你真会用setTimeout吗?

javascript

YaoTang 2015年11月11日发布

教科书里面的setTimeout

定义很简单

setTimeout() 方法用于在指定的毫秒数后调用函数或计算表达式。

广泛应用场景

定时器，轮播图，动画效果，自动滚动等等

上面一些应该是setTimeout在大家心中的样子，因为我们平常使用也不是很多。

但是setTimeout真的有那么简单吗？

测试题

一个题目，如果你在一段代码中发现下面内容

```
var startTime = new Date();
setTimeout(function () {
    console.log(new Date() - startTime);
}, 100)
```

请问最后打印的是多少?

我觉得正确答案是，取决于后面同步执行的js需要占用多少时间。

`MAX(同步执行的时间, 100)`。

再加一个题目,只有下面代码

```
setTimeout(function () {
    func1();
}, 0)
func2();
```

func1和func2谁会先执行？

这个答案应该比较简单，func2先执行，func1后面执行。

再来一题

```
setTimeout(function () {
    func1（）
}, 0)
```

和

```
setTimeout(function () {
    func1（）
})
```

有什么差别？

> 0秒延迟，此回调将会放到一个能立即执行的时段进行触发。javascript代码大体上是自顶向下的，但中间穿插着有关DOM渲染，事件回应等异步代码，他们将组成一个队列，零秒延迟将会实现插队操作。不写第二个参数，浏览器自动配置时间，在IE，FireFox中，第一次配可能给个很大的数字，100ms上下，往后会缩小到最小时间间隔，Safari，chrome，opera则多为10ms上下。

上面答案来自《javascript框架设计》

好了，看了上面几个题目是不是感觉setTimeout不是想象中那样了。

setTimeout和单线程

下面是我自己的一些理解 首先需要注意javascript是单线程的，特点就是容易出现阻塞。如果一段程序处理时间很长，很容易导致整个页面hold住。什么交互都处理不了怎么办？

简化复杂度？复杂逻辑后端处理？html5的多线程？

上面都是ok的做法，但是setTimeout也是处理这种问题的一把好手。

setTimeout一个很关键的用法就是分片，如果一段程序过大，我们可以拆分成若干细小的块。例如上面的情况，我们将那一段复杂的逻辑拆分处理，分片塞入队列。这样即使在复杂程序没有处理完时，我们操作页面，也是能得到即使响应的。其实就是将交互插入到了复杂程序中执行。

换一种思路，上面就是利用setTimeout实现一种伪多线程的概念。

有个函数库 `Concurrent.Thread.js` 就是实现js的多线程的。

一个简单使用的例子，引入 `Concurrent.Thread.js` 后

```
Concurrent.Thread.create(function(){
    for (var i = 0;i<1000000;i++) {
        console.log(i);
    };
});
$('#test').click(function  () {
    alert(1);
});
```

虽然有个巨大的循环，但是这时不妨碍你去触发alert();

是不是很厉害~

还有一种场景，当我们需要渲染一个很复杂的DOM时，例如table组件，复杂的构图等等，假如整个过程需要3s,我们是等待完全处理完成在呈现，还是使用一个setTimeout分片,将内容一片一片的断续呈现。

其实setTimeout给了我们很多优化交互的空间。

如何使用

setTimeout这么厉害，那么我们是需要在在项目中大量使用吗？
我这边的观点是非常不建议，在我们业务中，基本上是禁止在业务逻辑中使用setTimeout的，因为我所看到的很多使用方式都是一些问题不好解决，setTimeout作为一个hack的方式。

例如，当一个实例还没有初始化的前，我们就使用这个实例，错误的解决办法是使用实例时加个setTimeout，确保实例先初始化。
为什么错误？这里其实就是使用hack的手段
第一是埋下了坑，打乱模块的生命周期
第二是出现问题时，setTimeout其实是很难调试的。

我认为正确的使用方式是，看看生命周期（可参考《关于软件的生命周期》），把实例化提到使用前执行。

综上，setTimeout其实想用好还是很困难的， 他更多的出现是在框架和类库中，例如一些实现Promis的框架，就用上了setTimeout去实现异步。 所以假如你想去阅读一些源码，想去造一些轮子，setTimeout还是必不可少的工具。

微信公众号

博客地址

http://tangguangyao.github.io/

2015年11月11日发布 更多

2 推荐 收藏

你可能感兴趣的文章

JavaScript：setTimeout 和 setInterval 458 浏览

一个经典的javascript面试题的新探索 9 收藏，521 浏览

JavaScript的setTimeout和setInterval的深入理解 16 收藏，1.5k 浏览

1 条评论 默认排序

WoodenSail · 2015年11月11日

我在angular里用过setTimeOut做hack。需要当一个被ng-if控制的input出现时自动聚焦。然而angular只能让我监听ng-if的条件，没法监听input渲染完毕。所以只好hack了。

赞 回复

文明社会，理性评论

发布评论

YaoTang
715 声望

关注作者

发布于专栏

前端修炼

前端知识，技术分享，侧重分享一些商业系统级别的前端技术。

23 人关注

关注专栏

相关收藏夹

换一组

webpack
4 个条目 | 0 人关注

Vue demo
6 个条目 | 1 人关注

我的收藏
5 个条目 | 0 人关注

分享扩散：