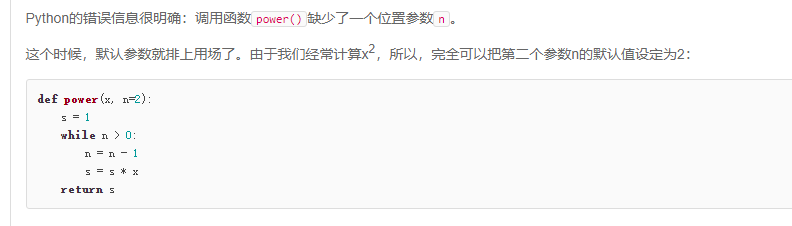
# Python的哲学就是简单优雅，尽量写容易看明白的代码，尽量写少的代码。

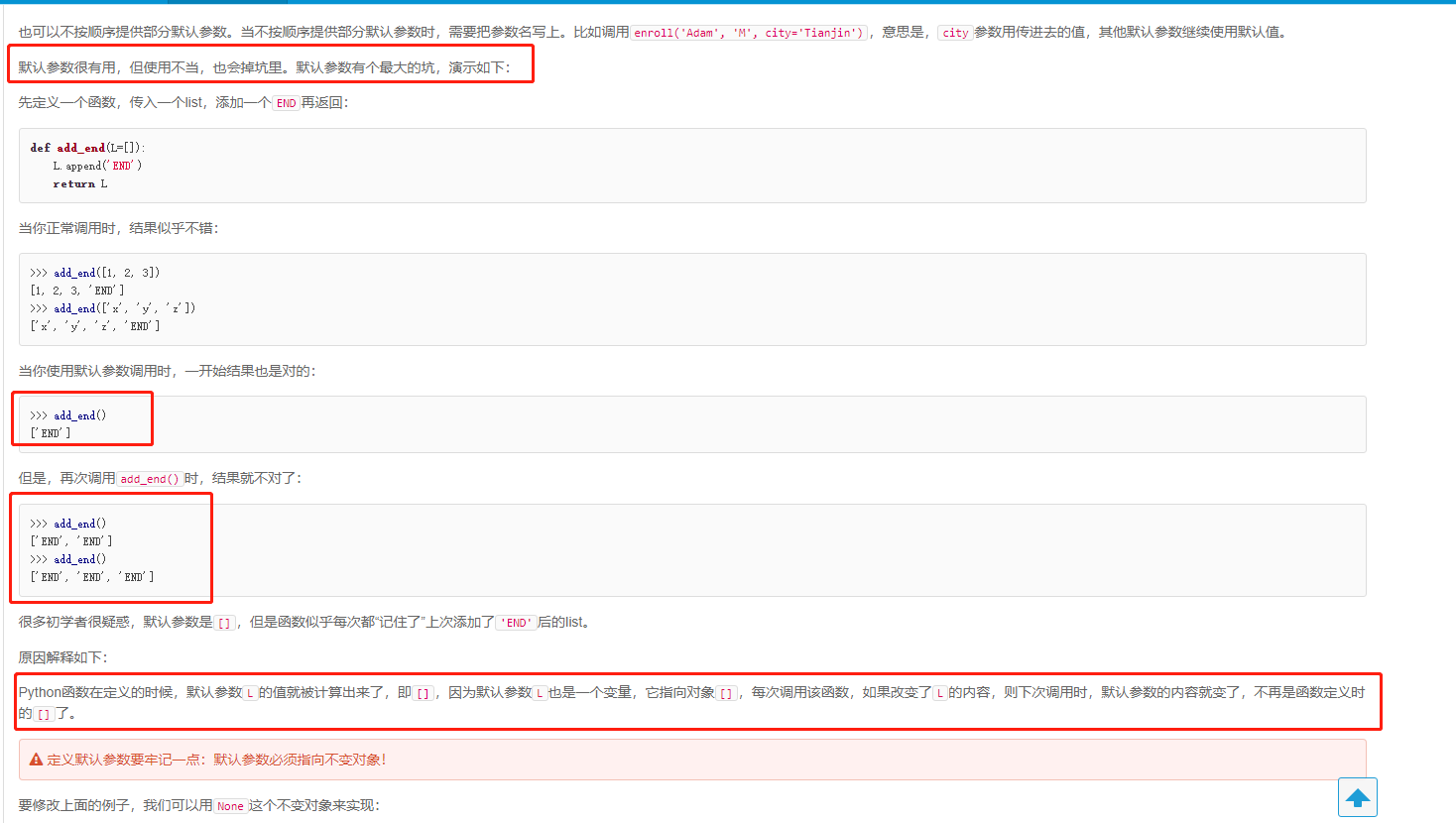
# 函数

## 定义函数参数给默认值



默认参数要放在必填参数的后边,如果有多个默认参数,那么变化大的(也就是经常会输入的放在前边)

在不需要输入第一个默认参数而需要输入第二个默认参数的时候



所以定义参数默认值应该使用不变对象来实现.

