

CIS 4004: Web Based Information Technology Spring 2013

Introduction To JavaScript – Part 6 The Event Object – Event Bubbling And More

Instructor : Dr. Mark Llewellyn
markl@cs.ucf.edu
HEC 236, 407-823-2790
<http://www.cs.ucf.edu/courses/cis4004/spr2013>

Department of Electrical Engineering and Computer Science
University of Central Florida



JavaScript – Part 6 – Bubbling Events

- As we've covered in the two previous sections of JavaScript notes, an event (e.g., a `click`) can trigger an event handler (`onclick`) that is attached to a DOM object.
- However, an event isn't just received by that one DOM element: events travel up and down the DOM tree and are received by any elements they pass through along the way.
- This feature is known as **event propagation**; and it gives you some interesting options in how and where events get handled, and which elements the event affects.
- Often an event's target is not the element that you want to change when the event occurs. For example, you might provide the user a button to click which changes the background image applied to the encompassing `<div>` element.



JavaScript – Part 6 – Bubbling Events

- To understand how to find and modify elements that are not the target of the event, you need to understand the concepts of **event capturing**, **event bubbling**, and **event delegation**.

Capturing and Bubbling

- When an event occurs (let's say it's a `click` event for the sake of this discussion), that `click` event does not go directly to the target of the event.
- It is first sent to the object at the top of the DOM hierarchy, `body`, and then moves down through the document tree to the target – the object that actually received the `click`. The message's downward journey to the target element is known as the **capturing phase**.



JavaScript – Part 6 – Bubbling Events

Capturing and Bubbling

- Once an event reaches the target object, it then travels back up the DOM to the `body` element: The upward journey is called the **bubbling phase**.
- After the event makes it all the way back up to the `body` element, its life ends.
- You can add an event handler to any element that receives the `click` event as it makes this “down-and-then-up-again” journey.
- You can exploit the potential of this method to bring great efficiency to your code, because you can now apply a technique called **event delegation**.



JavaScript – Part 6 – Bubbling Events

Event Delegation

- Event delegation is the technique by which you place an event handler on an element that is not the target element of the event.
- The most advantageous use of event delegation is when you attach the event handler to the parent of a large number of child elements that must all provide a response to a particular event.
- Significant coding economies can be realized by taking advantage of event delegation. Instead of attaching individual event handlers to every child, the event of interest is allowed to bubble up from whichever child triggered it and is handled by a single handler that is attached to the parent element.



JavaScript – Part 6 – Bubbling Events

Event Delegation

- If needed, the event object can supply the name of the target element: Then just that one “downstream” element – out of the many that might have triggered the parent’s event handler – can be modified.
- The W3C model supports both the capturing and bubbling phase.
- The third argument of a W3C event registration (adding an event listener) controls the capturing and bubbling.

```
someElement.addEventListener("focus", doHighLight, false)
```

- If the third argument is `true`, the event is to be handled in the capturing phase (when the event trickles down the DOM tree to its target). If the third argument is `false`, the event is to be handled in the bubbling phase (as the event travels back up the DOM tree).



JavaScript – Part 6 – Bubbling Events

- The capturing phase is not supported by the Microsoft event model; only the W3C model supports the capturing phase.
- Event delegation only makes sense as the message travels up the DOM tree from child to parents.
- For this reason, you will almost always use the bubbling phase to delegate events. Thus, in the W3C event model, the third argument will, almost without exception, be `false`.
- The following example illustrates the process of event capturing and event bubbling. [in class demo]



E v e n t B u b b l i n g

This demo page demonstrates how event bubbling works. To run this demo, click anywhere in the boxes below and note how the event is propagated up the object hierarchy from Level 1 up to the Web document itself. To cancel event bubbling at a given point in the document tree, click one of the radio buttons to stop the bubbling beyond that level:

Note that if you click a level above the location of the where the propagation ends, the onclick event propagates from that point upward without being canceled.

