Event bubbling

From Wikipedia, the free encyclopedia

[Jump to navigation](https://en.wikipedia.org/wiki/Event_bubbling#mw-head)[Jump to search](https://en.wikipedia.org/wiki/Event_bubbling#searchInput)

**Event bubbling** is a type of event propagation[[1]](https://en.wikipedia.org/wiki/Event_bubbling#cite_note-:2-1) where the event first triggers on the innermost target element, and then successively triggers on the ancestors (parents) of the target element in the same nesting hierarchy till it reaches the outermost [DOM](https://en.wikipedia.org/wiki/Document_Object_Model) element or document object[[2]](https://en.wikipedia.org/wiki/Event_bubbling#cite_note-2) (Provided the handler is initialized). It is one way that events are handled in the browser. Events are actions done by the user such as a button click, changing a field etc. [Event Handlers](https://en.wikipedia.org/wiki/Event_handlers) are used to execute code when a particular kind of user interface event occurs, such as when a button has been clicked or when a webpage has completed loading.

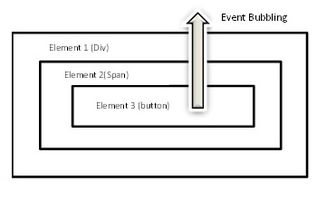


**Contents**

* [1Overview](https://en.wikipedia.org/wiki/Event_bubbling#Overview)
* [2Implementation](https://en.wikipedia.org/wiki/Event_bubbling#Implementation)
* [3Use of Event Bubbling](https://en.wikipedia.org/wiki/Event_bubbling#Use_of_Event_Bubbling)
* [4Preventing event bubbling](https://en.wikipedia.org/wiki/Event_bubbling#Preventing_event_bubbling)
* [5See also](https://en.wikipedia.org/wiki/Event_bubbling#See_also)
* [6References](https://en.wikipedia.org/wiki/Event_bubbling#References)

Overview[[edit](https://en.wikipedia.org/w/index.php?title=Event_bubbling&action=edit&section=1)]

Consider the DOM structure where there are 3 elements nested in the following order: Element 1 (Div), Element 2 (Span), Element 3 (Button) whose on-click handlers are handler1(), handler2() and handler3() respectively.

[](https://en.wikipedia.org/wiki/File:Event_bubbling.jpg)

How the event bubbling works in DOM structure

<div id="Element1" onclick="handler1()">

<span id="Element2" onclick="handler2()">

<input type="button" id="Element3" onclick="handler3()"/>

</span>

</div>

When the Element3 button is clicked, an event handler for Element 3 is triggered first, then event bubbles up and the handler for immediate parent element - Element 2 is called, followed by the handler for Element 1 and so on till it reaches the outermost DOM element.

Event handling order: handler3() - > handler2() -> handler1()

The innermost element from where the event is triggered is called the target element.[[3]](https://en.wikipedia.org/wiki/Event_bubbling#cite_note-3) Most of the browsers consider event bubbling as the default way of event propagation. However, there is another approach for event propagation known as [**Event Capturing**](http://javascript.info/tutorial/bubbling-and-capturing),[[4]](https://en.wikipedia.org/wiki/Event_bubbling#cite_note-4) which is the direct opposite of event bubbling, where event handling starts from the outermost element (or Document) of the DOM structure and goes all the way to the target element, executing the target element handler last in order.

Implementation[[edit](https://en.wikipedia.org/w/index.php?title=Event_bubbling&action=edit&section=2)]

All the event handlers consider event bubbling as the default way of event handling. But a user can manually select the way of propagation by specifying that as the last parameter in [**addEventListener()**](https://www.w3schools.com/jsref/met_document_addeventlistener.asp) [[5]](https://en.wikipedia.org/wiki/Event_bubbling#cite_note-5) of any element in JavaScript.

addEventListener("type", "Listener", "CaptureMode")

If the CaptureMode is False, the event will be handled using event bubbling.

If the CaptureMode is True, the event will be handled using event capturing.

