

**Cristian Fernando**

Posted on 22 de ago. de 2022



6



1

# Event bubbling en JavaScript 🍌

#javascript #webdev #spanish #programming

El usuario de una app interacciona con esta mediante la implementación de eventos, como por ejemplo:

- Dar click a un botón
- Hacer doble click o click derecho
- Redimensionar la pantalla
- Hacer scroll, etc.

En JavaScript existe un concepto denominado **event bubbling** que se da sobre todo con nodos del DOM que están anidados unos dentro de otros, es un concepto interesante y preguntado en entrevistas de trabajo, así que empecemos.

## Event bubbling en JavaScript

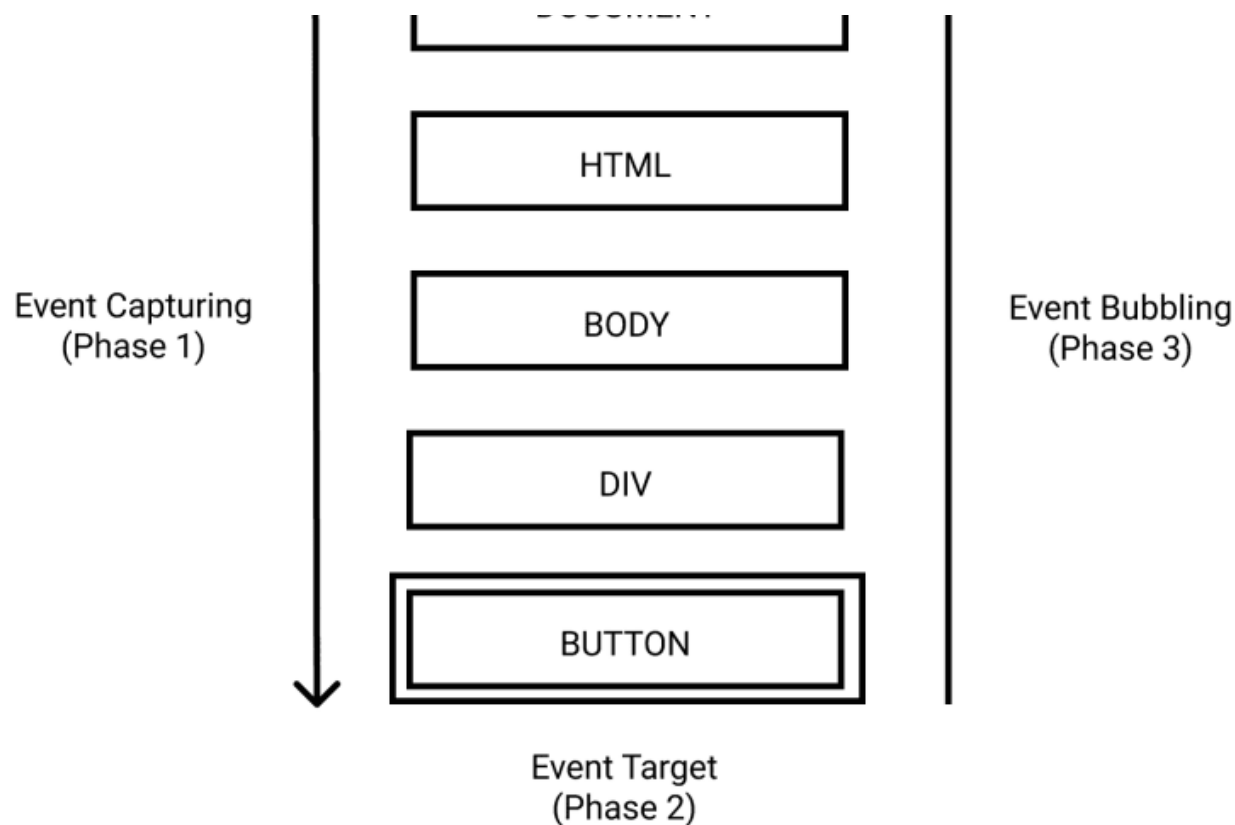
El **event bubbling** o burbujeo de eventos (por su traducción al español) es un método de propagación de eventos en la API del DOM.