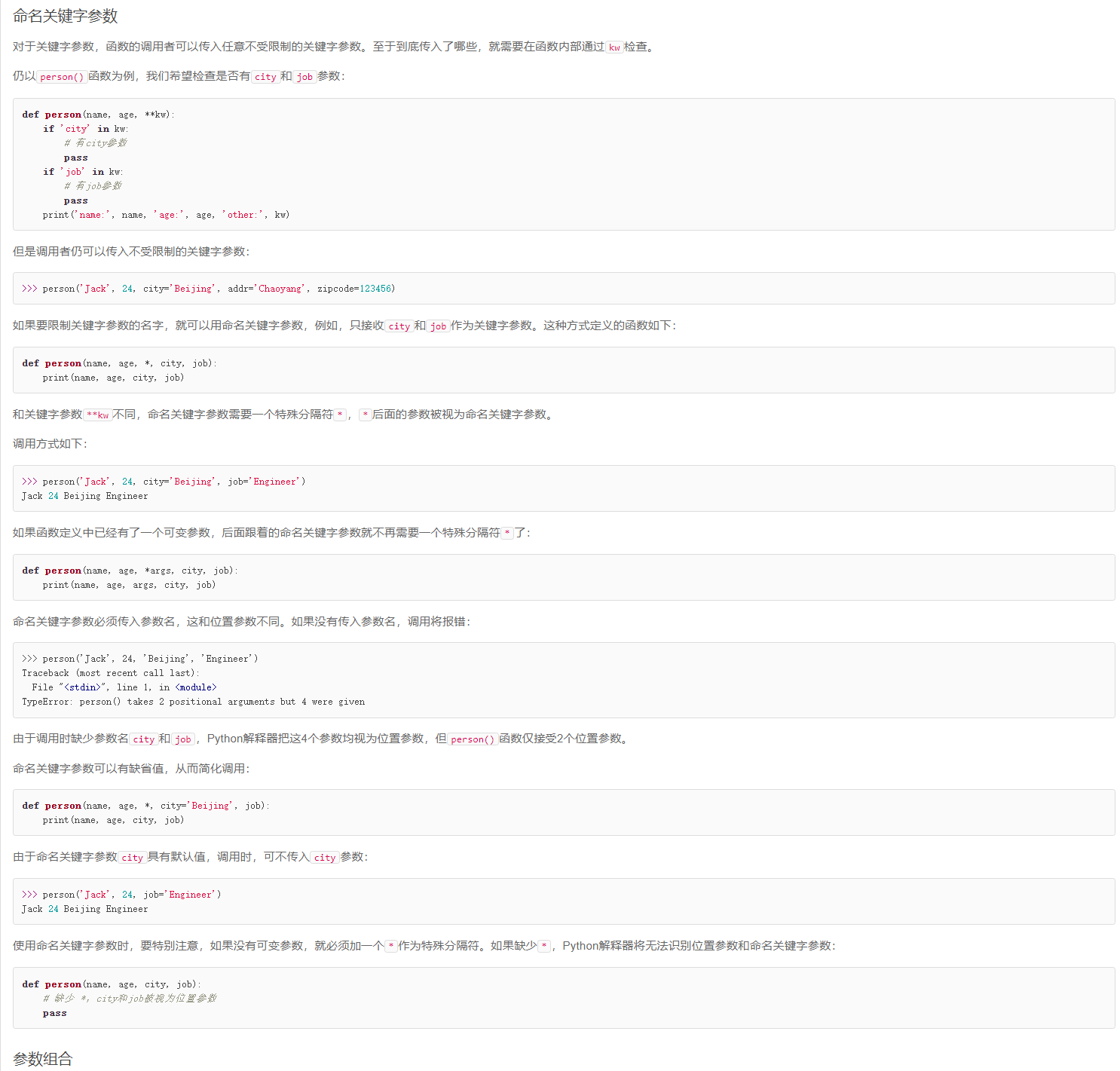
## 可变参数



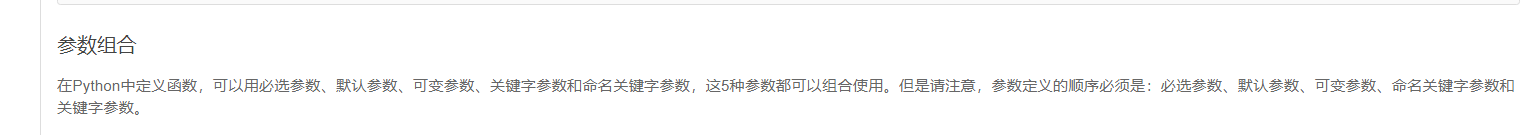
## 关键字参数



## 命名关键字参数



## 参数的使用顺序

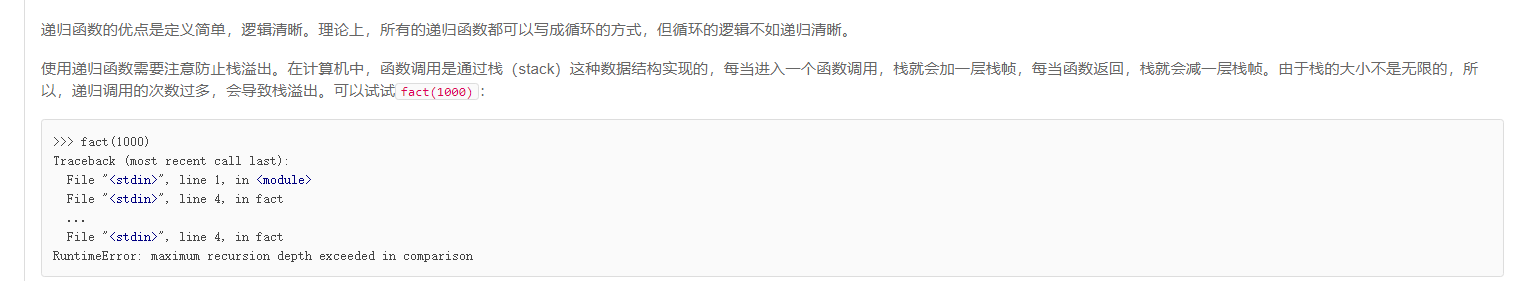


## 小结



# 递归

## 注意事项



## 解决栈溢出的问题

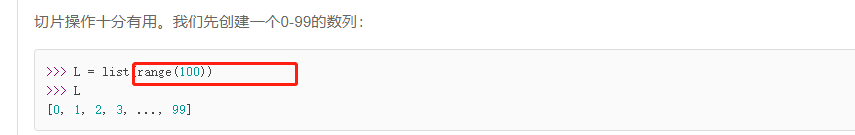