If a user doesn’t specify any value of CaptureMode argument, then it is by default considered as event bubbling. Most of the browser support both event bubbling and event capturing (Except IE <9 and Opera<7.0 which do not support event capturing).[[1]](https://en.wikipedia.org/wiki/Event_bubbling#cite_note-:2-1)

JavaScript also provides an event property called *bubbles* to check whether the event is bubbling event or not. It returns a Boolean value True or False depending on whether the event can bubble up to the parent elements in DOM structure or not.

var isBubblePossible = event.bubbles;

isBubblePossible : True, if event can bubble up to the ancestors

isBubblePossible : False, if event cannot bubble up[[6]](https://en.wikipedia.org/wiki/Event_bubbling#cite_note-6)

Use of Event Bubbling[[edit](https://en.wikipedia.org/w/index.php?title=Event_bubbling&action=edit&section=3)]

To handle cases where one event has more than one handler, event bubbling concept can be implemented. The major use of event bubbling is the registration of default functions present in the program. In recent times, not many developers use event capturing or bubbling in particular. It is not necessary to implement event bubbling; it may become complicated for the users to keep track of the actions getting executed because of an event.[[1]](https://en.wikipedia.org/wiki/Event_bubbling#cite_note-:2-1)

Preventing event bubbling[[edit](https://en.wikipedia.org/w/index.php?title=Event_bubbling&action=edit&section=4)]

It is sometimes useful to stop a single trigger on one element lead to multiple triggers on ancestors. JavaScript provides the following methods to prevent event bubbling:

1) **stopPropagation()**: This method stops the further propagation of any particular event to its parents, invoking only the event handler of the target element. Although supported by all [W3C compliant browsers](https://en.wikipedia.org/wiki/Standards-compliant), Internet Explorer below version 9 requires the historical alias **cancelBubble**[[7]](https://en.wikipedia.org/wiki/Event_bubbling#cite_note-:1-7), as in:

event.cancelBubble = true;

For all W3C compliant browsers:

event.stopPropagation();

2) **stopImmediatePropagation()**: This method will not only stop the further propagation but also stops any other handler of the target event from executing. In the DOM, the same event can have multiple independent handlers, so stopping the execution of one event handler generally doesn’t affect the other handlers of the same target. But stopImmediatePropagation() method prevents the execution of any other handler of the same target.[[7]](https://en.wikipedia.org/wiki/Event_bubbling#cite_note-:1-7)

For all W3C compliant browsers:

event.stopImmediatePropagation();

Another approach to stop event bubbling is to cancel the event itself, however this prevents the target handler execution as well.

See also[[edit](https://en.wikipedia.org/w/index.php?title=Event_bubbling&action=edit&section=5)]

* [DOM events](https://en.wikipedia.org/wiki/DOM_events)

References[[edit](https://en.wikipedia.org/w/index.php?title=Event_bubbling&action=edit&section=6)]

* 1. ^ [Jump up to:***a***](https://en.wikipedia.org/wiki/Event_bubbling#cite_ref-:2_1-0) [***b***](https://en.wikipedia.org/wiki/Event_bubbling#cite_ref-:2_1-1) [***c***](https://en.wikipedia.org/wiki/Event_bubbling#cite_ref-:2_1-2) [*"Javascript - Event order"*](http://www.quirksmode.org/js/events_order.html)*. www.quirksmode.org. Retrieved 2016-09-11.*
  2. [**^**](https://en.wikipedia.org/wiki/Event_bubbling#cite_ref-2) [*"HTML DOM Document Objects"*](https://www.w3schools.com/jsref/dom_obj_document.asp)*. www.w3schools.com. Retrieved 2016-09-11.*
  3. [**^**](https://en.wikipedia.org/wiki/Event_bubbling#cite_ref-3) <https://www.w3schools.com/jsref/event_target.asp>
  4. [**^**](https://en.wikipedia.org/wiki/Event_bubbling#cite_ref-4) [*"Bubbling and capturing | JavaScript Tutorial"*](http://javascript.info/tutorial/bubbling-and-capturing)*. javascript.info. Retrieved 2016-09-11.*
  5. [**^**](https://en.wikipedia.org/wiki/Event_bubbling#cite_ref-5) [*"HTML DOM addEventListener() Method"*](https://www.w3schools.com/jsref/met_document_addeventlistener.asp)*.*
  6. [**^**](https://en.wikipedia.org/wiki/Event_bubbling#cite_ref-6) [*"bubbles Event Property"*](https://www.w3schools.com/jsref/event_bubbles.asp)*. www.w3schools.com. Retrieved 2016-09-11.*
  7. ^ [Jump up to:***a***](https://en.wikipedia.org/wiki/Event_bubbling#cite_ref-:1_7-0) [***b***](https://en.wikipedia.org/wiki/Event_bubbling#cite_ref-:1_7-1) [*"Event Bubbling, How to prevent it ?"*](http://www.markupjavascript.com/2013/10/event-bubbling-how-to-prevent-it.html)*. www.markupjavascript.com. Retrieved 2016-09-11.*