1. 实现某种操作：总是测试 前提 是否满足
2. 逻辑运算：
   1. 布尔运算：真、假
   2. 与&&、或||、非!、异或^
3. 命令都有其状态返回值：成功——0，真；失败——1-255，假
4. Bash条件测试：
   1. 命令执行成功与否即为条件测试
   2. 比较运算：>,<,>=,<=,==,!=,=~
   3. Test命令：test EXPR，例如test $n1 > $n2；echo $?——输出命令成功与否
   4. [ EXPR ]：
   5. [[ EXPR ]]：关键字
   6. 测试类型：根据比较时的操作数的类型
5. 整型测试：整数比较
6. 字符测试：字符串比较
7. 文件测试：判断文件的存在性及属性等
8. 注意：比较运算通常只在同一类型间进行
   1. 整型测试：
9. –gt：例如[ $n1 –gt $n2]
10. –lt：例如[ $n1 –lt $n2]
11. –ge：例如[ $n1 –ge $n2]
12. –le：例如[ $n1 –le $n2]
13. –eq：例如[ $n1 –eq $n2]
14. –nq：例如[ $n1 –nq $n2]
    1. 字符串测试：
15. 双目：
    1. >：[ “$str1” > “$str2” ]
    2. <：
    3. >=：
    4. <=：
    5. ==：
    6. !=：
16. 单目：
    1. –n String：是否不空，不空则为真，空为假
    2. –z String：是否为空，空为真，不空为假
17. 过程式编程：顺序，选择（if和case），循环（for）
18. If：三种格式语句
    1. If 测试条件; then
19. 在bash中是命令，（test EXPR或[ EXPR ]）或关键字（[[ EXPR ]]）
20. If 命令：在bash运行至if时，其后的命令会被执行，其状态结果作为判断标准：0表示真，1-255表示假；
21. 如果条件包含比较之意，则必须使用上述三种格式，即（test EXPR或[ EXPR ]）或关键字（[[ EXPR ]]）
    1. 单分支的if语句：
22. if 测试条件；then

选择分支（表示条件测试状态返回值为真则执行选择分支）

fi

例if ! id $userName &> /dev/null; then

Useradd &username

Fi

1. 练习：写一个脚本，接受一个参数，参数是用户名；用户存在显示其用户ID
   1. 双分支的if语句：
2. if 测试条件； then

选择分支1

else

选择分支2

fi

1. 上面两个分支执行其中之一
2. 练习：通过命令行传递两个整数参数给脚本，脚本可以返回较大数
3. 练习：通过命令行传递任意个整数给脚本，脚本可以返回其大者
4. 练习：通过命令行给定一个文件路径，而后判断：如果此文件中存在空白行，则显示其空白行的数目；否则显示无空白行
5. 注意：如果把命令执行成功与否当作条件，则if语句后必须只跟命令本身，而不能引用（加反引号）
   1. 多分支的if语句：
6. if 测试条件1； then

选择分支1

elif 测试条件2； then

选择分支2

elif 测试条件3； then

选择分支3

…

else

选择分支n

fi

1. 练习：传递一个参数给脚本，如果参数为quit，显示你要退出；参数为yes，显示你要继续；其他任意参数，则显示无法识别
2. 练习：传递一个用户名给脚本，如果此用户的id号为0，则显示这是管理员；如果用户id大于等于500，显示这是普通用户；否则，这是系统用户。
3. 自定义shell进程的状态返回值：exit [n]