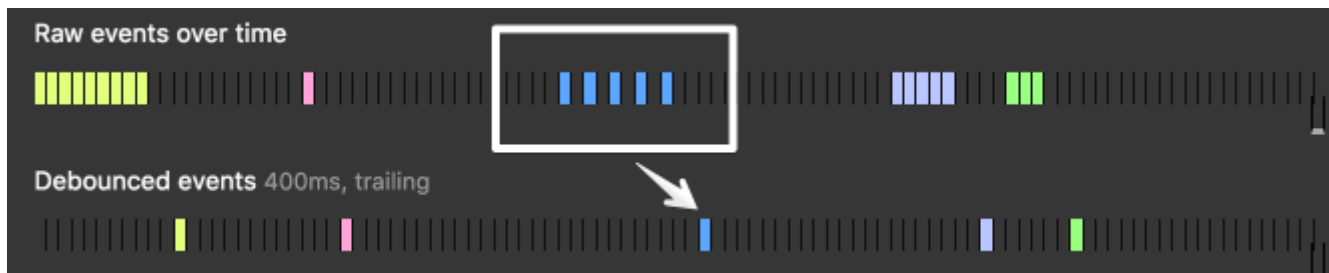


Debounce technique



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

1. Debounce usages
2. Simple implementation
3. Read more on [lodash.debounce](#)

1. Debounce usages

- **Debounce**: là một kỹ thuật được sử dụng để giảm thiểu số lượng lần một hàm được gọi khi một sự kiện xảy ra liên tục trong một khoảng thời gian ngắn.
- Kỹ thuật debounce thường được sử dụng trong các tình huống như xử lý sự kiện **scroll**, **resize** hoặc **keyup/keydown**, nơi mà có thể có nhiều sự kiện xảy ra liên tục và chúng ta chỉ quan tâm đến sự kiện sau cùng hoặc không muốn xử lý sự kiện quá nhiều lần gây tốn tài nguyên.

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

2. Simple implementation

```
function debounce(callback, wait) {  
  let timeoutId;  
  return function () {  
    if (timeoutId) {  
      clearTimeout(timeoutId);  
    }  
    timeoutId = setTimeout(callback, wait);  
  };  
}  
  
function log() {  
  console.log('tada');  
}  
  
log();  
log();  
log();  
log();  
log();  
// log tada 5 times  
const debounceLog = debounce(log, 500);  
debounceLog();
```

```
debounceLog();
debounceLog();
debounceLog();
debounceLog();
// log tada only 1 time
```

Ứng dụng thực tế:

```
<input type="text" id="textInput" placeholder="Type something..." />
<div id="output"></div>
```

```
// Hàm debounce
function debounce(func, delay) {
  let timeoutId;

  return function () {
    clearTimeout(timeoutId);

    const context = this;
    const args = arguments;

    timeoutId = setTimeout(() => {
      func.apply(context, args);
    }, delay);
  };
}

// Hàm xử lý sự kiện input
function handleInput(event) {
  const text = event.target.value;
  document.getElementById("output").innerText = text;
}

// Tạo một phiên bản debounce của hàm xử lý sự kiện input
const debouncedHandleInput = debounce(handleInput, 500);

// Gắn sự kiện input vào input element
const inputElement = document.getElementById("textInput");
inputElement.addEventListener("input", debouncedHandleInput);
```

3. Read more on [lodash.debounce](#)

cách 1: NPM

```
# npm
npm i lodash.debounce
```

```
# or yarn
yarn add lodash.debounce
```

cách 2: CDN

```
<!-- CDN -->
<script src="https://cdn.jsdelivr.net/npm/lodash@4.17.21/lodash.min.js"></script>
```

html

```
<input type="text" id="textInput" placeholder="Type something..." />
<div id="output"></div>
```

javascript

```
// import debounce from 'lodash.debounce'; // -> ES6
// const debounce = require('lodash.debounce'); -> ES5

// Hàm xử lý sự kiện input
function handleInput(event) {
  const text = event.target.value;
  document.getElementById("output").innerText = text;
}

// Tạo một phiên bản debounce của hàm xử lý sự kiện input sử dụng lodash debounce
const debouncedHandleInput = _.debounce(handleInput, 300);

// Gắn sự kiện input vào input element
const inputElement = document.getElementById("textInput");
inputElement.addEventListener("input", debouncedHandleInput);
```

Tham khảo

- <https://css-tricks.com/debouncing-throttling-explained-examples/>
- <https://davidwalsh.name/javascript-debounce-function>
- <https://kettanaito.com/blog/debounce-vs-throttle>