Level 4

Level 3

Level 2

Level 1

Cancel Event Bubbling	At Level	Event Detected
<input type="radio"/>	Document	
<input checked="" type="radio"/>	Level 4	
<input type="radio"/>	Level 3	
<input type="radio"/>	Level 2	
<input type="radio"/>	Level 1	



E v e n t B u b b l i n g

This demo page demonstrates how event bubbling works. To run this demo, click anywhere in the boxes below and note how the event is propagated up the object hierarchy from Level 1 up to the Web document itself. To cancel event bubbling at a given point in the document tree, click one of the radio buttons to stop the bubbling beyond that level:

Note that if you click a level above the location of the where the propagation ends, the onclick event propagates from that point upward without being canceled.

Level 4

Level 3

Level 2

Level 1

Cancel Event Bubbling	At Level	Event Detected
<input type="radio"/>	Document	onclick event detected
<input checked="" type="radio"/>	Level 4	onclick event detected
<input type="radio"/>	Level 3	onclick event detected
<input type="radio"/>	Level 2	onclick event detected
<input type="radio"/>	Level 1	onclick event detected

User clicks in level 1 – event captured at level 1 and then event bubbles all the way to the document level



E v e n t B u b b l i n g

This demo page demonstrates how event bubbling works. To run this demo, click anywhere in the boxes below and note how the event is propagated up the object hierarchy from Level 1 up to the Web document itself. To cancel event bubbling at a given point in the document tree, click one of the radio buttons to stop the bubbling beyond that level:

Note that if you click a level above the location of the where the propagation ends, the onclick event propagates from that point upward without being canceled.

Level 4

Level 3

Level 2

Level 1

Cancel Event Bubbling	At Level	Event Detected
<input type="radio"/>	Document	onclick event detected
<input checked="" type="radio"/>	Level 4	onclick event detected
<input type="radio"/>	Level 3	onclick event detected
<input type="radio"/>	Level 2	
<input type="radio"/>	Level 1	

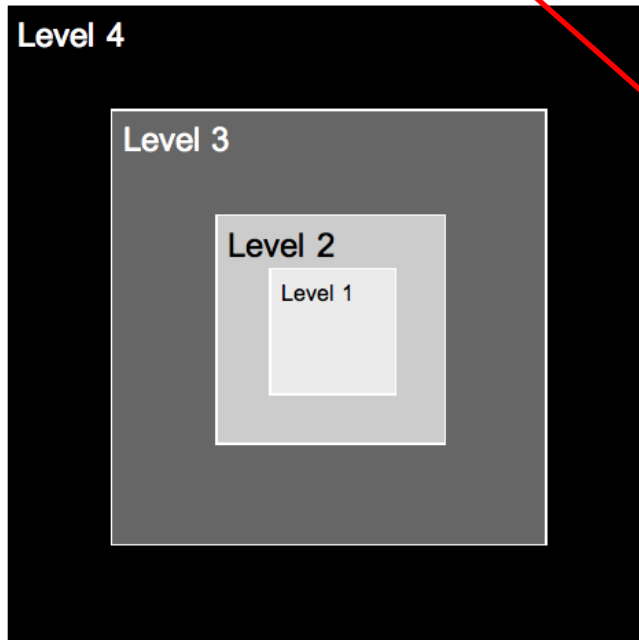
User clicks in level 3 – event is captured at level 3 and then event bubbles all the way to the document level



E v e n t B u b b l i n g

This demo page demonstrates how event bubbling works. To run this demo, click anywhere in the boxes below and note how the event is propagated up the object hierarchy from Level 1 up to the Web document itself. To cancel event bubbling at a given point in the document tree, click one of the radio buttons to stop the bubbling beyond that level:

Note that if you click a level above the location of the where the propagation ends, the onclick event propagates from that point upward without being canceled.