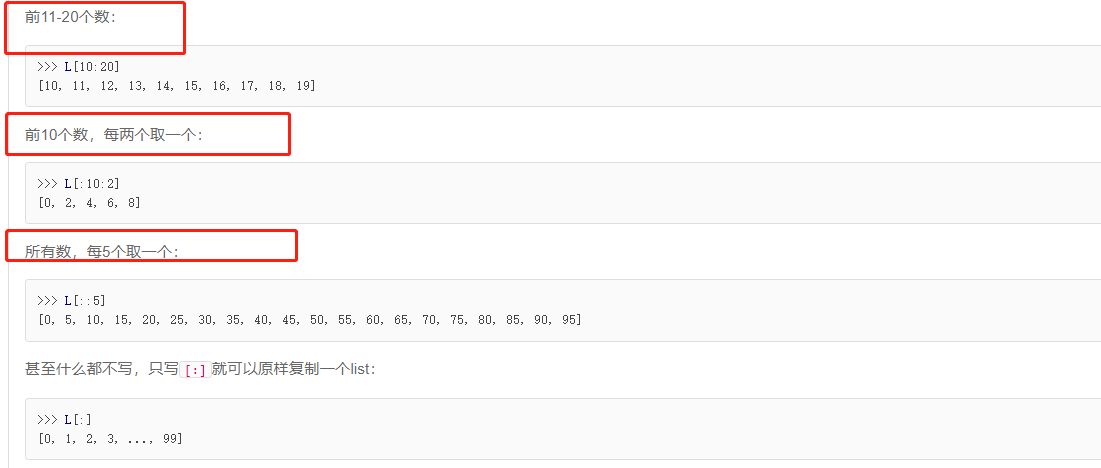
# 高级特性

## 切片



range还能这样用,貌似range的作用就是生成n个数字



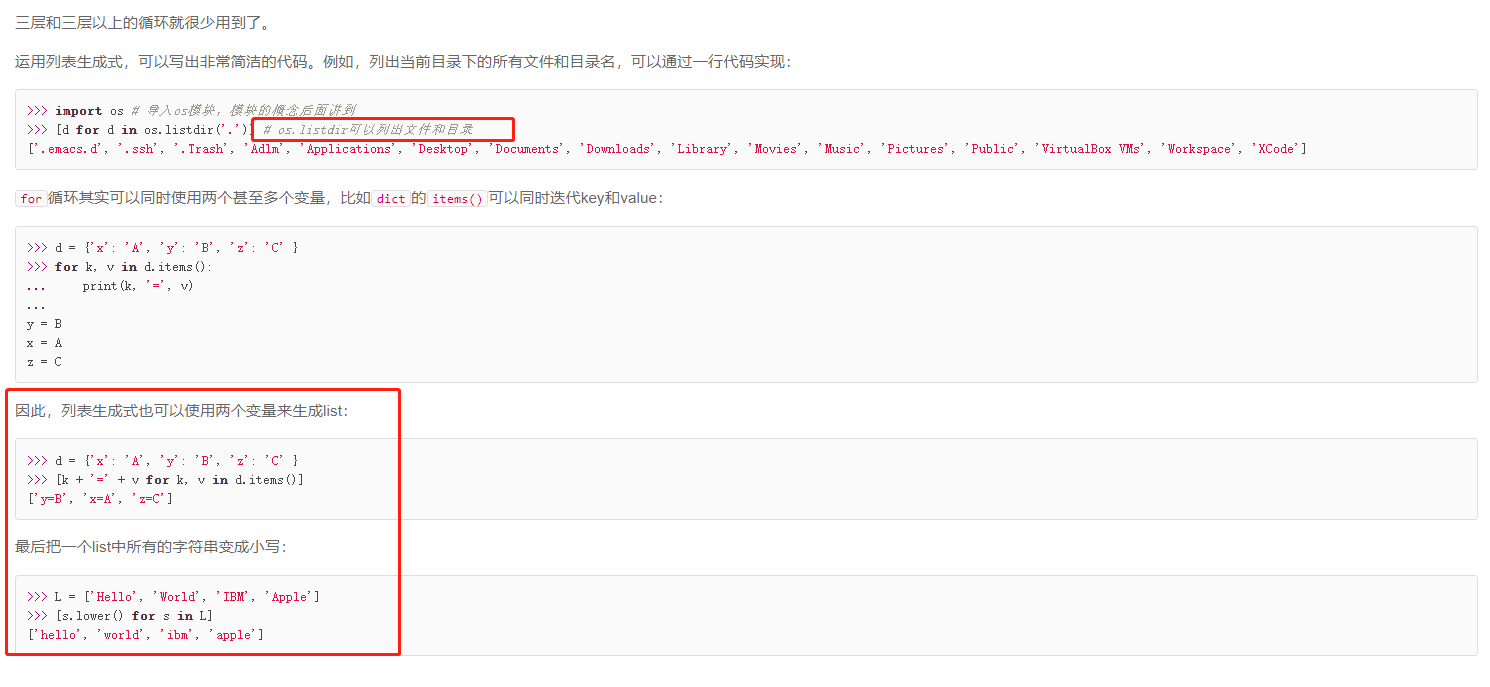


## 迭代

迭代dic

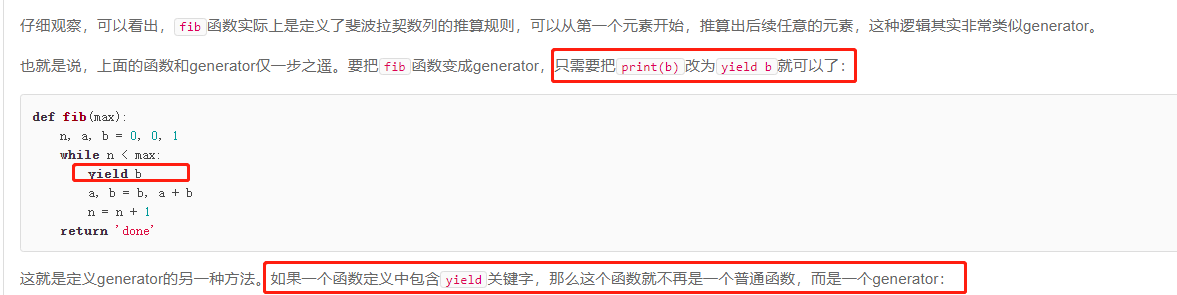


## 列表生成式



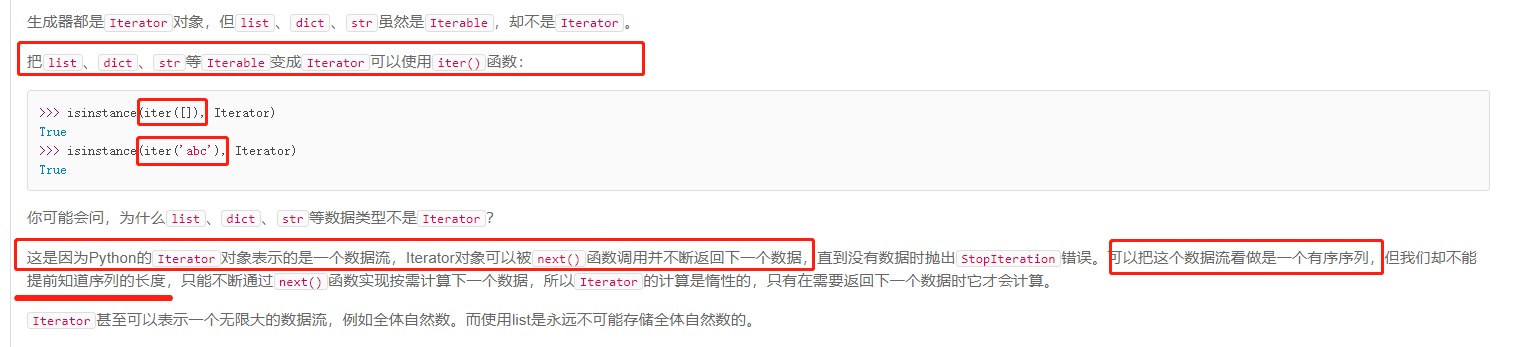
