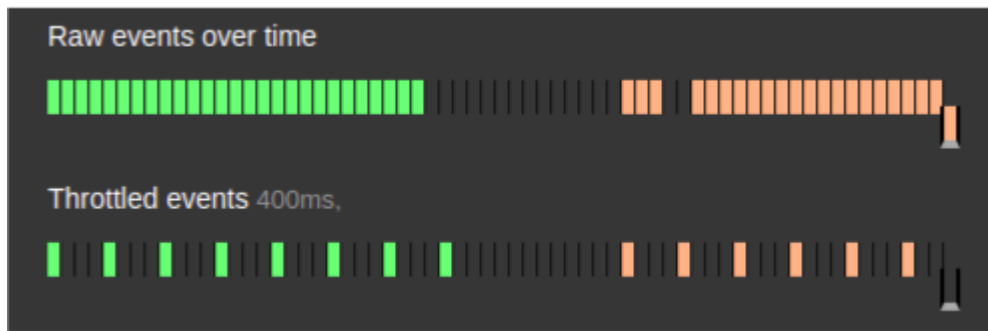


Throttle



Source: <https://css-tricks.com/debouncing-throttling-explained-examples/>

1. Throttle usages
2. Simple implementation
3. debounce and throttle

1. Throttle usages

- **throttle**: là một kỹ thuật để giảm tần suất của các hàm được gọi liên tục, giới hạn chúng chỉ được gọi một lần sau một khoảng thời gian nhất định.
- Điều này thường được sử dụng để kiểm soát tần suất của các sự kiện như **scroll**, **resize** hoặc **keystroke** trong ứng dụng web.

<https://webdesign.tutsplus.com/javascript-debounce-and-throttle--cms-36783t>

2. Simple implementation

```
function throttle(callback, wait) {
  let isThrottling = false;
  return function () {
    if (isThrottling) return; // 2, 3, 4 come here
    isThrottling = true; // 1 come here
    setTimeout(() => {
      callback();
      isThrottling = false;
    }, wait);
  };
}

const throttledLog = throttle(log, 500);
// 0 -> 500
throttledLog(); // set timeout 500 --> call --> turn of flag throttle
throttledLog(); // check throttling --> ignore
throttledLog(); // check throttling --> ignore
throttledLog(); // check throttling --> ignore
// 600 --> 1100
setTimeout(throttledLog, 600); // start throttling
setTimeout(throttledLog, 600);
setTimeout(throttledLog, 600);
```

```
setTimeout(throttledLog, 600);
setTimeout(throttledLog, 700);
setTimeout(throttledLog, 700);
```

3. debounce and throttle

#	debounce	throttle
concept	không giới hạn số lần gọi lại hàm, mà chỉ xác định thời gian trôi qua trước khi hàm được gọi.	giới hạn tần suất gọi hàm chỉ một lần sau một khoảng thời gian nhất định.
reduce number of calls	YES	YES
common case	search when stop typing	do something on scroll