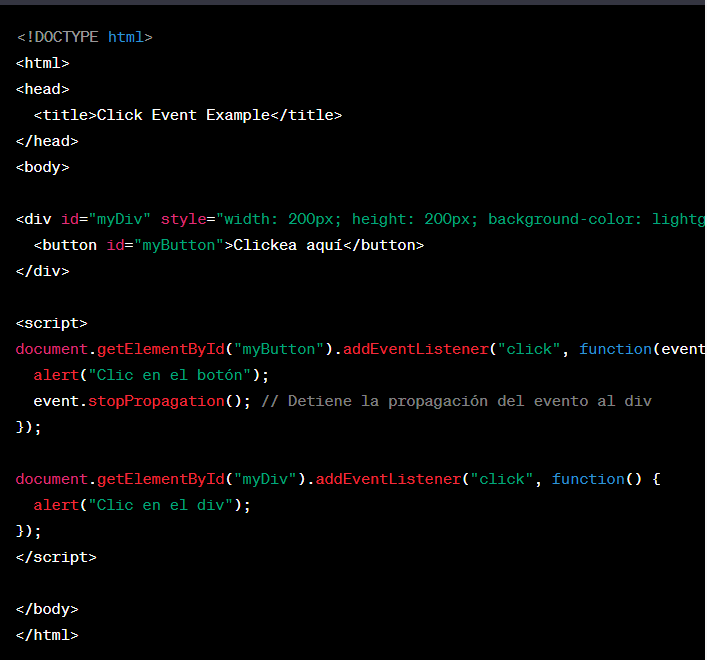
**Material útil para entender eventos:**

* <https://developer.mozilla.org/es/docs/Learn/JavaScript/Building_blocks/Events>
* <https://www.w3schools.com/jsref/dom_obj_event.asp>

**Si tengo un boton dentro de un div y quiero que me salte un alert cuando doy click en el boton y otro alert distinto cuando doy click en el div, como hago para que al hacer click en el boton no me salten los dos carteles (porque el boton esta dentro del div)?**

Para lograr esto, necesitas detener la propagación del evento desde el botón hasta el div cuando haces clic en el botón. Puedes hacer esto utilizando el método **stopPropagation()** del objeto de evento. Aquí tienes un ejemplo de cómo hacerlo:



En este ejemplo, cuando haces clic en el botón, se mostrará un alerta que dice "Clic en el botón". Además, dado que hemos utilizado **event.stopPropagation()**, la propagación del evento se detendrá y el evento de clic en el div no se disparará. Por lo tanto, no verás el segundo alerta ("Clic en el div") cuando hagas clic en el botón.

**Por que hay necesidad de detener el evento? no termino de entender? es por una regla de js?**

Sí, la necesidad de detener el evento utilizando **event.stopPropagation()** está relacionada con cómo funciona la propagación de eventos en JavaScript y en el modelo de DOM (Document Object Model).

Cuando haces clic en un elemento dentro de otro elemento (por ejemplo, un botón dentro de un div), el evento de clic se propaga desde el elemento interno (botón) hasta el elemento externo (div) en lo que se llama "burbuja" o fase de propagación. Esto significa que primero se manejará el evento del elemento interno y luego se propagará al elemento externo.

En el ejemplo que proporcioné, si no se detiene la propagación del evento en el botón utilizando **event.stopPropagation()**, el evento de clic del botón se propagaría al div después de manejar el evento del botón. Esto resultaría en la ejecución del manejador de evento en el div, lo que causaría el segundo alerta ("Clic en el div").

Deteniendo la propagación con **event.stopPropagation()**, se asegura de que solo se maneje el evento del botón y no se propague más allá del botón. Esto evita que el manejador de evento del div se active cuando haces clic en el botón.

En resumen, es una técnica para controlar cómo se propagan y manejan los eventos en elementos anidados dentro de otros elementos. Si deseas que los eventos se manejen de manera independiente en el botón y el div, detener la propagación del evento es una buena práctica para evitar que ambos eventos se activen.