## 生成器





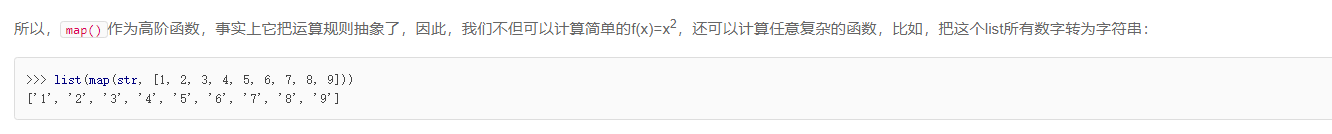
这里调用next遇到yield才会停止,而for循环不会停止

## 迭代器



# 函数式编程

## 高阶函数

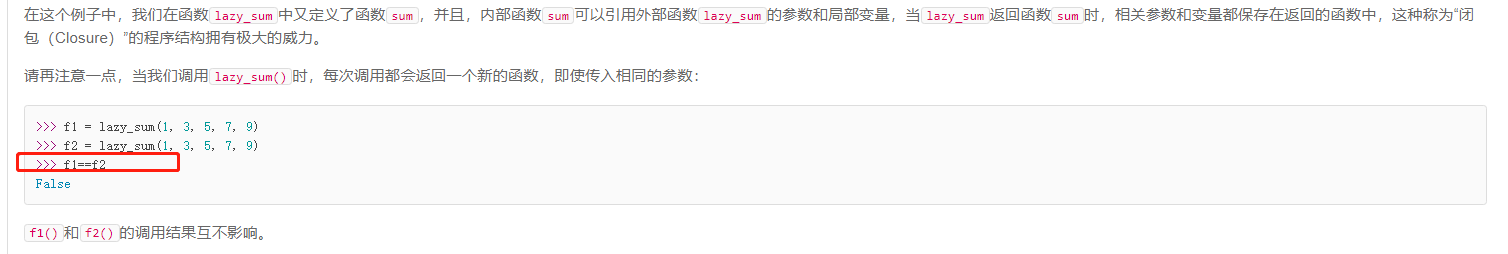


filter

sort



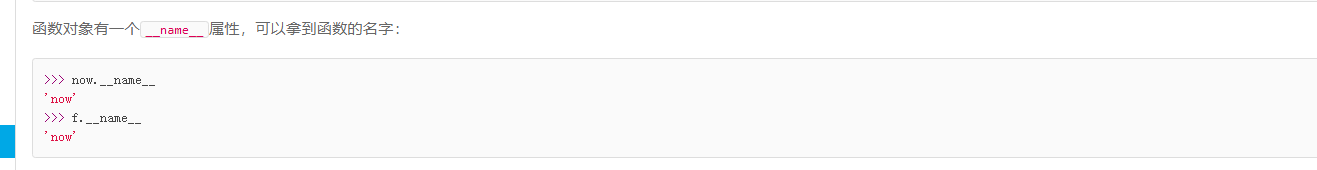
