主要参考programming in haskell ppt

chap1:

在linux 下装上hugs，然后可以运行一些简单的运算 1+1

退出 :quit

还可以在文档中写函数 double x=x+x

存为.hs文件

在路径下面hugs test.hs

如果改变了文件内容 :reload 就可以使用函数了

chap3: type

add x y =x+y 一次获取两个值，叫做curried functionS

mult x y z =((multx)y)z

Num 数字类型

Eq 等

Ord 有序

(+) :: Num a ? a ? a ? a

(==) :: Eq a ? a ? a ? Bool

(<) :: Ord a ? a ? a ? Bool

chap4:

主要讲了curry这个概念，讲了一些变量类型

chap5:

-lists

> [(x,y) | y ? [4,5], x ? [1,2,3]]

[(1,4),(2,4),(3,4),(1,5),(2,5),(3,5)]

dependent generators

concat :: [[a]] ? [a]

concat xss = [x | xs ? xss, x ? xs]

> concat [[1,2,3],[4,5],[6]]

[1,2,3,4,5,6]

guards

[x | x ? [1..10], even x]

zip function

> zip [’a’,’b’,’c’] [1,2,3,4]

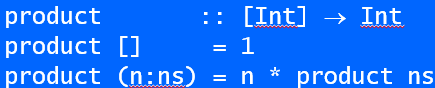
[(’a’,1),(’b’,2),(’c’,3)]

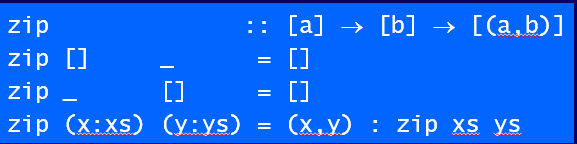
可以扩展为pairs结对函数，结对函数可以判断是否有序

string

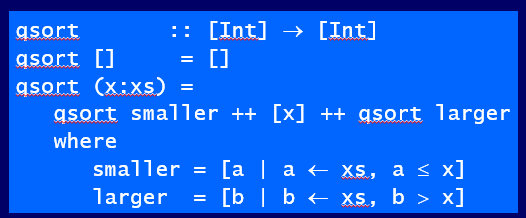
# chap6:递归

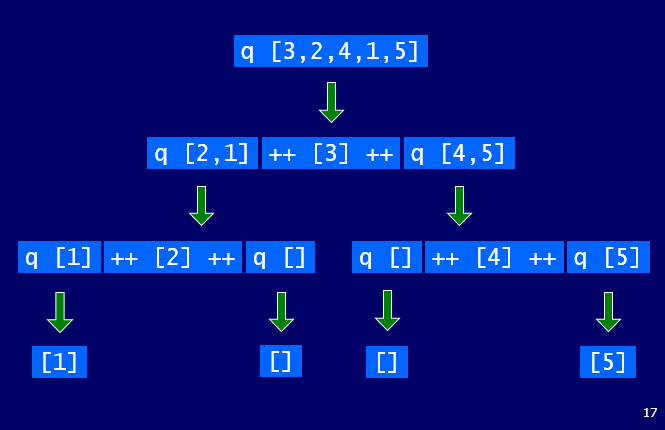
递归实现（带list）





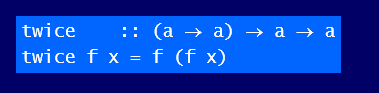
快速排序的实现



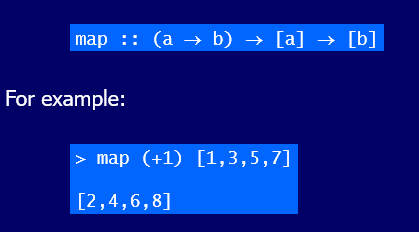


# Chap7 higher order functions

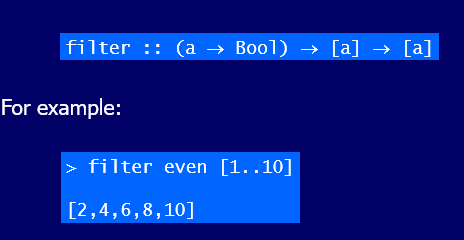
把一个函数作为输入或者输出返回的函数叫做高级函数



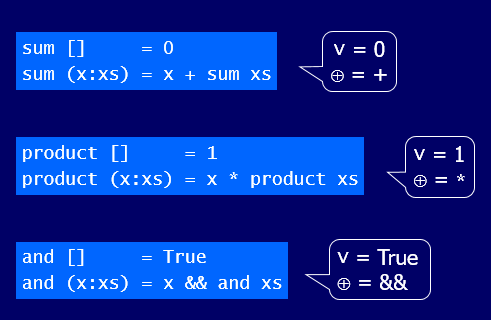
Map：在list中每个元素上作用一个函数

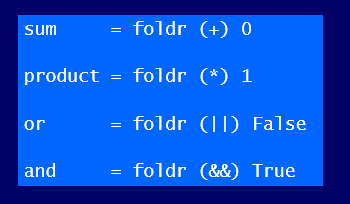


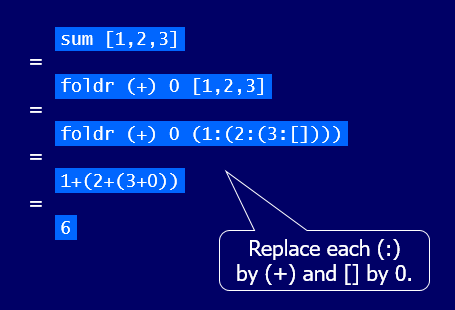
Filter:选出list中满足条件的元素



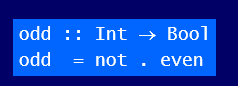
Foldr function:



简写



.：把两个函数合并成一个



# Chap8 functional parsers

Item: 输入为空则失败，否则返回第一个字母

Failure: 返回失败

Return v: 返回成功，不输入任何值返回v

P+++q:如果p成功就使用parser p 否则q

Parse:把一个parser用于一个string