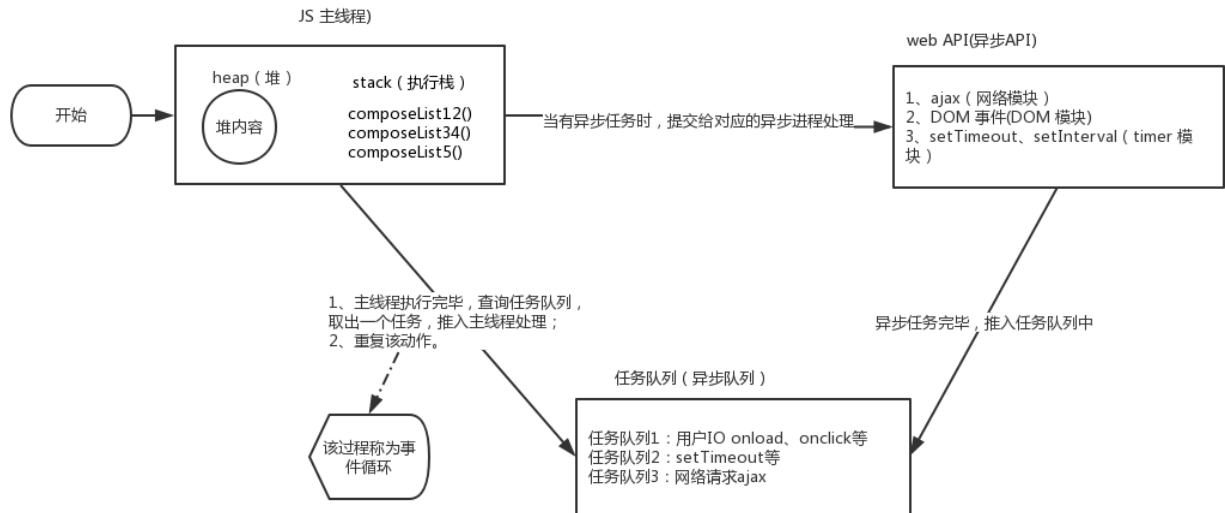


虽然负责解释和执行js代码的线程只有1个，但是浏览器内部还有其他线程来专门负责异步任务的，比如定时器，UI，事件，网络等专门线程来负责相关任何的处理。



1. step1：主线程读取JS代码，此时为同步环境，形成相应的堆和执行栈；
2. step2：主线程遇到异步任务，指给对应的异步进程进行处理（webAPI）；
3. step3：异步进程处理完毕（Ajax返回，DOM事件处理，Timer到时等），将相应的异步任务推入任务队列；
4. step4：主线程执行完当前代码时，都会检查异步任务队列中是否有任务需要执行，如果有，则取出一个任务推入主线程处理（先进先出）；
5. step5：重复执行step2、3、4；称为事件循环。（如何处理异步队列中的代码碰到了异步代码，那么再次交给浏览器中的其他线程，然后跳过这些执行，直到当前代码段结束，然后检查异步队列，如此循环往复。。。所以叫事件循环。）

执行的大意：

同步环境执行（step1）-> 事件循环1（step4）-> 事件循环2（step4）...

