

event bubbling

El "event bubbling" (burbujeo de eventos) es un concepto importante en el modelo de eventos de JavaScript en el contexto del Document Object Model (DOM). Se refiere a la propagación de eventos desde el elemento objetivo del evento hacia arriba a través de la jerarquía de elementos DOM.

Cuando se produce un evento en un elemento DOM, como un clic en un botón, el evento no se limita al elemento en el que ocurrió. En cambio, el evento se propaga hacia arriba a través de los elementos padres del elemento objetivo. Esto permite que los elementos contenedores también tengan la oportunidad de responder al evento si tienen event listeners asociados a ese tipo de evento.

El evento primero atraviesa el elemento objetivo (por ejemplo, el botón), luego su padre, luego el abuelo y así sucesivamente, hasta llegar al elemento raíz del documento (generalmente el `` o el ``). Durante cada fase de la propagación, los event listeners asociados a los elementos pueden ser activados si coinciden con el tipo de evento que se está propagando.

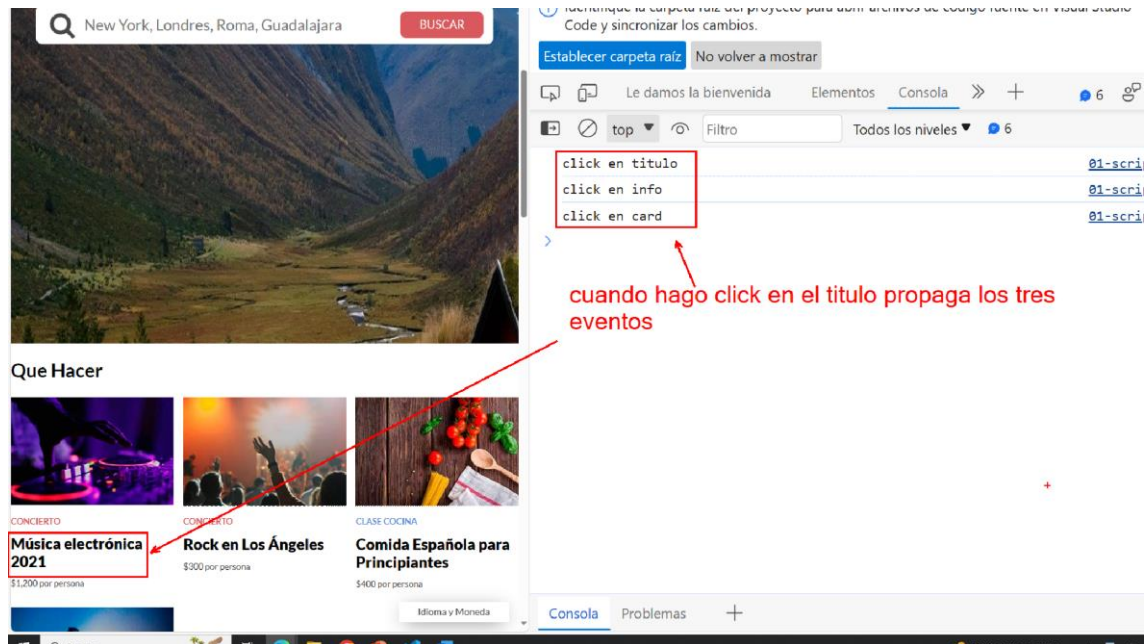
El "event bubbling" tiene ventajas en la organización y el manejo de eventos en una página web, ya que permite capturar eventos en elementos contenedores en lugar de tener que agregar un event listener a cada elemento individual. Sin embargo, en ocasiones, puede resultar en comportamientos inesperados o conflictos si no se gestiona adecuadamente.

Para evitar el burbujeo de eventos, puedes utilizar el método `event.stopPropagation()` en el manejador de eventos para detener la propagación del evento en una fase específica. Por ejemplo:

El problema:

```
const carDiv = document.querySelector('.card')
const infoDiv = document.querySelector('.info')
const tituloDiv = document.querySelector('.titulo')

carDiv.addEventListener('click', () => {
  console.log('click en card')
})
infoDiv.addEventListener('click', () => {
  console.log('click en info')
})
tituloDiv.addEventListener('click', () => {
  console.log('click en titulo')
})
```



La solución: **event.stopPropagation()**

```
carDiv.addEventListener('click', (event) => {
  event.stopPropagation()
  console.log('click en card')
})
infoDiv.addEventListener('click', (e) => {
  e.stopPropagation()
  console.log('click en info')
})
tituloDiv.addEventListener('click', (e) => {
  e.stopPropagation()
  console.log('click en titulo')
})
```

Ten en cuenta que si deseas cancelar completamente el evento (incluso su comportamiento predeterminado), puedes usar `event.preventDefault()` en lugar de `event.stopPropagation()`.