

Gestión de eventos III

avanzado

Captura y Burbuja

- Estas son dos etapas distintas de la ejecución de un evento.
- La primera **captura** (*capture en Inglés*) se ejecuta antes del comienzo del evento,
- Mientras que la segunda, **burbuja** (*bubbling en Inglés*), se ejecuta después de que el evento se activó.
- Estos dos parámetros definen el sentido de propagación de los eventos.

Ejemplo de propagación de un evento

Código: HTML

```
<div>
```

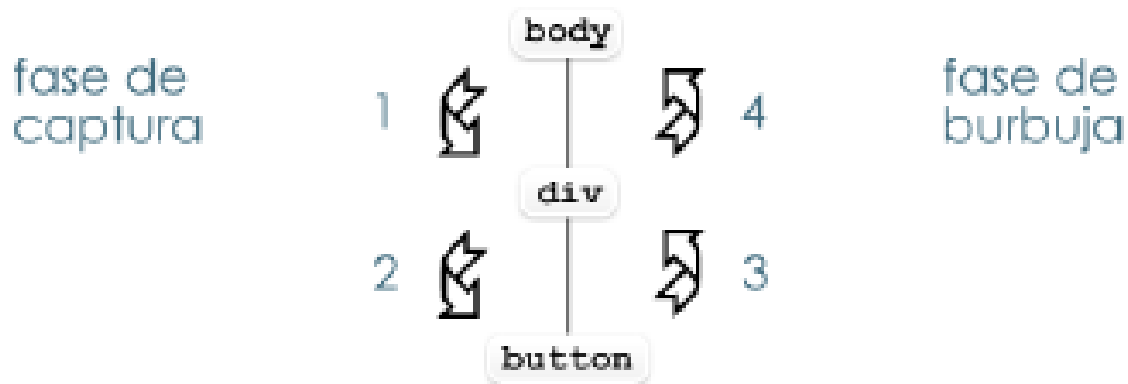
```
<span> texto </span>
```

```
</div>
```

Si asignamos un evento click a cada uno de los dos elementos ¿cuál se ejecuta primero?

Ejemplo de propagación de un evento

- La respuesta esta en el método de propagación que utilicemos.



Código html

- `<div id="capt1">`
- `Instrucciones para la fase de captura. </ span>`
- `</div>`
- `<div id="burb1">`
- `Instrucciones para la fase de burbuja. </ span>`
- `</div>`

script

Extraemos los nodos

- `capt1 = document.getElementById ('capt1')`
- `capt2 = document.getElementById ('capt2')`
- `burb1 = document.getElementById ('burb1')`
- `burb2 = document.getElementById ('burb2');`

Script:addEventListener(False/True)

```
capt1.addEventListener ('click', function () {alert ("El  
evento de div acaba de desencadenarse.");},true);  
capt2.addEventListener ('click', function () {alert ("El  
evento de span acaba de desencadenarse.");},true);
```

```
burb1.addEventListener ('click', function () {alert ("El  
evento de div acaba de desencadenarse.");},false);  
burb2.addEventListener('click', function() {alert ("El  
evento de span acaba de desencadenarse.");},false);
```

Event.stopPropagation

- Evita el burbujeo de eventos

HTML:

```
<div id="parent">  
  <div id="child"></div>  
</div>
```

Javascript:

```
var parent = document.querySelector('#parent');  
var child = document.querySelector('#child');  
  
child.addEventListener('click', function(e) {  
  e.stopPropagation();  
  alert('child clicked!');  
});  
  
parent.addEventListener('click', function(e) {  
  alert('parent clicked!');  
});
```


preventDefault

- Para anular la acción de un evento utilizamos este método del objeto event.
- Sintaxis:
e.preventDefault();

```
const enlace = document.getElementById('mi-enlace');  
  
enlace.addEventListener('click', event => {  
  // Código que quieres que se ejecute  
  event.preventDefault();  
});
```