

本笔记为北京千锋 V2017.1 《03 Linux Shell 脚本自动化编程实战》，笔记中所涉及到的项目均基于 Centos7u3 x86_64 或 Centos6u8 环境。

千锋所有学员均可自由使用和转载该笔记，为了尊重作者的辛劳，敬请注明出处！

Shell 条件测试

格式 1: **test** 条件表达式

格式 2: **[** 条件表达式 **]**

格式 3: **[[** 条件表达式 **]]**

man test

===文件测试 [操作符 文件或目录]

```
[root@yangs ~]# test -d /home
```

```
[root@yangs ~]# echo $?
```

0

```
[root@yangs ~]# test -d /home11111
```

```
[root@yangs ~]# echo $?
```

1

```
[root@yangs ~]# [ -d /home ]
```

[-e dir|file]

[-d dir]

[-f file] 是否存在，而且是文件

[-r file] 当前用户对该文件是否有读权限

[-x file]

[-w file]

[-L file]

```
[root@yangs ~]# [ ! -d /ccc ] && mkdir /ccc
```

```
[root@yangs ~]# [ -d /ccc ] || mkdir /ccc
```

===数值比较 [整数 1 操作符 整数 2]

[1 -gt 10] 大于

[1 -lt 10] 小于

[1 -eq 10] 等于

[1 -ne 10] 不等于

[1 -ge 10] 大于等于

[1 -le 10] 小于等于

```
[root@yangs ~]# disk_use=$(df -P |grep '/'$' |awk '{print $5}' |awk -F% '{print $1}')
```

```
[root@yangs ~]# [ $disk_use -gt 90 ] && echo "war....."
```

```
[root@yangs ~]# [ $disk_use -gt 60 ] && echo "war....."
```

war.....

```
[root@yangs ~]# id -u
```

0

```
[root@yangs ~]# [ $(id -u) -eq 0 ] && echo "当前是超级用户"
```

当前是超级用户

```
[alice@yangs ~]$ [ $UID -eq 0 ] && echo "当前是超级用户" || echo "you 不是超级用户"
```

you 不是超级用户

==C 语言风格的数值比较

```
[root@yangs ~]# ((1<2));echo $?  
0  
[root@yangs ~]# ((1==2));echo $?  
1  
[root@yangs ~]# ((1>2));echo $?  
1  
[root@yangs ~]# ((1>=2));echo $?  
1  
[root@yangs ~]# ((1<=2));echo $?  
0  
[root@yangs ~]# ((1!=2));echo $?  
0  
[root@yangs ~]# (`id -u`>0);echo $?  
1  
[root@yangs ~]# (($UID==0));echo $?  
0
```

===字符串比较

提示：使用双引号

```
[root@yangs ~]# [ "$USER" = "root" ];echo $?  
0  
[root@yangs ~]# [ "$USER" == "root" ];echo $?  
0  
  
[root@yangs ~]# BBB=""  
[root@yangs ~]# echo ${#BBB}  
0  
[root@yangs ~]# [ -z "$BBB" ] 字符长度是为 0  
[root@yangs ~]# echo $?  
0  
[root@yangs ~]# [ -n "$BBB" ] 字符长度不为 0  
[root@yangs ~]# echo $?  
1
```

=====

```
[root@yangs ~]# var1=111
```

```

[root@yangs ~]# var2=
[root@yangs ~]# //var3 变量没有定义
[root@yangs ~]# echo ${#var1}
3
[root@yangs ~]# echo ${#var2}
0
[root@yangs ~]# echo ${#var3}
0
[root@yangs ~]# [ -z "$var1" ];echo $?
1
[root@yangs ~]# [ -z "$var2" ];echo $?
0
[root@yangs ~]# [ -z "$var3" ];echo $?
0
[root@yangs ~]# [ -n "$var1" ];echo $?
0
[root@yangs ~]# [ -n "$var2" ];echo $?
1
[root@yangs ~]# [ -n "$var3" ];echo $?
1

```

小结：变量为空 或 未定义： 长度都为 0

=====

```

[root@yangs ~]# [ "$USER" = "root" ];echo $?
0
[root@yangs ~]# [ "$USER" = "alice" ];echo $?
1
[root@yangs ~]# [ "$USER" != "alice" ];echo $?
0

```

```

[root@yangs ~]# [ 1 -lt 2 -a 5 -gt 10 ];echo $?
1
[root@yangs ~]# [ 1 -lt 2 -o 5 -gt 10 ];echo $?
0

```

```

[root@yangs ~]# [[ 1 -lt 2 && 5 -gt 10 ]];echo $?
1
[root@yangs ~]# [[ 1 -lt 2 || 5 -gt 10 ]];echo $?
0

```

```

[root@yangs ~]# [ "$USER" = "root" ];echo $?

```

```

0
[root@yangs ~]# [ "$USER" =~ ^r ];echo $?
bash: [: =~: binary operator expected
2
[root@yangs ~]# [[ "$USER" =~ ^r ]];echo $? //使用正则
0

```

判断变量是不是数字：

```

[root@yangs ~]# num10=123
[root@yangs ~]# num20=ssss1114ss
[root@yangs ~]# [[ "$num10" =~ ^[0-9]+$ ]];echo $?
0
[root@yangs ~]# [[ "$num20" =~ ^[0-9]+$ ]];echo $?
1
=====

```

案例 1:

```

[root@yangs ~]# cat test02.sh
#!/bin/bash
#判断用户输入的是否是数字
read -p "请输入一个数值:" num

```

```

if [[ ! "$num" =~ ^[0-9]+$ ]];then
echo "你输入的不是数字，程序退出!!!"
exit
fi

```

```

echo ccc

```

案例 2:

```

[root@yangs ~]# cat test03.sh
#!/bin/bash
#判断用户输入的是否是数字
read -p "请输入一个数值:" num

```

```

while :
do
if [[ $num =~ ^[0-9]+$ ]];then
break
else
read -p "不是数字，请重新输入数值:" num
fi
done

```

echo "你输入的数字是:\$num"

() 子 shell 中执行

(()) 数值比较，运算 C 语言

\$() 命令替换

\$(()) 整数运算

{ }

\${ }

[] 条件测试

[[]] 条件测试，支持正则 =~

\$() 整数运算

执行脚本：

./01.sh 需要执行权限 在子 shell 中执行

bash 01.sh 不需要执行权限 在子 shell 中执行

. 01.sh 不需要执行权限 在当前 shell 中执行

source 01.sh 不需要执行权限 在当前 shell 中执行

提示：通常修改系统配置文件中如 /etc/profile 的 PATH 等变量后，使之在当前 shell 中生效

调试脚本：

sh -n 02.sh 仅调试 syntax error

sh -vx 02.sh 以调试的方式执行，查询整个执行过程

流程控制： if

=====

单分支结构

```
if 条件测试
then 命令序列
fi
```

双分支结构

```
if 条件测试
then 命令序列
else 命令序列
fi
```

多分支结构

```
if 条件测试 1
then 命令序列

[elif 条件测试 2
then 命令序列

elif 条件测试 3
then 命令序列]...

else 命令序列
fi
```

```
read -p "确认开始安装 KVM [y]: " kvm_install
if [ ! "${kvm_install}" = "y" ];then
echo -e "$red_col 输入不正确! $reset_col"
exit
fi
```