Cancel Event Bubbling	At Level	Event Detected
<input type="radio"/>	Document	onclick event detected
<input checked="" type="radio"/>	Level 4	
<input type="radio"/>	Level 3	
<input type="radio"/>	Level 2	
<input type="radio"/>	Level 1	

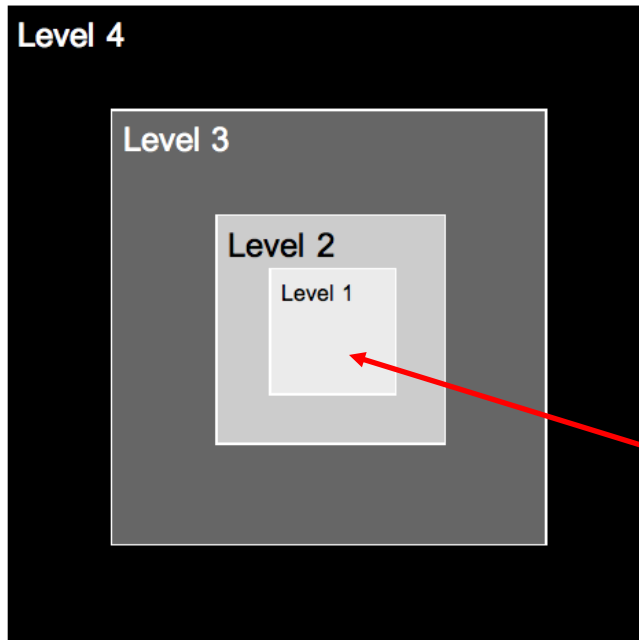
User clicks elsewhere in the document – event is captured only at the document (highest) level 3 and there is no bubbling (no where to bubble).



E v e n t B u b b l i n g

This demo page demonstrates how event bubbling works. To run this demo, click anywhere in the boxes below and note how the event is propagated up the object hierarchy from Level 1 up to the Web document itself. To cancel event bubbling at a given point in the document tree, click one of the radio buttons to stop the bubbling beyond that level:

Note that if you click a level above the location of the where the propagation ends, the onclick event propagates from that point upward without being canceled.



Cancel Event Bubbling	At Level	Event Detected
<input type="radio"/>	Document	
<input checked="" type="radio"/>	Level 4	
<input type="radio"/>	Level 3	
<input checked="" type="radio"/>	Level 2	onclick event detected
<input type="radio"/>	Level 1	onclick event detected

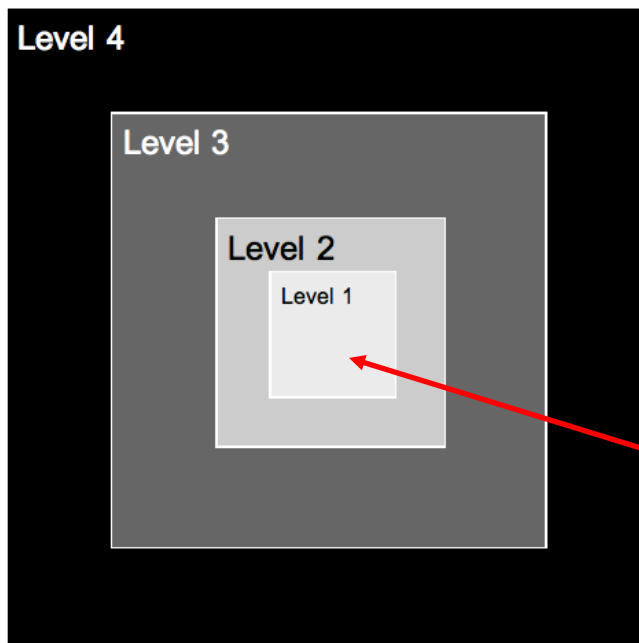
User cancels bubbling beyond level 2 and clicks in level 1 – event is captured at level 1 and bubbles only one more level. This simulates that level 2 would handle the event.



E v e n t B u b b l i n g

This demo page demonstrates how event bubbling works. To run this demo, click anywhere in the boxes below and note how the event is propagated up the object hierarchy from Level 1 up to the Web document itself. To cancel event bubbling at a given point in the document tree, click one of the radio buttons to stop the bubbling beyond that level:

Note that if you click a level above the location of the where the propagation ends, the onclick event propagates from that point upward without being canceled.