## 返回函数



## 匿名函数

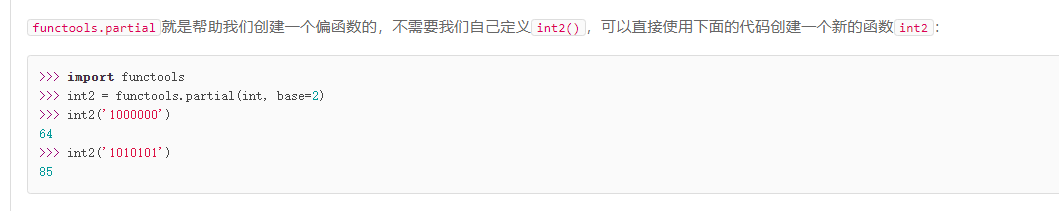


## 装饰器





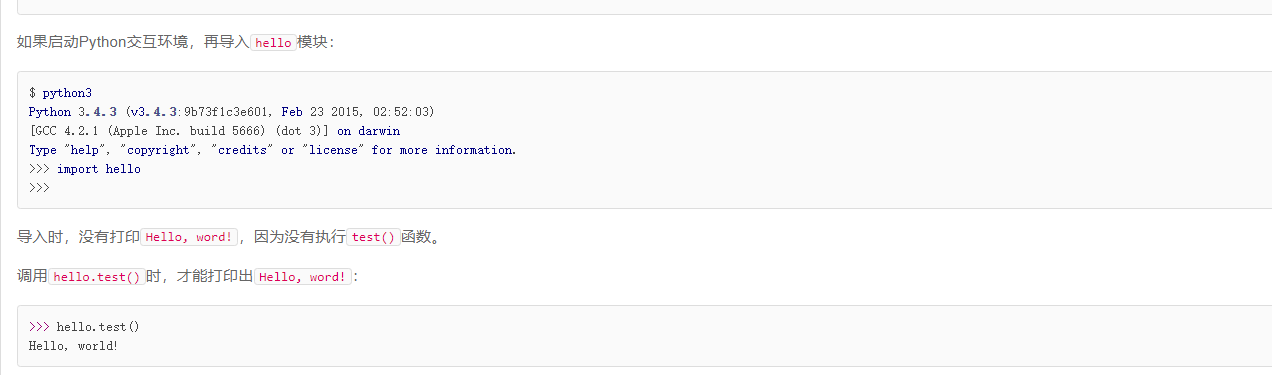
## 偏函数



# 模块

## 使用模块





## 使用第三方的模块



# 面向对象编程

## 类和实例



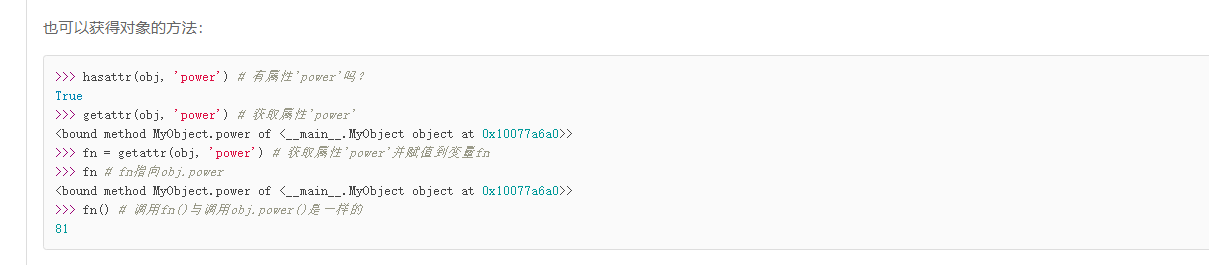
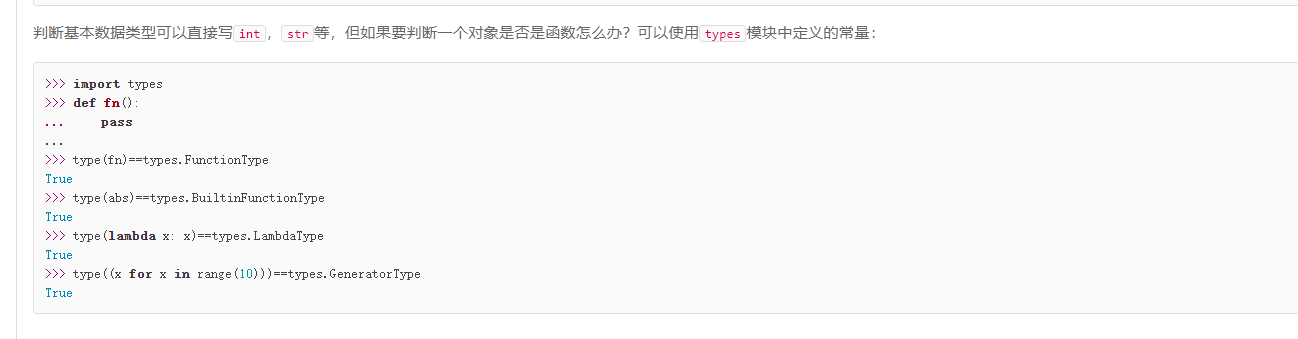
## 访问限制



