<http://blog.sina.com.cn/s/blog_6151984a0100ekl6.html>

**if 语句格式**

|  |
| --- |
| if  条件 then  Command else  Command **fi                              别忘了这个结尾** |
| If语句忘了结尾fi test.sh: line 14: syntax error: **unexpected end of fi** |

    **if 的三种条件表达式**

|  |  |
| --- | --- |
| if command then  if  函数 then | 命令执行成功，等于返回0 （比如grep ,找到匹配） 执行失败，返回非0 （grep,没找到匹配） |
| if [ expression\_r\_r\_r  ] then | 表达式结果为真，则返回0，if把0值引向then |
| if test expression\_r\_r\_r then | 表达式结果为假，则返回非0，if把非0值引向then |

        
   **[ ] &&  ——快捷if**

|  |
| --- |
| [ -f "/etc/shadow" ] && echo "This computer uses shadow passwors" |
| **&& 可以理解为then**     如果左边的表达式为真则执行右边的语句 |

    **shell的if与c语言if的功能上的区别**

|  |  |
| --- | --- |
| shell if | c语言if |
| **0为真**，走then | 正好相反，非0走then |
| 不支持整数变量直接if 必须:if [ i –ne 0 ]  但支持字符串变量直接if if [ str ] 如果字符串非0 | 支持变量直接if if (i ) |

   
=================================以command作为if 条件===================================  
    
    **以多条command或者函数作为if 条件**

|  |
| --- |
| echo –n “input:” read user  if 多条指令,这些命令之间相当于**“and”（与）** grep $user /etc/passwd >/tmp/null       who -u | grep $user then             上边的指令**都**执行成功,返回值$?为0，**0为真**，运行then  echo "$user has logged" else     指令执行失败，$?为1，运行else                              echo "$user has not logged" fi |
| # sh test.sh input : macg macg     pts/0        May 15 15:55   .          2075 (192.168.1.100) macg has logged     # sh test.sh input : ddd ddd has not logged |

    **以函数作为if条件  (函数就相当于command,函数的优点是其return值可以自定义)**

|  |
| --- |
| if 以函数作为if条件， getyn then   函数reture值**0为真**，走then echo " your answer is yes" else  函数return值非0为假，走else echo "your anser is no" fi |

   **if command  等价于 command+if $?**

|  |  |
| --- | --- |
| $ vi testsh.sh #!/bin/sh  **if cat 111-tmp.txt | grep ting1 then** echo found else echo "no found" fi | $ vi testsh.sh #!/bin/sh  **cat 111-tmp.txt | grep ting1**  **if [ $? -eq 0 ]** **then** **echo $?** echo found else **echo $?** echo "no found" fi |
| $ sh testsh.sh no found | $ sh testsh.sh 1 no found |
| $ vi 111-tmp.txt that is 222file thisting1 is 111file  $ sh testsh.sh thisting1 is 111file found | $ vi 111-tmp.txt that is 222file thisting1 is 111file  $ sh testsh.sh thisting1 is 111file 0 found |

     
========================================以条件表达式作为 if条件=============================  
  
    **传统if 从句子——以条件表达式作为 if条件**  
if [ 条件表达式 ]  
then  
 command  
 command  
 command  
else  
 command  
 command  
fi  
     
**条件表达式**

* 文件表达式
* 整数变量表达式

**if [ int1 -eq int2 ]    如果int1等于int2     
if [ int1 -ne int2 ]    如果不等于**   
if [ int1 -ge int2 ]       如果>=  
if [ int1 -gt int2 ]       如果>  
if [ int1 -le int2 ]       如果<=  
if [ int1 -lt int2 ]       如果<  
   

* 字符串变量表达式

If  [ $a = $b ]                 如果string1等于string2  
                                **字符串允许使用赋值号做等号**  
if  [ $string1 !=  $string2 ]   如果string1不等于string2         
**if  [ -n $string  ]             如果string 非空(非0），返回0(true)**    
if  [ -z $string  ]             如果string 为空  
**if  [ $sting ]                  如果string 非空，返回0 (和-n类似)**    
  
  
    **条件表达式引用变量要带$**

|  |
| --- |
| if [ a = b ] ;then     echo equal else echo no equal fi |
| [macg@machome ~]$ sh test.sh input a: 5 input b: 5 **no equal  （等于表达式没比较$a和$b,而是比较和a和b,自然a!=b)** |

改正：

|  |
| --- |
| if [ $a = $b ] ;then        echo equal else echo no equal fi |
| [macg@machome ~]$ sh test.sh input a: 5 input b: 5 equal |

