# 知识点：

一般启动异步编程的方式，定义一个async main()函数，里面写入await asyncio.gather(task1(), task2()),由main去调用需要异步的函数

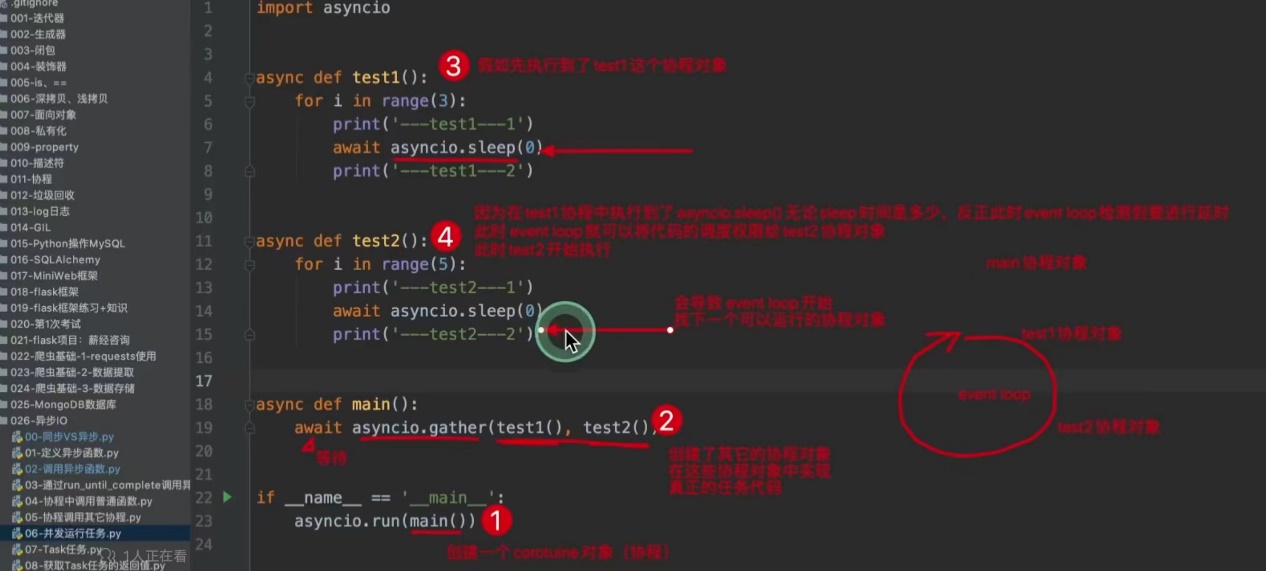
async 关键字 定义的函数，在调用时，其实是生成了一个协程对象（比如def定义的同步函数里用yield，是生成器的写法，调用时不一定就是我们之前理解的调用而是返回一个协程对象）

asyncio.run(main())

asyncio.gather()

asyncio.sleep()

await关键字 是等待这行执行完成再往下走



# Task类

学了可以让协程对象更多功能（比如，判断是否执行完成，取消协程，判断是否被取消得到）

task = asyncio.create\_task() 将协程对象封装

task.done()

asyncio.wait(list) 等待且运行列表中的任务执行完成，参数添加由多个封装后的协程对象组成的列表

task.result() 获取task封装后的协程对象返回(return)的值(可以是元组列表字典)

# 回调

理解回调：就是提前告诉回调函数做法，还没发生呢，如果以后发生了什么，按我提前说好的做（还没干呢，我先给你说好接下来怎么干哈，要是遇到什么问题你先干什么去）（还没发生什么，我先告诉你，等到接下来那件事产生了，你就去干，没产生就不干）

task.add\_done\_callback(一个同步的函数名) 当task对应得异步函数完成后去回调一个函数，并且将自己协程对象当作参数传入被回调的函数

# 给回调函数传参数

Import functools

task.add\_done\_callback(Functools.partial(一个同步的函数名, num=, num=)