Cancel Event Bubbling	At Level	Event Detected
<input type="radio"/>	Document	
<input checked="" type="radio"/>	Level 4	onclick event detected
<input type="radio"/>	Level 3	onclick event detected
<input type="radio"/>	Level 2	onclick event detected
<input type="radio"/>	Level 1	onclick event detected

User cancels bubbling beyond level 4 and clicks in level 1 – event is captured at level 1 and bubbles up to level 4 then stops. This simulates that level 4 would handle the event.



E v e n t B u b b l i n g

This demo page demonstrates how event bubbling works. To run this demo, click anywhere in the boxes below and note how the event is propagated up the object hierarchy from Level 1 up to the Web document itself. To cancel event bubbling at a given point in the document tree, click one of the radio buttons to stop the bubbling beyond that level:

Note that if you click a level above the location of the where the propagation ends, the onclick event propagates from that point upward without being canceled.

Level 4

Cancel Event Bubbling	At Level	Event Detected
<input type="radio"/>	Document	onclick event detected
<input checked="" type="radio"/>	Level 4	onclick event detected
<input type="radio"/>	Level 3	onclick event detected
<input checked="" type="radio"/>	Level 2	
<input type="radio"/>	Level 1	

User cancels bubbling beyond level 2 and clicks in level 3 – event is captured at level 3 and bubbles up to the document level. This simulates that event bubbling which is suppressed at lower levels does not affect higher levels of event propagation.

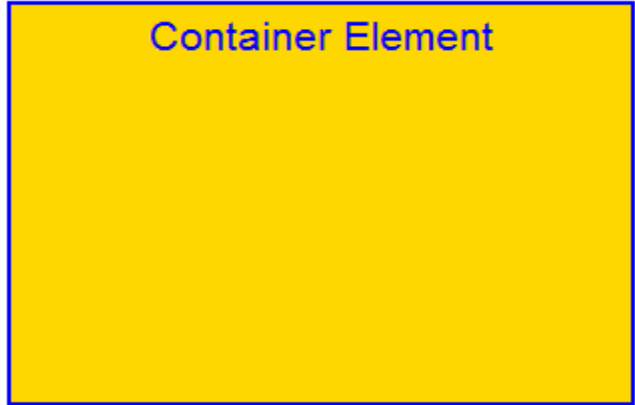


The Event Object

The IE and DOM event models support event objects which return information on events occurring within the browser. To view the properties of events within this browser window, move your mouse pointer around the Web page. The table below will show the values returned by the different object properties. Also, click your mouse button or press a keyboard key to view the properties associated with those events. Note that the offsetX and offsetY properties are only supported in the IE event model.

Event Model IE event model DOM Level 2 Event Object window.event type

Mouse-Related Properties	
button	<input type="text"/>
screenX, screenY	<input type="text" value="742"/> , <input type="text" value="433"/>
clientX, clientY	<input type="text" value="705"/> , <input type="text" value="259"/>
offsetX, offsetY	<input type="text" value="705"/> , <input type="text" value="259"/>
Keyboard-Related Properties	
keyCode	<input type="text" value="18"/>
altKey	<input type="text" value="true"/>
ctrlKey	<input type="text" value="false"/>
shiftKey	<input type="text" value="false"/>



The Event Object

The IE and DOM event models support event objects which return information on events occurring within the browser. To view the properties of e within this browser window, move your mouse pointer around the Web page. The table below will show the values returned by the different obje properties. Also, click your mouse button or press a keyboard key to view the properties associated with those events. Note that the offsetX and properties are only supported in the IE event model.

Event Model Event Object type

Mouse-Related Properties	
button	<input type="text"/>
screenX, screenY	<input type="text" value="426"/> , <input type="text" value="572"/>
clientX, clientY	<input type="text" value="381"/> , <input type="text" value="461"/>
offsetX, offsetY	<input type="text" value=""/> , <input type="text" value=""/>
Keyboard-Related Properties	
keyCode	<input type="text" value="17"/>
altKey	<input type="text" value="false"/>
ctrlKey	<input type="text" value="true"/>
shiftKey	<input type="text" value="false"/>





