

(Notes) Debounce and Throttling

Question: How does the JavaScript code implement debounce functionality for button clicks?

Explanation:

The JavaScript code uses lodash's `_.debounce` function to create a debounced version of a callback function. It tracks the number of times a button is pressed and updates the count after a specified debounce time (**800ms** in this case).

Question: How does the JavaScript code implement throttle functionality for button clicks?

Explanation:

The JavaScript code uses lodash's `_.throttle` function to create a throttled version of a callback function. It tracks the number of times a button is pressed and updates the count at intervals of **1000ms**, ensuring that the callback function is not invoked more than once in that interval.

Question: What is the purpose of the JavaScript code for a custom debounce function (polyfill)?

Explanation:

The JavaScript code defines a custom debounce function `myDebounce` that mimics the behavior of lodash's `_.debounce`. It creates a debounced version of a callback function and ensures that the callback is executed only after a specified debounce time (**800ms** in this case) since the last invocation.

Question: What is the purpose of the JavaScript code for a custom throttle function (polyfill)?

Explanation:

The JavaScript code defines a custom throttle function `myThrottle` that mimics the behavior of lodash's `_.throttle`. It creates a throttled version of a callback function and ensures that the callback is invoked at most once per specified interval (**1000ms** in this case), preventing rapid successive invocations.