## 继承和多态



## 获取对象信息

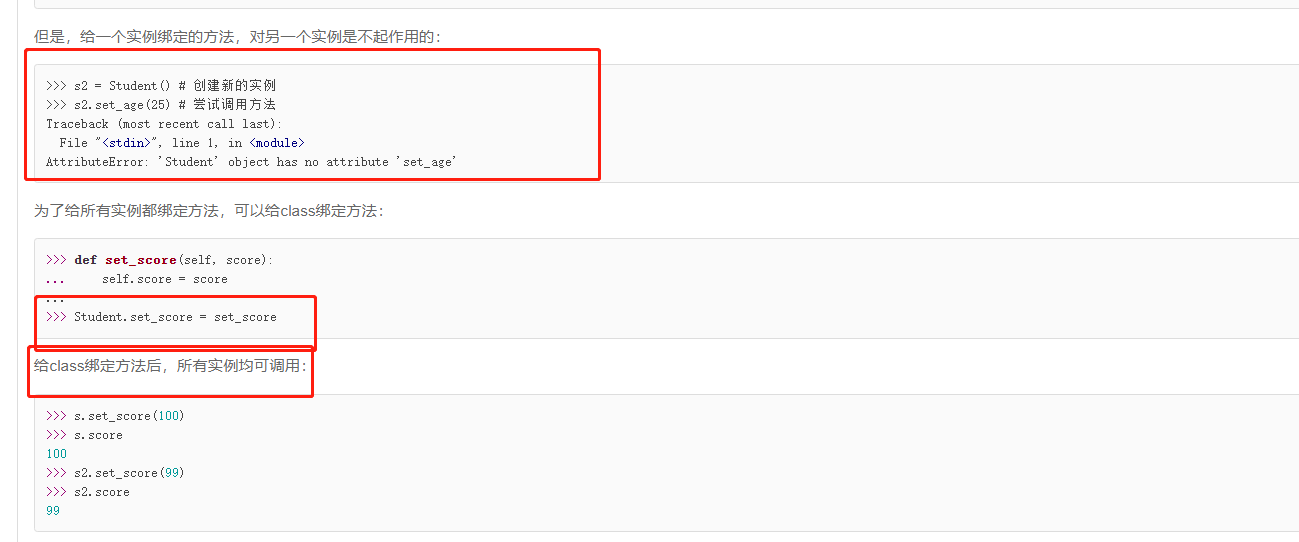
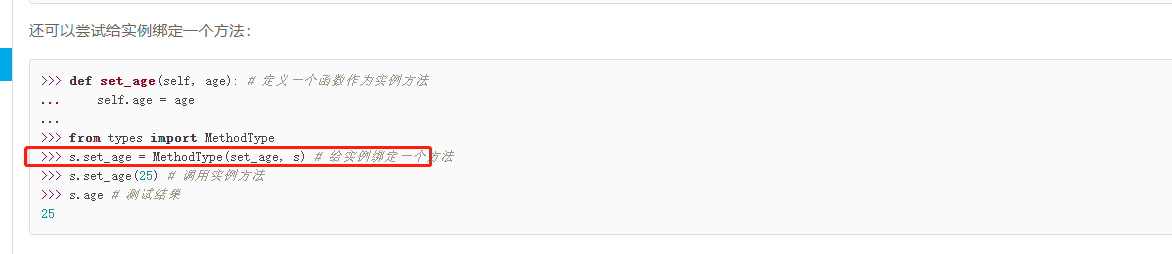


