

浏览器和 nodejs 事件循环有什么区别

JS 是单线程，需要异步执行。先执行同步代码，同步代码执行完了，就执行异步代码。异步代码中，分为微任务和宏任务，先执行微任务、再执行宏任务。宏任务中又有很多任务，会先放入队列，等到该执行宏任务时，按照先排队，先执行的顺序依次执行。

1. 微任务和宏任务

微任务和宏任务是异步任务的两种类型。

- 宏任务，如 `setTimeout`, `setInterval`, 网络请求。 `requestAnimationFrame`, `requestIdleCallback`。
- 微任务，如 `promise`, `async/await`, `MutationObserver`（监听 DOM 变化的任务），`process.nextTick`（在 Nodejs 环境下，在当前执行栈结束后立即执行的任务）。

微任务在下一轮 DOM 渲染之前执行，宏任务在之后执行。也就是说执行顺序是“微任务->DOM渲染->宏任务”。

event loop（事件轮训/事件循环）会一遍一遍的监听微任务队列（micro-task-queue）和宏任务队列（macro-task-queue）中是否有新的待执行任务。

2. 区别

nodejs 的微任务和宏任务，划分优先级和类型的。`process.nextTick` 优先级最高。

即

1. 最先执行同步代码。
2. 执行微任务（`process.nextTick` 优先级最高）
3. 按顺序执行 6 个类型的宏任务。

总结

- 浏览器和 nodejs 的 `event loop`（事件循环）流程基本相同。
- nodejs 宏任务和微任务分类型和优先级。