计时器定义：JavaScript一个设定的时间间隔之后来执行代码，称之为计时事件。

setTimeout（），setInterval（）都是window对象的方法。

setTimeout（） 延迟定时器 延迟多长时间之后 执行该函数 只执行一次

第一个参数：需要执行功能（函数）

第二参数 时间 单位是毫秒

setInterval（） 循环定时器 每隔一段时间之后 执行该函数 重复循环执行

第一个参数：需要执行功能（函数）

第二参数 时间 单位是毫秒

定时器如何清除：setTimeout（）的清除方法 clearTimeout();

定时器如何清除：setInterval（）的清除方法 clearInterval();

关于延迟时间：由于javascript 的事件循环机制，导致第二个参数并不代表延迟delay毫秒之后立即执行回调函数，而是尝试将回调函数加入到事件队列。

* 区别：setTimeout：延时delay毫秒之后，啥也不管，直接将回调函 数加入事件队列。
* setInterval: 延时delay毫秒之后，先看看事件队列中是否存在还没有执行的回调函数（setInterval的回调函数），如果存在，就不要再往事件队列里加入回调函数了。

浏览器的内核是多线程的，它们在内核制控下相互配合以保持同步，一个浏览器至少实现3个常驻线程：javascript引擎线程，GUI渲染线程，浏览器事件触发线程。

JavaScript引擎：是基于事件驱动单线程执行的，JS引擎一直等待着任务队列中任务的到来然后加以处理，浏览器无论再什么时候都只有一个JS线程在运行JS程序。  
 GUI渲染线程：负责渲染浏览器界面，当界面需要重绘(Repaint)或由于某种操作引发回流(reflow)时,该线程就会执行。但需要注意 GUI渲染线程与JS引擎是互斥的，当JS引擎执行时GUI线程会被挂起，GUI更新会被保存在一个队列中等到JS引擎空闲时立即被执行。  
 浏览器事件触发线程：当一个事件被触发时该线程会把事件添加到待处理队列的队尾，等待JS引擎的处理。这些事件可来自JavaScript引擎当前执行的代码块如setTimeOut、也可来自浏览器内核的其他线程如鼠标点击、AJAX异步请求等，但由于JS的单线程关系所有这些事件都得排队等待JS引擎处理。