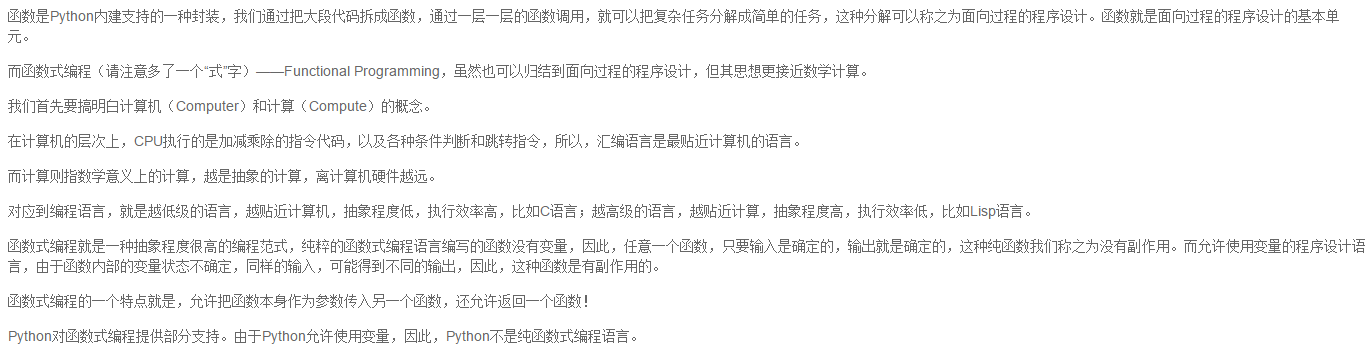
**一．函数式编程**

**1.高阶函数** 函数本身也可以赋值给变量，即：变量可以指向函数。

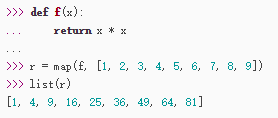
函数名也是变量。对于abs()这个函数，完全可以把函数名abs看成变量，它指向一个可以计算绝对值的函数！

既然变量可以指向函数，函数的参数能接收变量，那么一个函数就可以接收另一个函数作为参数，这种函数就称之为高阶函数。编写高阶函数，就是让函数的参数能够接收别的函数。把函数作为参数传入，这样的函数称为高阶函数，函数式编程就是指这种高度抽象的编程范式。

**1-1.map/reduce**

Python内建了map()和reduce()函数。

map()函数接收两个参数，一个是函数，一个是Iterable，map将传入的函数依次作用到序列的每个元素，并把结果作为新的Iterator返回。map()传入的第一个参数是f，即函数对象本身。由于函数返回结果是一个Iterator，Iterator是惰性序列，因此通过list()函数让它把整个序列都计算出来并返回一个list。



reduce把一个函数作用在一个序列[x1, x2, x3, ...]上，这个函数必须接收两个参数，reduce把结果继续和序列的下一个元素做累积计算，其效果就是：



**1-2.filter**

Python内建的filter()函数用于过滤序列。和map()类似，filter()也接收一个函数和一个序列。和map()不同的是，filter()把传入的函数依次作用于每个元素，然后根据返回值是True还是False决定保留还是丢弃该元素。可见用filter()这个高阶函数，关键在于正确实现一个“筛选”函数。

filter()的作用是从一个序列中筛出符合条件的元素。由于filter()使用了惰性计算，所以只有在取filter()结果的时候，才会真正筛选并每次返回下一个筛出的元素。

**1-3.sorted**

**2.返回函数**

**3.匿名函数**

**4.装饰器**

**5.偏函数**

**二．模块**

**1.使用模块**

**2.安装第三方模块**