bin/sh
4 read -p "请输入第一个数值:" data1
5 read -p "请输入第二个数值:" data2
6
7 [ $data1 -eq $data2 ]
8 echo "相等: $?"
9
10 [ $data1 -lt $data2 ]
```

```
11 echo "小于: $?"  
12  
13 [ $data1 -ge $data2 ]  
14 echo "大于等于: $?"
```

结果:

```
1 [root@iZuf6gy6qwk16k8vp21vrZ script]# ./compare.sh  
2 请输入第一个数值:2  
3 请输入第二个数值:2  
4 相等: 0  
5 小于: 1  
6 大于等于: 0
```

## 符合语句测试

命令执行控制:

&&:

command1 && command2

&&左边命令 (command1) 执行成功(即返回 0) shell 才执行&&右边的命令 (command2)

||

command1 || command2

||左边的命令 (command1) 未执行成功(即返回非 0) shell 才执行||右边的命令 (command2)

例:

test -e /home && test -d /home && echo "true"

test 2 -lt 3 && test 5 -gt 3 && echo "equal"

test "aaa" = "aaa" || echo "not equal" && echo "equal"

### 多重条件判定

-a	(and)两状况同时成立! <code>test -r file -a -x file</code> file 同时具有 r 与 x 权限时, 才为 true.
-o	(or)两状况任何一个成立! <code>test -r file -o -x file</code> file 具有 r 或 x 权限时, 就传回 true.
!	相反状态 <code>test ! -x file</code> 当 file 不具有 x 时, 回传 true.

[https://blog.csdn.net/weixin\\_3288201](https://blog.csdn.net/weixin_3288201)