1+1

4-6

6/2-1

433\*3

((1+2-3))/4

a=1

b<-2

3->c

#d<-4

#命名不要以数字开头，注意大小写，不能含特殊字符，

#命名不要重复

#ls()

#rm(list=ls())

#\_zhao\_<-3 下划线不能在前

#单等号赋值，双等号判断

.z<-3

?ls

DNA<-c("A","T","C","G")

sample(x=DNA,size=1)

#args 参数解析

#c combine

#size次数 replace 有放回抽样，系统默认无放回抽样

sample(x=DNA,size=2,replace=TRUE)

#构建函数：命名 参数 括号

turn<-function(){

DNA<-c("A","T","C","G")

dinucleotide<-sample(x=DNA,size=2,replace=TRUE)

dinucleotide

}

turn()

#默认参数：不给的话从中选，给的话从给的里面选

#要先运行上面的turn，再运行turn()

turn<-function(wheel){

DNA<-c("A","T","C","G")

dinucleotide<-sample(x=DNA,size=2,replace=TRUE)

dinucleotide

}

turn3<-function(wheel=c("A","T","C","G"),n=2){

dinucleotide<-sample(x=wheel,size=2,replace=TRUE)

dinucleotide

}

turn2<-function(wheel=c("A","T","C","G")){

dinucleotide<-sample(x=wheel,size=2,replace=TRUE)

dinucleotide

}

turn2()

turn3()

turn3(wheel=1:4)

turn4<-function(wheel=1:4,n=2){

dinucleotide<-sample(x=wheel,size=2,replace=TRUE)

sum(dinucleotide)

}

turn4()

#replicate()

replicate(1000,turn4())

x<-replicate(1000,turn4())

hist(x)

colors()

#sample 的概率功能

turn5<-function(wheel=1:4,n=2){

dinucleotide<-sample(x=wheel,size=2,replace=TRUE,prob=c(0.1,0.2,0.3,0.4))

sum(dinucleotide)

}

turn(5)

replicate(1000,turn(5))

#原子向量 可用typeof 来查是哪种类型，

#数值型，字符型,整形，逻辑性，复杂型，粗糙性

#数值型 字符型混合在一起，会判断为字符型

NUM<-c(1,2,3,4)

typeof(NUM)

NUM2<-c(1L,2L,3L,4L)

typeof(NUM2)

AGS<-c("V","1","2L")

typeof(AGS)

#NUM 把原子数据改成矩阵

NUM3<-1:20

dim(NUM3)<-c(2,2,5)

NUM3

#改变排列顺序

m<-matrix(data=NUM3,nrow=10,ncol=2)

m

#array 数组 三维

array(data=NUM3,dim=c(2,2,5))

#向下取整

array(data=NUM3,dim=c(0.5,4,5))

array(data=NUM3,dim=c(4,5))