                                                                                     
  **-eq  -ne  -lt  -nt只能用于整数，不适用于字符串，字符串等于用赋值号=**

|  |
| --- |
| [macg@machome ~]$ vi test.sh echo -n "input your choice:" read var if  [ $var -eq "yes" ] then echo $var fi [macg@machome ~]$ sh -x test.sh input your choice: y **test.sh: line 3: test: y: integer expression\_r\_r\_r expected                        期望整数形式，即-eq不支持字符串** |

  **=放在别的地方是赋值,放在if [ ] 里就是字符串等于,shell里面没有==的,那是c语言的等于**  
  
  
   **无空格的字符串，可以加" ",也可以不加**

|  |
| --- |
| [macg@machome ~]$ vi test.sh echo "input a:" read a echo "input is $a" if [ $a = 123 ] ; then echo equal123 fi |
| [macg@machome ~]$ sh test.sh input a: 123 input is 123 equal123 |

 **= 作为等于时，其两边都必须加空格，否则失效**  
等号也是操作符，必须和其他变量，关键字，用空格格开 (等号做赋值号时正好相反，两边不能有空格）

|  |  |
| --- | --- |
| [macg@machome ~]$ vi test.sh  echo "input your choice:" read var if [ **$var="yes"** ] then echo $var echo "input is correct" else echo $var echo "input error" fi | [macg@machome ~]$ vi test.sh  echo "input your choice:" read var if [ **$var = "yes"** ]   在等号两边加空格 then echo $var echo "input is correct" else echo $var echo "input error" fi |
| [macg@machome ~]$ sh test.sh input your choice: y y input is correct [macg@machome ~]$ sh test.sh input your choice: n     n input is correct  输错了也走then,都走then,为什么? **因为if把$var="yes"连读成一个变量，而此变量为空，返回1，则走else** | [macg@machome ~]$ sh test.sh input your choice: y y input error [macg@machome ~]$ sh test.sh input your choice: no                        no input error 一切正常 |

    **If  [  $ANS  ]     等价于  if [ -n $ANS ]**  
      如果字符串变量非空（then） , 空(else)

|  |
| --- |
| echo "input your choice:" read ANS  if [ $ANS ] then echo no empty else echo empth fi |
| [macg@machome ~]$ sh test.sh input your choice:                       回车                                                  empth                                    **说明“回车”就是空串** [macg@machome ~]$ sh test.sh input your choice: 34 no empty |

   
    **整数条件表达式，大于，小于**，s**hell里没有> 和< ,会被当作尖括号，只有-ge,-gt,-le,lt**

|  |
| --- |
| [macg@machome ~]$ vi test.sh  echo "input a:" read a if  [ $a -ge 100 ] ; then echo 3bit else echo 2bit fi |
| [macg@machome ~]$ sh test.sh input a: 123 3bit [macg@machome ~]$ sh test.sh input a: 20 2bit |

 **整数操作符号-ge,-gt,-le,-lt, 别忘了加-**

|  |
| --- |
| if  test $a  **ge** 100 ; then  [macg@machome ~]$ sh test.sh test.sh: line 4: test: ge: binary operator expected |
| if  test $a -ge 100 ; then  [macg@machome ~]$ sh test.sh input a: 123 3bit |

============================逻辑表达式=========================================  
 **逻辑非 !                   条件表达式的相反**  
if [ ! 表达式 ]  
if [ ! -d $num ]                        如果不存在目录$num  
  
  
**逻辑与 –a**                    **条件表达式的并列**  
if [ 表达式1  –a  表达式2 ]  
  
  
    **逻辑或 -o**          **条件表达式的或**  
if [ 表达式1  –o 表达式2 ]  
  
    **逻辑表达式**

* 表达式与前面的=  != -d –f –x -ne -eq -lt等合用
* 逻辑符号就正常的接其他表达式，没有任何括号（ ），就是并列

if [ -z "$JHHOME" -a -d $HOME/$num ]

* 注意逻辑与-a与逻辑或-o很容易和其他字符串或文件的运算符号搞混了

  **最常见的赋值形式，赋值前对=两边的变量都进行评测**  
左边测变量是否为空，右边测目录(值)是否存在（值是否有效）

|  |
| --- |
|  |
| [macg@mac-home ~]$ vi test.sh : echo "input the num:" read num echo "input is $num"  **if [ -z "$JHHOME" -a -d $HOME/$num ]** 如果变量$JHHOME为空，且$HOME/$num目录存在 **then JHHOME=$HOME/$num**                   则赋值 fi  echo "JHHOME is $JHHOME" |
| ----------------------- [macg@mac-home ~]$ sh test.sh input the num: ppp input is ppp JHHOME is  目录**-d $HOME/$num** 不存在，所以$JHHOME没被then赋值 |
| [macg@mac-home ~]$ mkdir ppp [macg@mac-home ~]$ sh test.sh input the num: ppp input is ppp JHHOME is /home/macg/ppp |