Se da cuando activamos el evento de un elemento, y si su nodo padre tiene registrado otro evento, este último se activara automáticamente y así ira escalando en la jerarquía del DOM.

## Fases del Event bubbling

El Event bubbling pasa por 3 fases bien definidas:



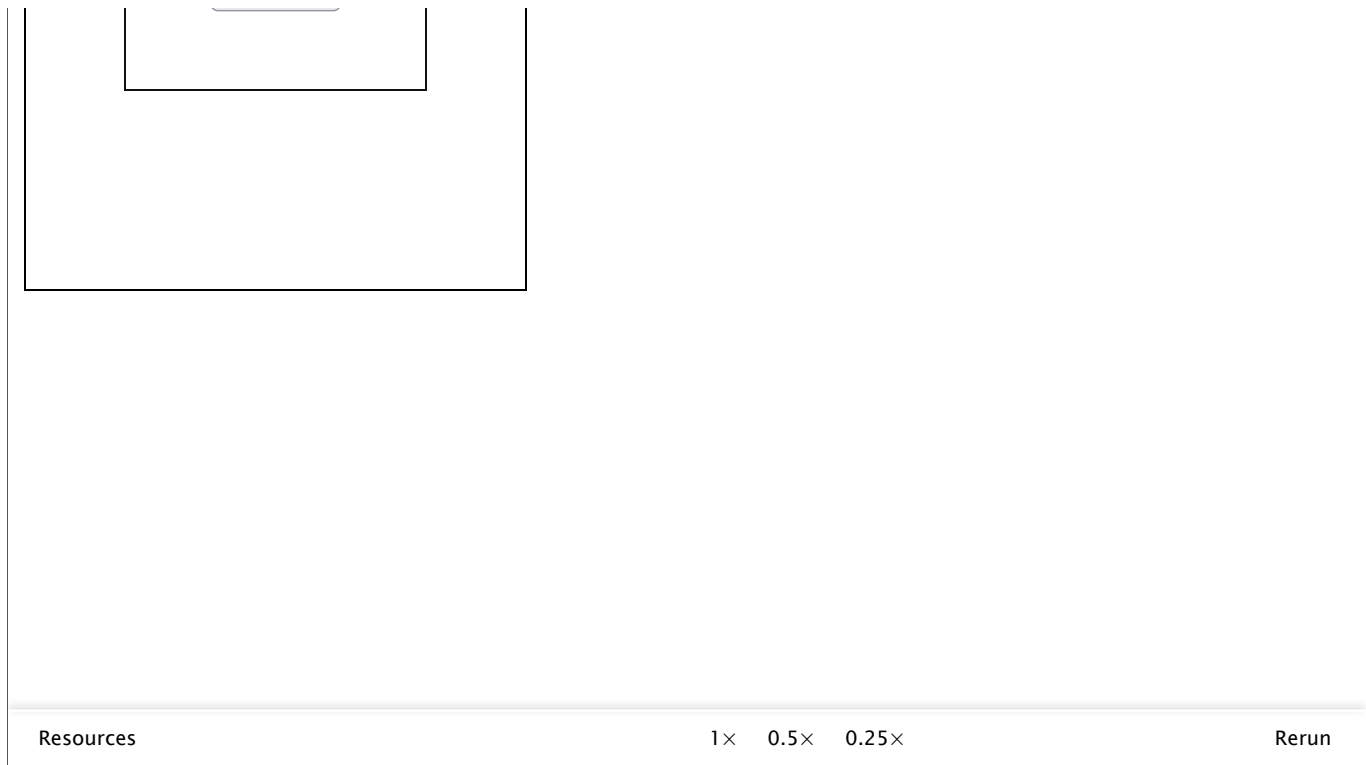


- **Fase de Captura:** Se busca el elemento mas profundo en el DOM que tenga registrado un evento en su listener.
- **Fase de Target:** Ejecuta el evento del elemento en si.
- **Fase de Burbuja:** Verifica si los elementos padre de dicho elemento tienen eventos registrados en sus listeners, si es así, ejecuta dichos eventos de manera automática.

