



章节



问答



课签



笔记



资料

10-3 异步总结

异步相对同步而言，同步意味着有序，异步意味着无序，正因为异步的无序，使得各个程序间的协调成为一大难题，异步编程就是解决这一难题的编程，它是以进程、线程、协程、函数/方法作为执行任务程序的基本单位，结合回调、事件循环、信号量等机制，以提高程序整体执行效率和并发能力的编程方式

Python中如何实现异步

方法一：

- 如何定义一个异步，使用关键字： `async`
- 在一个异步程序里，如何调用另一个异步，使用关键字： `await`
- 如何调用一个异步函数，使用模块： `asyncio`

如图：

```
import asyncio

async def a():
    print("我是异步代码a")

async def b():
    print("我是异步代码b")

async def handle():
    await b()

async def main():
    await asyncio.gather(a(),handle()) 将异步代码加入其中批量执行

if __name__ == "__main__":
    asyncio.run(main()) 使用run方法执行异步函数
```

file_exe (1) ×

"D:\Program Files\Python38\python.exe" D:/my/code/file_exe.py

我是异步代码a

我是异步代码b

方法二：通过异步包gevent

- 如何创建协程对象： `spawn` 函数
- 如何批量处理协程对象： `joinall`

如下图：