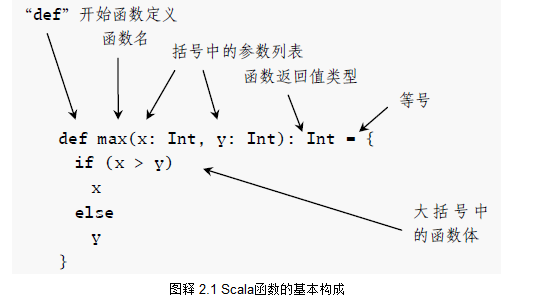
# Scala概念

1. 仿函数：
2. 类定义：
3. 构造函数：
4. 数据结构：
5. 数据集合：<http://twitter.github.io/scala_school/zh_cn/collections.html>
6. 模式匹配（case类）：<http://scalachina.com/node/20>
7. 类型（和模式匹配有关，涉及“上下文”）：
8. This关键字：
9. 特质/trail：相当于Java的接口，一个trait 就是把一些共同的性质抽象出来，哪个类需要就混入.
10. –> ：可以用来创建二元组。<http://twitter.github.io/scala_school/zh_cn/collections.html>
11. 星号\*作用：当导入一个包中所有的类或者符号时，你应该使用下划线（**\_**）而不是星号（\*），星号在Scala中是一个有效的标识符（例如作为方法名称）；
12. 关键字some（和option有关）：
13. 抽象类，以及抽象关键字Abstract：

# 问题

1. List集合可以使用：：串接两个List，那如果对两个整数呢？ 如 1 ：： 2？ 如果可以，那么在List、Array[]等进行集合操作，如map后串接这些返回值。
2. 如何定义一个函数，函数返回值，如何写？

答：参考<http://scalachina.com/node/50>，如下图所示



1. 为什么在抽象类中，可以把变量类型定义为type、def等？就像<http://blog.csdn.net/y3wegy/article/details/8618169>里面的  
   abstract class BuilkReader {    
     type In    
     val source:In    
     def read:String    
   }