Estas 3 fases se ejecutan por defecto en JavaScript, es posible cambiar dicho comportamiento, pero esto lo veremos mas adelante.

Explicarlo en palabras es un poco confuso, mejor verlo con ejemplos:





En el ejemplo:

- Al hacer click en el botón, activamos su evento, y mostramos una alerta.
- Como el contenedor padre del button tiene también registrado un evento, por Event bubbling este se activa, mostrando automáticamente una segunda alerta.

Justamente con este ejemplo se comprende mejor por que se llama evento de burbuja, por que los eventos se van expandiendo desde el elemento mas profundo al más externo, como si simulara una burbuja.

Una buena manera de desactivar este comportamiento es usando el método `stopPropagation()` del evento en si, de la siguiente manera:

```
button.addEventListener("click", (e) => {  
  e.stopPropagation()  
  alert("Clicked!")  
})
```

El nombre del método es muy intuitivo, simplemente desactiva la propagación de eventos, dejando el resultado de una manera mas obvia y simple.

Inténtalo tu mismo alterando el codepen de ejemplo!

## Modo captura

Por defecto, JavaScript se comporta con Event bubbling, pero podemos alterar el orden de los eventos cambiando al modo captura.

Esto se logra añadiendo un tercer parámetro a nuestro listener padre, como objeto, pasamos la propiedad `capture:true`.

Como vemos en el ejemplo:

- Si damos click al container, todo en orden y sin novedades ni sorpresas.
- Si damos click, por ejemplo, al `div-1` primero mostrara `container` y luego recién `div-1`, por el alert, esto se debe a que tenemos activado el modo captura en el

div-1 por el alert, esto se debe a que tenemos activado el modo captura en el contenedor padre.

- El modo captura invierte las cosas, antes con el event bubbling los eventos se expandían desde lo mas profundo hasta lo mas externo del árbol de nodos; ahora se expedirá desde lo mas externo hasta lo mas profundo.

## Referencias

- <https://www.geeksforgeeks.org/event-bubbling-in-javascript/>
- <https://www.javatpoint.com/event-bubbling-and-capturing-in-javascript>

## Conclusiones

- El event bubbling es una característica un poco extraña pero necesaria de comprender para hacer apps mas robustas.
- El event bubbling tiene tres faces: captura, target y bubbling.
- El event bubbling se da desde lo mas profundo del arbol DOM y sale hacia los elementos mas externos.
- El modo captura se da desde los elementos mas externos y va entrando a los elementos mas profundos.
- El modo captura se activa pasando al listener un tercer parámetro que es un objeto.
- La mejor manera de detener una propagación de eventos es usando el método `stopPropagation()`

---

## Top comments (2)



Felipe • 2 de dec. de 23



Mil gracias!! Me has ayudado a entender el efecto burbujeo fácilmente.



Cristian Fernando 🌟 • 3 de dec. de 23



gracias a ti por leer mi post y dejar tu comentario



## Cristian Fernando

Literatura, tecnología y café (mucho café ☕) es todo lo que necesito.

### LOCATION

La Paz, Bolivia

### EDUCATION

Universidad La Salle - Bolivia

### JOINED

20 de maio de 2020

## More from Cristian Fernando

Ibuprofeno.py 🍌 | #68: Explica este código Python

#python #spanish #programming #learning

Lunes de música... ¿Qué estás escuchando? 🎵 (31-03-2024)

#spanish #music #discuss

Ibuprofeno.py 🍌 | #67: Explica este código Python

#spanish #beginners #learning #python