## 实例属性类属性



# 面向对象高级编程

## 使用\_\_slots\_\_



## 使用@property



## 多重继承

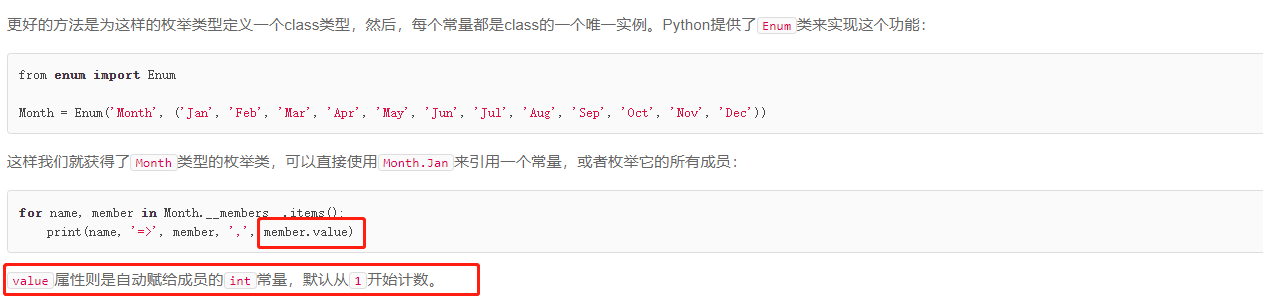


## 定制类

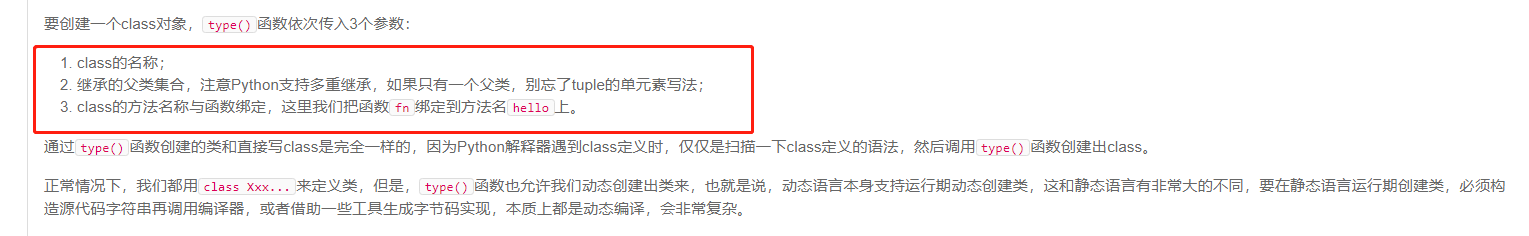




## 使用枚举类



## 使用元类



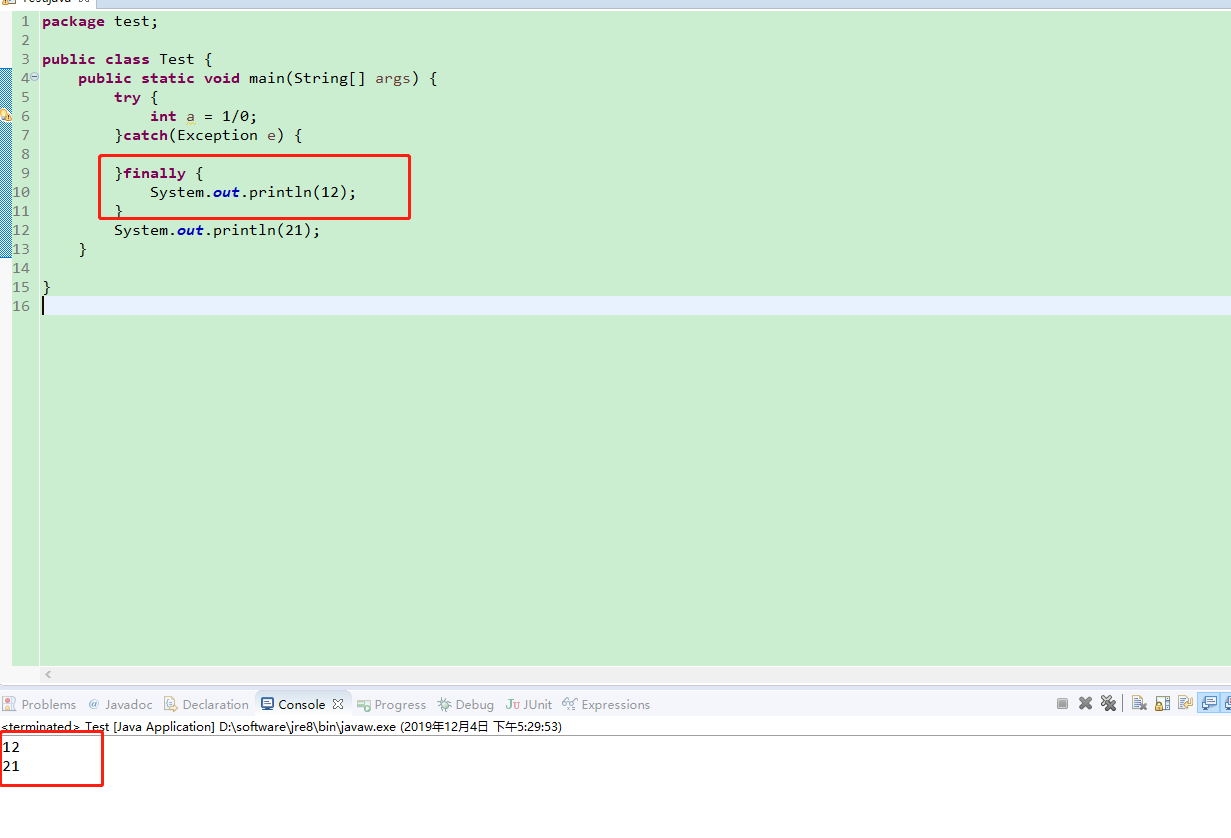
# 错误调试和测试

## 错误处理



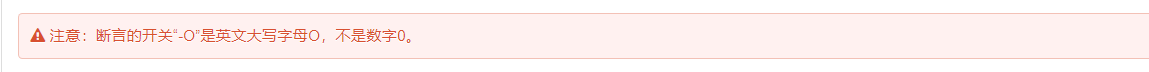


