# 定时器

## 方法（windows）

setTimeout

启用一个定时器，在一段**延迟**后执行回调。

clearTimeout

定时器如果没有触发，清除定时器

setInterval

启动一个定时器，没隔一段时间执行回调。

clearInterval

清除定时器

## 区别

setTimeout（fn , n）表示：延迟n后执行fn

setInterval(fn,n) 表示：每隔n尝试执行fn

如有fn执行时间大于n，异步执行需要排队，（如果一直延迟，甚至可能导致fn最后执行不需要延迟）

## 调整事件队列

setTimeout(fn,0)

将fn置于所有事件（event loop）之后----（包括渲染）的后面）

## 处理昂贵的计算过程

/pages/feature/timeout/timeout.html

## 中央定时器控制

/pages/feature/timeout/central-timer-control.html

对于大量定时器，浏览器将会增加垃圾回收任务发生的可能性。

定时器是个特殊的问题，因为它们通常是JavaScript单线程引擎之外的进程中进行管理的（其他浏览器线程）

1. 每个页面在同一时间只需要运行一个定时器。
2. 可以根据需要暂停和恢复定时器
3. 删除回调函数的过程变得简单

# 线程

在 web Worker之前，所有JavaScript代码都在单线程中执行。

调用