 **一个-o的例子，其中却揭示了”=”必须两边留空格的问题**

|  |
| --- |
| echo "input your choice:" read ANS  if [ **$ANS="Yes"** -o $ANS="yes" -o $ANS="y" -o $ANS="Y" ] then ANS="y" else ANS="n" fi  echo $ANS |
| [macg@machome ~]$ sh test.sh input your choice: n y [macg@machome ~]$ sh test.sh input your choice: no y 为什么输入不是yes,结果仍是y(走then） 因为=被连读了，成了变量**$ANS="Yes"，而变量又为空，所以走else了** |

|  |
| --- |
| [macg@machome ~]$ vi test.sh  echo "input your choice:" read ANS    echo "input your choice:" read ANS  if [ **$ANS = "Yes"** -o $ANS = "yes" -o $ANS = "y" -o $ANS = "Y" ] then ANS="y" else ANS="n" fi  echo $ANS |
| [macg@machome ~]$ sh test.sh input your choice: no n [macg@machome ~]$ sh test.sh input your choice: yes y [macg@machome ~]$ sh test.sh input your choice: y y |

===================以  test 条件表达式 作为if条件===================================  
  
    **if test $num -eq 0      等价于   if [ $num –eq 0 ]**  
  
  **test  表达式,没有 [  ]**  
if test $num -eq 0                  
then  
echo "try again"  
else  
echo "good"  
fi  
  
**man test**

|  |
| --- |
| [macg@machome ~]$ man test [(1)                             User Commands                            [(1)  SYNOPSIS        test EXPRESSION        [ EXPRESSION ]          [-n] STRING               the length of STRING is nonzero          -n和直接$str都是非0条件         -z STRING               the length of STRING is zero         STRING1 = STRING2               the strings are equal         STRING1 != STRING2               the strings are not equal         INTEGER1 -eq INTEGER2               INTEGER1 is equal to INTEGER2         INTEGER1 -ge INTEGER2               INTEGER1 is greater than or equal to INTEGER2         INTEGER1 -gt INTEGER2               INTEGER1 is greater than INTEGER2         INTEGER1 -le INTEGER2               INTEGER1 is less than or equal to INTEGER2         INTEGER1 -lt INTEGER2               INTEGER1 is less than INTEGER2         INTEGER1 -ne INTEGER2               INTEGER1 is not equal to INTEGER2         FILE1 -nt FILE2               FILE1 is newer (modification date) than FILE2         FILE1 -ot FILE2               FILE1 is older than FILE2         -b FILE               FILE exists and is block special         -c FILE               FILE exists and is character special         -d FILE               FILE exists and is a directory         -e FILE               FILE exists                                 文件存在         -f FILE               FILE exists and is a regular file     文件存在且是普通文件         -h FILE               FILE exists and is a symbolic link (same as -L)         -L FILE               FILE exists and is a symbolic link (same as -h)         -G FILE               FILE exists and is owned by the effective group ID         -O FILE               FILE exists and is owned by the effective user ID         -p FILE               FILE exists and is a named pipe          -s FILE               FILE exists and has a size greater than zero         -S FILE               FILE exists and is a socket         -w FILE               FILE exists and is writable         -x FILE FILE exists and is executable |

======================if简化语句=================================  
  
    **最常用的简化if语句**

|  |
| --- |
| **&& 如果是“前面”，则“后面”** [ -f /var/run/dhcpd.pid ] && rm /var/run/dhcpd.pid    检查 文件是否存在，如果存在就删掉 |
| **||   如果不是“前面”，则后面** [ -f /usr/sbin/dhcpd ] || exit 0    检验文件是否存在，如果存在就退出 |

    **用简化 if 和$1,$2,$3来检测参数，不合理就调用help**  
**[ -z "$1" ] && help                 如果第一个参数不存在（-z  字符串长度为0 ）**  
[ "$1" = "-h" ] && help                        如果第一个参数是-h,就显示help  
  
  
**例子**  
#!/bin/sh  
  
[ -f "/etc/sysconfig/network-scripts/ifcfg-eth0" ] && rm -f /etc/sysconfig/network-scripts/ifcfg-eth0  
cp ifcfg-eth0.bridge /etc/sysconfig/network-scripts/ifcfg-eth0  
  
[ -f "/etc/sysconfig/network-scripts/ifcfg-eth1" ] && rm -f /etc/sysconfig/network-scripts/ifcfg-eth1  
cp ifcfg-eth1.bridge /etc/sysconfig/network-scripts/ifcfg-eth1  
  
[ -f "/etc/sysconfig/network-scripts/ifcfg-eth0:1" ] && rm -f /etc/sysconfig/network-scripts/ifcfg-eth0:1