链滴

前端异步是什么？哪些情况下会发生异步？

作者：zhujie
原文链接：https://ld246.com/article/1588408217055
来源网站：链滴

前端异步是什么？哪些情况下会发生异步？

异步是什么？

这里就不拿官方的解释来解答了，只以个人理解来回答问题，轻喷。我们知道JavaScript是单线程的不像java等语言是多线程的，所以一般情况下，js代码是一行一行的执行的。但是某些时候需要用异来提升性能，比如说一个网络请求需要服务端返回数据30s，js不可能一直等待服务器返回再执行其他码，这时候js就会跳过这个过程，继续往下执行，直到没有代码要执行了，这时候后端返回数据了，j再接着执行返回数据之后的代码。这个过程就是一个异步。

例子

再举个栗子，你跟你女朋友不在一个班，但是你去找她的时候，发现她不在座位上，这时候你有2种择，1. 在座位上等她，然后啥都不干。2. 你回到自己的座位上，继续抄作业，给她发消息等她回来了让她告诉你，你再去找她。第一种就是同步，第二种就是异步，实际上就是js的**事件委托**。很显然第种方式效率更高。

哪些情况下会发生异步

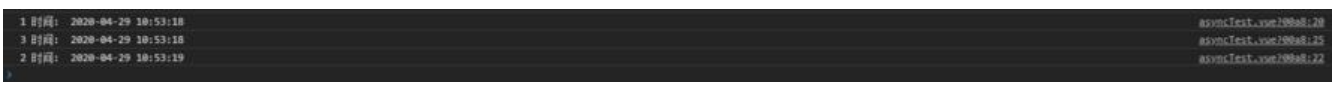
个人总结了一下，总共有下列情况会发生异步

1. 回调函数，这个很常见，很多内置函数都支持接收回调函数来异步代码
2. 事件监听，很多dom操作，　click事件等都是异步的
3. 订阅与发布，这个常见是在　angular和vue中，用on来监听事件，emit来发布事件，长用与父子件交互
4. promise是es6新增的特性，能通过resolve和reject来执行异步操作，长与async await配合使用。

具体代码示例

```
asyncFun() {
    return new Promise((resolve) => {
        console.log('1', '时间：', moment().format('YYYY-MM-DD hh:mm:ss'))
        setTimeout(() => {
            console.log('2', '时间：', moment().format('YYYY-MM-DD hh:mm:ss'))
            resolve()
        }, 1000)
        console.log('3', '时间：', moment().format('YYYY-MM-DD hh:mm:ss'))
    })
},
async run() {
    await this.asyncFun()
}
```

运行截图：



原文链接：前端异步是什么？哪些情况下会发生异步？

可以看到，代码采用setTimeout的回调函数来执行一个异步，采用promise配合async await来完成个过程，代码的执行循序就是1,3然后过1s输出2.后面会有详细的promise async await的讲解。

线上体验地址

所有的源码都可以在我的仓库地址：下载

个人公众号：

原文链接：前端异步是什么？哪些情况下会发生异步？