java也是如此,只是针对try起来的代码





## 调试



# io编程

## 文件的读写



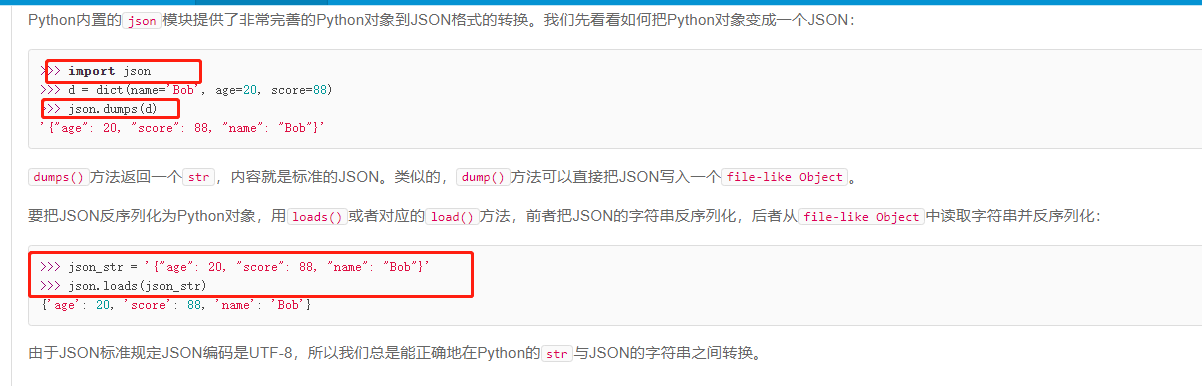
## stringIo和BytesIo



## 操作文件和目录

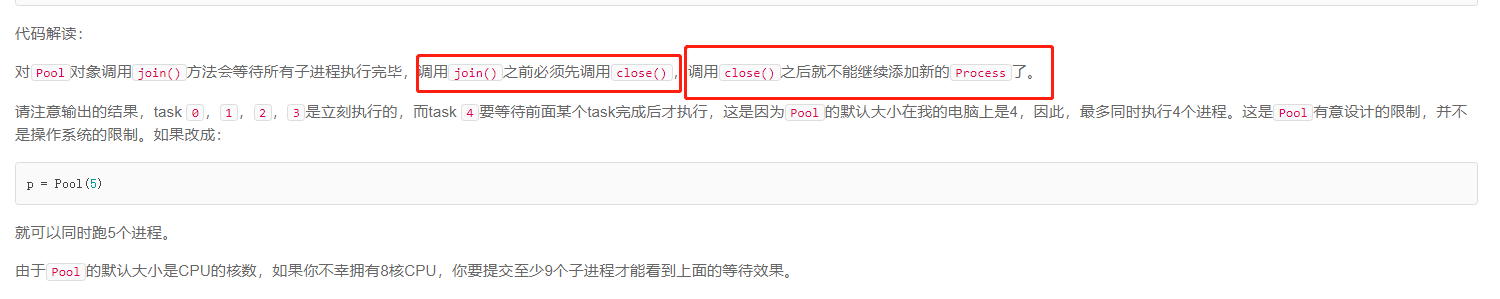
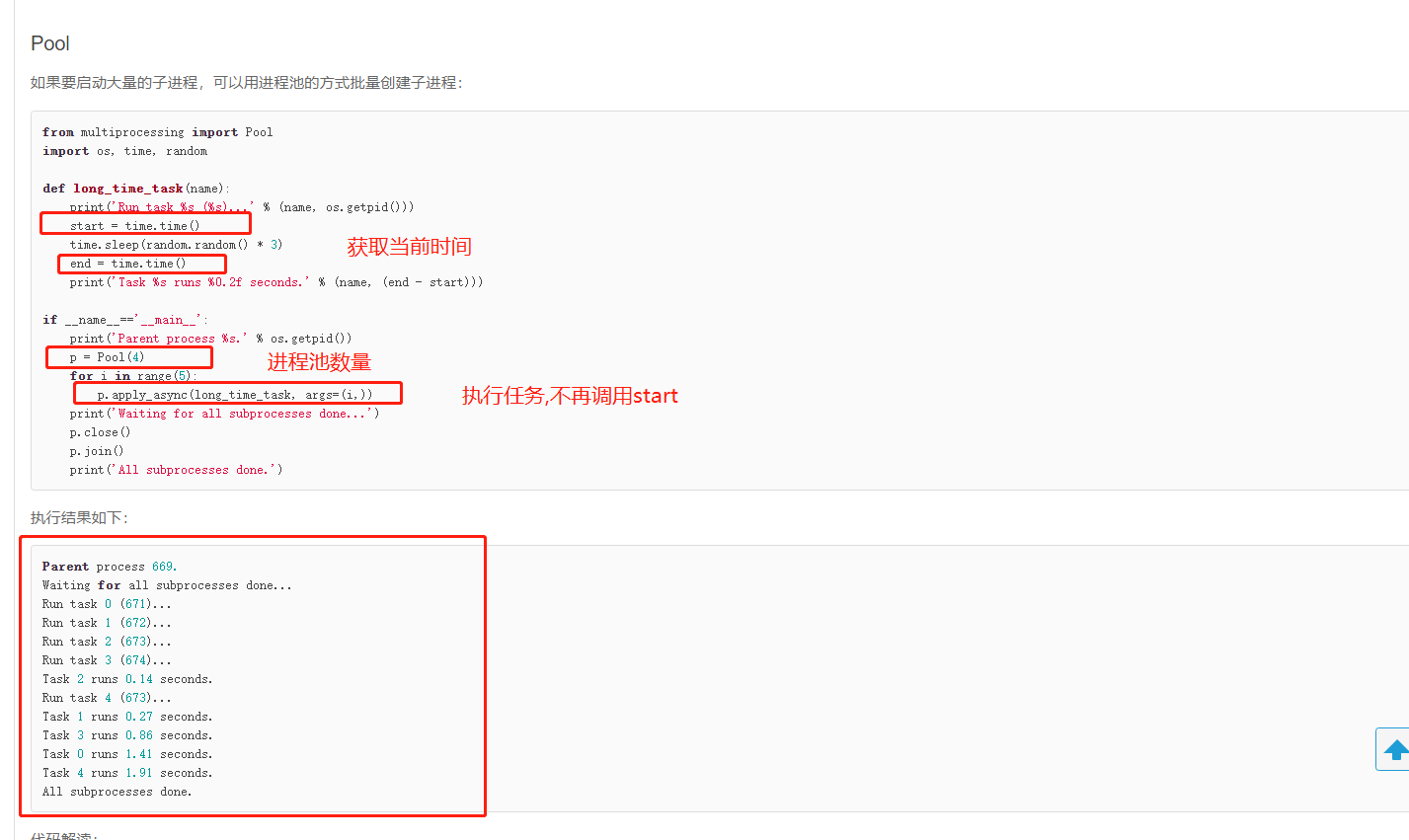


## 序列化



# 进程和线程

## 多进程



## 多线程



## ThreadLocal



## 进程 vs. 线程

## 分布式进程