线程池和进程池

当我们对某些网站内容进行抓取的时候非常容易遇到这样一种情况.

看这个网站, 我们发现这网站的数据太多了. 有一万多页. 也就对应着一万多个url. 那我们设计多线程的时候如果每个url对应一个线程就会产生新问题. 朋友, 你一定要知道. 创建线程本身也是要消耗你的计算机资源的. 线程不是变魔术变出来的. 那这时我们就可以考虑能不能重复的使用线程呢? 答案当然可以. 线程池就可以帮你搞定.

线程池工作原理:

创建一个大池子, 存放固定数量的线程. 然后把我们要执行的任务丢给线程池. 由线程池去分配哪个线程来完成该任务. 其他的事情都不需要你来管. 舒服吧.

废话不多说, 上代码

```
from concurrent.futures import
ThreadPoolExecutor, ProcessPoolExecutor

# 线程池
def fn(name):
    for i in range(1000):
        print(name, i)


if __name__ == '__main__':
    with ThreadPoolExecutor(10) as t:
        for i in range(100):
            t.submit(fn, name=f"线程{i}")

```

至于进程池. 就把ThreadPoolExecutor更换为ProcessPoolExecutor就可以了. 其他一模一样