##### Bash弱类型

1. 变量=值
   1. 任何变量无需事先声明，可直接使用；
   2. 值默认都是字符型；
   3. a=abc（加括号也可以，子串中间有空白字符必须加引号），b=3
   4. 赋值：a=4（等号两端不可随意添加空格）
   5. 增强型赋值机制：+=，-=，\*=，/=，%=，（a=$[$a+1]相当于let a+=1,export PATH=$PATH;/usr/local/apache/bin）
   6. 自加/减：++，--（++v,v++，--v，v--）
   7. Unset：撤销变量
2. 算术运算：bash会对数字执行隐式类型转换
   1. let VAR\_NAME=Integer\_Value
   2. declare –i Var\_Name=Integer\_Value
   3. 操作符：+，-，\*，/，%，\*\*（次方）——双目操作符
3. Bash算术运算的方式：
   1. let Var\_Name=EXPRESSION(let res=$na\*$n2;echo $res)，依赖于let命令，是一个语句
   2. bash不支持浮点运算，所得结果近似为整数
   3. Var\_Name=$[EXPRESSION]，不依赖于外部命令，是一个表达式，可用$[EXPRESSION]直接引用结果，可以不用赋值
   4. $((EXPRESSION))等价于$[EXPRESSION]
   5. expr ARG1 OP AGR2（expr是独立的命令，使用使要使用命令引用，必须加空格）
   6. 显示子串且子串中无变量可用单双引号，子串中有变量如果要替换必须用双引号，不想替换用单引号，引用的结果是一个命令要用反引号（‘1’左边的键）
   7. 练习：创建10个用户：tuser601-tuser610
   8. Bash 中的子串连接：变量引用后方直接跟子串时变量名要加花括号，例如animal=pig; echo “${animal}s”
4. 练习：
   1. 求100以内所有的整数和
   2. 计算当前系统上的所有用户的id之和
   3. 写一个脚本：
5. 脚本可以接受一个以上的文件路径作为参数；
6. 显示每个文件的行数；
7. 计算本次对多少个文件执行了行数统计；
8. 显示文件的总行数。
   1. 新建10个用户：tuser401-tuser410，并求它们的id之和
   2. 写一个脚本：
9. 创建用户tuser501-tuser510；
10. 创建目录/tmp/dir-当前日期时间；
11. 在/tmp/dir-当前日期时间 目录中创建10个空文件夹file101-file110；
12. 将file101的属主改为tuser501，以此类推，一直将file110的属主改为tuser510.
    1. 分别统计 /etc/rc.d/rc.sysinit、/etc/rc.d/init.d/functions和/etc/inittab文件中以#开头的行的行数和空白行数
    2. 写一个脚本：显示当前系统上所有默认shell为bash的用户的用户名、UID及其所有此类用户的UID之和
    3. 写一个脚本，显示当前系统上有附加组的用户的用户名，并统计有多少个此类用户