

day07知识体系

迭代器

迭代概念

记录下一个元素的位置信息便于下次访问

可迭代对象

for item in iterable

from collections import Iterable

isinstance(obj, Iterable)

本质 提供迭代器

实现__iter__提供迭代器 iter(可迭代对象) 可迭代对象.__iter__()

迭代器

作用-通过next(迭代)遍历可迭代对象的数据

迭代器本身也是可迭代对象

实现__iter__ __next__方法

from collections import Iterator

isinstance(obj, Iterator)

生成器概念

生成器自带了迭代器的功能-天生就实现了迭代器的迭代操作-是一种特殊的迭代器

两种生成器

生成器表达式 (x for i in range(x))

生成器函数 带有yield关键的函数

yield关键字

挂起当前函数 将yield后面表达式的值返回到调用生成器对象的地方

当再次调用的时候 会紧接上次运行的地方继续运行(可以接收额外的数据赋值给yield左边可能存在的变量)

使用next() 和 send() 唤醒操作

g.send(None) == next(g)

第一次在执行生成器对象的时候 必须使用next(g)

多任务-协程

切换

yield实现

greenbelt实现协程

gevent实现协程-重点

创建并且执行 协程 gevent.spawn

阻塞等待协程运行完成 gevent.join()

阻塞等待所有协程退出
gevent.joinall([])

monkey,patch_all()作用就是将recv recvfrom time.sleep进行破解
在调用的时候可以切换到别的任务继续执行

注意: join join all的主要作用是保持主进程存活