Js异步编程：因为js为单线程模式，容易造成代码阻塞，而js异步编程相当于是js的另一条线程，如果代码中有操作耗时比较久，那我们可以在将它编写到异步代码中，这样当浏览器解析到它时不会因为它的操作未完成而阻塞其他代码执行。

EventLoop：事件循环，当js解析到异步任务时并不会等它解析完成，当异步返回结果时而将异步放入事件队列，当主线程执行完毕后，会从事件队列中拿取任务执行，如任务中仍有返回结果，仍会将结果放入事件队列中等待执行，形成事件循环。

消息队列：虽然异步编程相当于是js的另一条线程，但是js解析时的线程仍然是单线程，所有任务包括异步任务，都会到消息队列中排队执行。

宏任务:计时器代码执行的任务为宏任务，当微任务队列任务执行完毕时，才会从宏任务队列中提取代码执行。

微任务：promise执行的代码为微任务，当主线程任务执行完毕会先从微任务队列中提取任务执行。