DOM Event Capturing vs Bubbling

El trabajo con DOM (Document Object Model) es continuo en JavaScript. Pero quizás lo que más se usa es la gestión de eventos (click, mouseover,mouseout, etc). Este es uno de los apartados de JavaScript que se puede decir es sencillo de entender. Normalmente se genera un evento y se asocia un handler a él. **Lamentablemente las cosas no siempre son lo que parecen.**

# Un ejemplo clásico

Para poder entender mejor el funcionamiento de los eventos hay que partir de un bloque de código reducido:

|  |
| --- |
| <div id="zona">      <div id="subzona">          <p><input type="button" id="a" value="a"></p>      </div>  </div> |

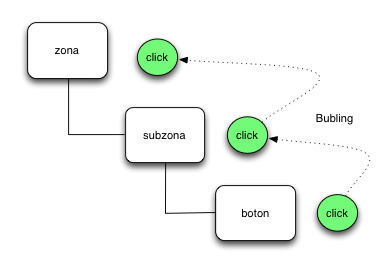
|  |  |
| --- | --- |
|  |  |

A este bloque se le asocia el siguiente código de JavaScript:

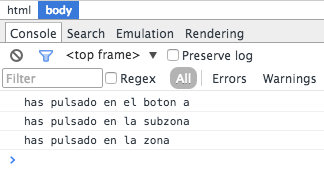
|  |
| --- |
| window.onload=function() {             var a= document.getElementById("a");         var zona= document.getElementById("zona");         var subzona=document.getElementById("subzona");             a.addEventListener("click",f1);         subzona.addEventListener("click",f2);         zona.addEventListener("click",f3);       }       function f1() {           console.log("has pulsado en el boton a");     }     function f2() {           console.log("has pulsado en la subzona")     }       function f3() {           console.log("has pulsado en la zona")     } |

|  |  |
| --- | --- |
|  |  |

El código permite ver en la consola como se han registrado los diferentes manejadores **y como al pulsar el botón "a", el evento "click"** reacciona como una burbuja subiendo por el árbol DOM desde la etiqueta "input" hasta el body.



La consola muestra como reaccionan cada uno de los manejadores:



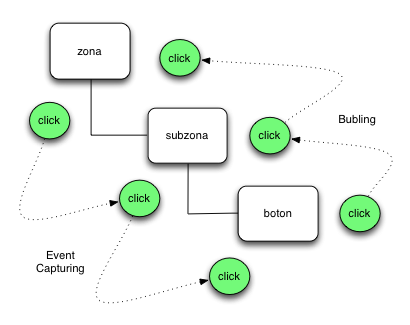
# Event Capturing

JavaScript siempre nos sorprende **y permite una segunda vuelta de tuerca**, ya que aparte de soportar event bubbling soporta también **event capturing**. Para ello deberemos registrar los manejadores para que se ejecuten también en la fase de captura.

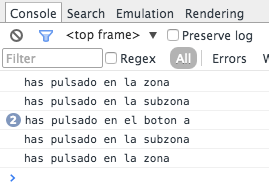
|  |
| --- |
| var a= document.getElementById("a");  var zona= document.getElementById("zona");  var subzona=document.getElementById("subzona");    a.addEventListener("click",f1,**true**);  subzona.addEventListener("click",f2,**true**);  zona.addEventListener("click",f3,**true**);    a.addEventListener("click",f1);  subzona.addEventListener("click",f2);  zona.addEventListener("click",f3); |

|  |  |
| --- | --- |
|  |  |

Esta operación se realiza a**ñadiendo un tercer parámetro al método addEventListener** **y asignando true**. De esta forma habremos registrado un listener en ambas fases.



La consola mostrará el resultado:



En la mayor parte de las situaciones es suficiente con gestionar los eventos a través del sistema de bubling. **Sin embargo existen algunos eventos que no realizan bubling como focus y blur en donde la fase de capturing es clave.**