

题目：

## js防抖和节流剖析

题目标签

学习时长：20分钟

题目难度：中等

知识点标签：前端、JavaScript

相关分析

### 防抖 (debounce)

防抖，顾名思义，防止抖动，以免把一次事件误认为多次，敲键盘就是一个每天都会接触到的防抖操作。

想要了解一个概念，必先了解概念所应用的场景。在 JS 这个世界中，有哪些防抖的场景呢

1. 登录、发短信等按钮避免用户点击太快，以致于发送了多次请求，需要防抖
2. 调整浏览器窗口大小时，`resize` 次数过于频繁，造成计算过多，此时需要一次到位，就用到了防抖
3. 文本编辑器实时保存，当无任何更改操作一秒后进行保存

代码

```
function debounce (f, wait) {  
  let timer  
  return (...args) => {  
    clearTimeout(timer)  
    timer = setTimeout(() => {  
      f(...args)  
    }, wait)  
  }  
}
```

### 节流 (throttle)

节流，顾名思义，控制水的流量。控制事件发生的频率，如控制为1s发生一次，甚至1分钟发生一次。与服务端(server)及网关(gateway)控制的限流 (Rate Limit) 类似。

1. `scroll` 事件，每隔一秒计算一次位置信息等
2. 浏览器播放事件，每个一秒计算一次进度信息等
3. `input` 框实时搜索并发送请求展示下拉列表，每隔一秒发送一次请求 (也可做防抖)

代码

```
function throttle (f, wait) {
  let timer
  return (...args) => {
    if (timer) { return }
    timer = setTimeout(() => {
      f(...args)
      timer = null
    }, wait)
  }
}
```

总结 (简要答案)

防抖：防止抖动，单位时间内事件触发会被重置，避免事件被误伤触发多次。代码实现重在清零 `clearTimeout`。防抖可以比作等电梯，只要有一个人进来，就需要再等一会儿。业务场景有避免登录按钮多次点击的重复提交。

节流：控制流量，单位时间内事件只能触发一次，与服务器端的限流 (Rate Limit) 类似。代码实现重在开锁关锁 `timer=timeout; timer=null`。节流可以比作过红绿灯，每等一个红灯时间就可以过一批。