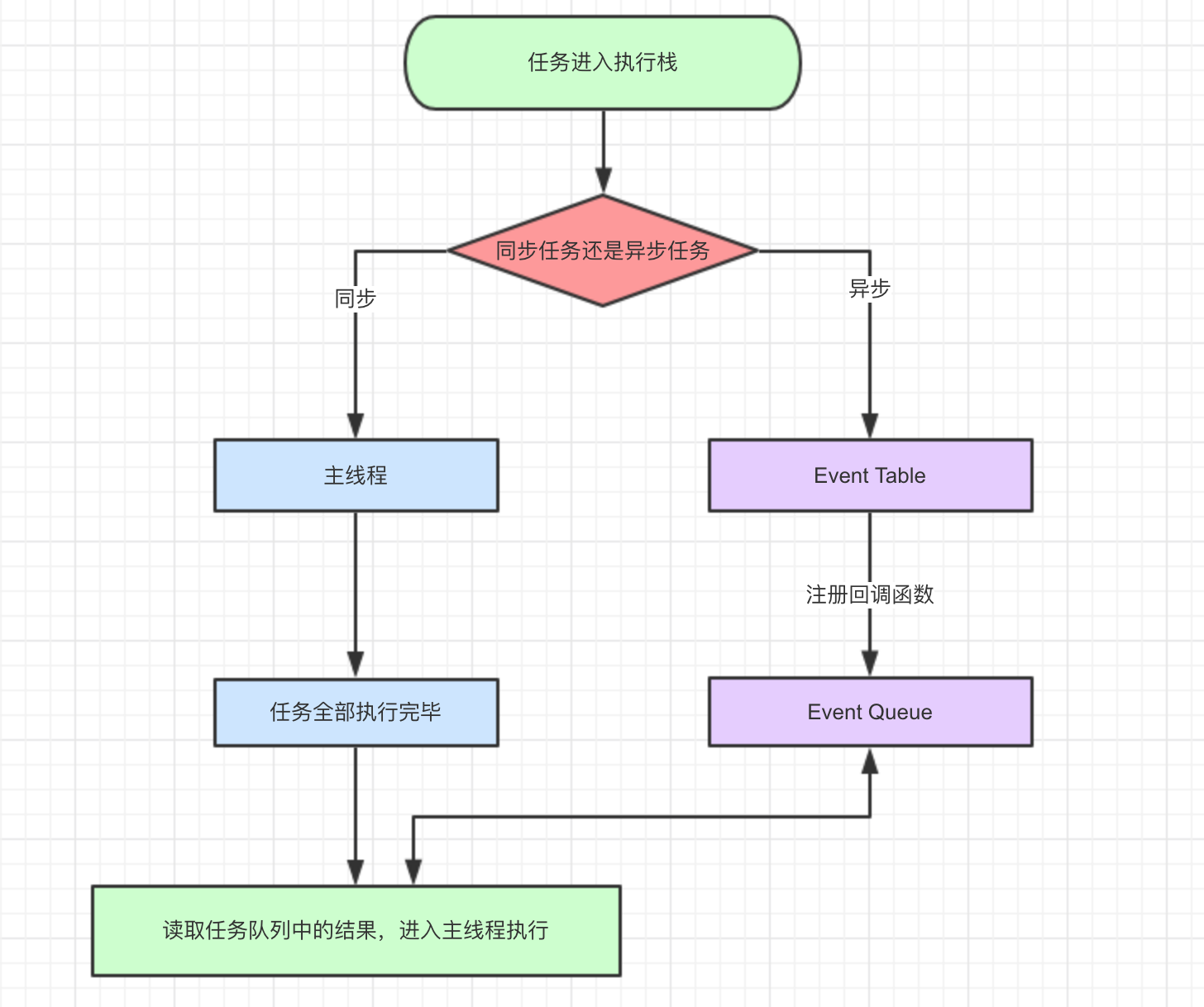
# 关于javascript事件循环

javascript是一门单线程语言，js任务是一个一个顺序执行，分为同步任务和异步任务，同步任务会进入主线程，异步任务会进入Event Table（事件表），当时间表中的异步任务完成后就会在Event Queue（事件队列）中注册回调函数，主线程任务全部完成后，才会执行Event Queue中的回调，js解析器会不断重复检查主线程执行栈是否为空，然后重复第3步，这就是Event Loop（事件循环）

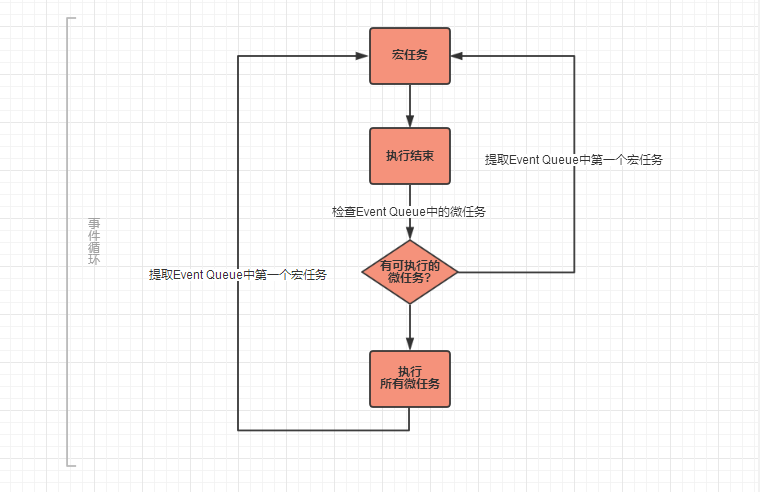
任务又可以进一步分为宏任务和微任务

宏任务：setTimeout setInterval

微任务：promise process.nextTick



在事件循环（主线程——>事件队列）中的操作流程，即（宏任务——>微任务）



通常情况下，页面中js执行顺序：

第一轮事件循环：

1. 主线程执行js整段代码，将ajax，setTimeout等回调函数注册到事件队列，并区分宏任务和微任务
2. 主线程提取并执行事件队列中的所有微任务，并注册微任务中的异步任务到事件队列中

第二轮事件轮询：

1. 主线程提取事件队列中的第一个宏任务
2. 主线程执行第一个宏任务，并注册这个宏任务中的异步任务到事件队列
3. 执行事件队列中的所有微任务，并将微任务中的异步任务到事件队列中

宏任务每执行完一个，就清空一次事件队列中的微任务

事件队列中分‘宏任务队列’和‘微任务队列’，每执行一次任务都可能注册新的宏任务或微任务到响应的任务队列中