

防抖和节流是针对响应跟不上触发频率这类问题的两种解决方案。在给DOM绑定事件时，有些事件我们是无法控制触发频率的。如鼠标移动事件onmousemove，滚动滚动条事件onscroll，窗口大小改变事件onresize，瞬间的操作都会导致这些事件会被高频触发。如果事件的回调函数较为复杂，就会导致响应跟不上触发，出现页面卡顿，假死现象。在实时检查输入时，如果我们绑定onkeyup事件发请求去服务端检查，用户输入过程中，事件的触发频率也会很高，会导致大量的请求发出，响应速度会大大跟不上触发。针对此类快速连续触发和不可控的高频触发问题，debounce 和 throttling 给出了两种解决策略：

debounce，去抖动。策略是当事件被触发时，设定一个周期延迟执行动作，若期间又被触发，则重新设定周期，直到周期结束，执行动作。这是debounce的基本思想，在后期又扩展了前缘debounce，即执行动作在前，然后设定周期，周期内有事件被触发，不执行动作，且周期重新设定。

throttling，节流的策略是，固定周期内，只执行一次动作，若有新事件触发，不执行。周期结束后，又有事件触发，开始新的周期。

作者：起晚的蜗牛

来源：CSDN

原文：<https://blog.csdn.net/hupian1989/article/details/80920324>

版权声明：本文为博主原创文章，转载请附上博文链接！