

一、asyncio

下面通过举例来对比同步代码和异步代码编写方面的差异，其次看下两者性能上的差距，我们使用sleep(1)模拟耗时1秒的io操作。

同步代码：

```
import time def hello():
    time.sleep(1) def run(): for i in range(5):
        hello() print('Hello World:%s' % time.time()) # #####Hello World ##### if __name__ == '__main__':
    run()
```

输出：（间隔差不多是1s）

Hello World:1527595175.4728756 Hello World:1527595176.473001 Hello World:1527595177.473494 Hello World:1527595178.4739306 Hello World:1527595179.474482

异步代码：

```
import time import asyncio # ##### async def hello():
    asyncio.sleep(1) print('Hello World:%s' % time.time()) def run(): for i in range(5):
    loop.run_until_complete(hello())
```

```
loop = asyncio.get_event_loop() if __name__ == '__main__':
    run()
```

输出：

Hello World:1527595104.8338501 Hello World:1527595104.8338501 Hello World:1527595104.8338501 Hello World:1527595104.8338501 Hello World:1527595104.8338501

async def

用来定义异步函数，其内部有异步操作。每个线程有一个事件循环，主线程调用asyncio.get_event_loop()时会创建事件循环，你需要把异步的任务丢给这个循环的run_until_complete()

二、aiohttp

如果需要并发http请求怎么办呢，通常是用requests，但requests是同步的库，如果想异步的话需要引入aiohttp。这里引入一个类，from aiohttp import ClientSession，首先要建立一个session对象，然后用session对象去打开网页。session可以进行多项操作，比如post, get, put, head等。

基本用法：

```
async with ClientSession() as session:
    async with session.get(url) as response:
```

aiohttp异步实现的例子：

```
import asyncio from aiohttp import ClientSession

tasks = []
url = "https://www.baidu.com/{}" async def hello(url): async with ClientSession() as session:
    async with session.get(url) as response:
        response = await response.read() print(response) if __name__ == '__main__':
    loop = asyncio.get_event_loop()
    loop.run_until_complete(hello(url))
```

首先async def 关键字定义了这个是异步函数，await

关键字加在需要等待的操作前面，response.read()等待request响应，是个耗IO操作。然后使用ClientSession类发起http请求。

多链接异步访问

如果我们需要请求多个URL该怎么办呢，同步的做法访问多个URL只需要加个for循环就可以了。但异步的实现方式并没那么容易，在之前的基础上需要将hello()包装在asyncio.gather()中。

```
import time import asyncio from aiohttp import ClientSession

tasks = []
url = "https://www.baidu.com/{}" async def hello(url): async with ClientSession() as session:
    async with session.get(url) as response:
        response = await response.read() # print(response) print('Hello World:%s' % time.time()) def run(): for i in range(5):
    tasks.append(hello(url))
loop = asyncio.get_event_loop()
loop.run_until_complete(loop.run_until_complete(hello(url)))
```




[ftkahzmodan](#) 2018-06-13 16:05:24

代码块这个排版看的很难受
能缓解一下吗。。。

0 回复Ta



[wonderkun](#) 2018-06-26 00:23:29

编码看的强迫症都犯了

0 回复Ta



[北风飘然](#) 2018-09-04 17:03:37

你限制并发数500-600 和标题百万并发 开玩笑么.....

0 回复Ta

[登录](#) 后跟帖

先知社区

[现在登录](#)

热门节点

[技术文章](#)

[社区小黑板](#)

目录

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)