

Debouncing and Throttling in Javascript



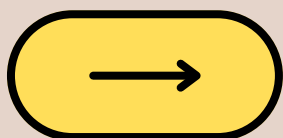
swipe for more →

①

Debouncing

Debouncing ensures that a function is only executed after a certain period of inactivity. If the function is called again before the delay period is over, the timer resets.

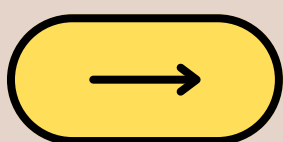
- Useful for scenarios where we want to wait until the user has stopped performing an action, such as typing in a search box or resizing a window.



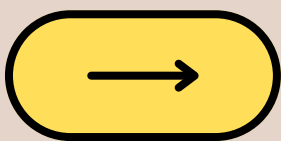
Throttling ②

Throttling ensures that a function is only executed once every specified period, regardless of how many times the event is triggered. This limits the number of times a function is called.

- Useful for scenarios where we want to limit the execution rate of an event handler, such as handling scroll events or API requests at regular intervals



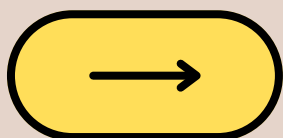
Differences Between Debouncing and Throttling



Debouncing

③

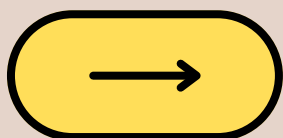
- Executes the function after a specified period of inactivity.
- Good for scenarios where we want to ensure an action is completed before the function is called.
- Function is called only once, after the user stops performing actions.
- Example: Autocomplete search input, resizing windows.



Throttling

④

- Executes the function at regular intervals, no matter how many times the event is triggered.
- Good for scenarios where we need to limit the number of times a function is called.
- Function is called at regular intervals, regardless of how often the user performs actions.
- Example: Scroll events, API rate limiting.



THANK
YOU

Follow for more



Amit Ranjan
Web Developer