

# Dynamic HTML: Event Model

## Outline

- 14.1 Introduction
- 14.2 Event onclick
- 14.3 Event onload
- 14.4 Error Handling with onerror
- 14.5 Tracking the Mouse with Event onmousemove
- 14.6 Rollovers with onmouseover and onmouseout
- 14.7 Form Processing with onfocus and onblur
- 14.8 More Form Processing with onsubmit and onreset
- 14.9 Event Bubbling
- 14.10 More DHTML Events
- 14.11 Web Resources

# Objectives

- In this lesson, you will learn:
  - To understand the notion of events, event handlers and event bubbling.
  - To be able to create event handlers that respond to mouse and keyboard events.
  - To be able to use the event object to be made aware of and, ultimately, respond to user actions.
  - To understand how to recognize and respond to the most popular events.

## 14.1 Introduction

- Event model
  - Scripts can respond to user
  - Content becomes more dynamic
  - Interfaces become more intuitive

## 14.2 Event onclick

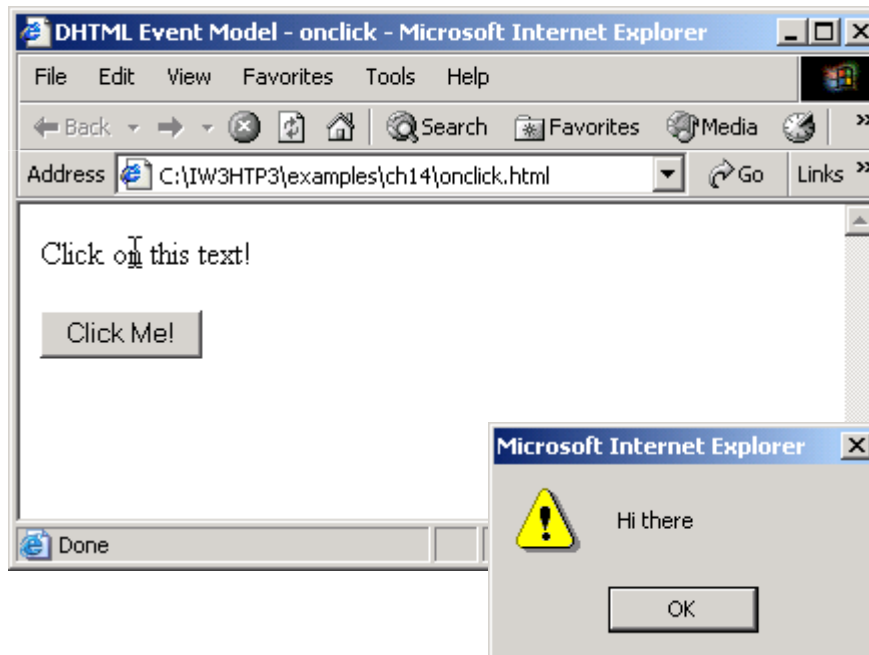
- onclick
  - Invoked when user clicks the mouse on a specific item

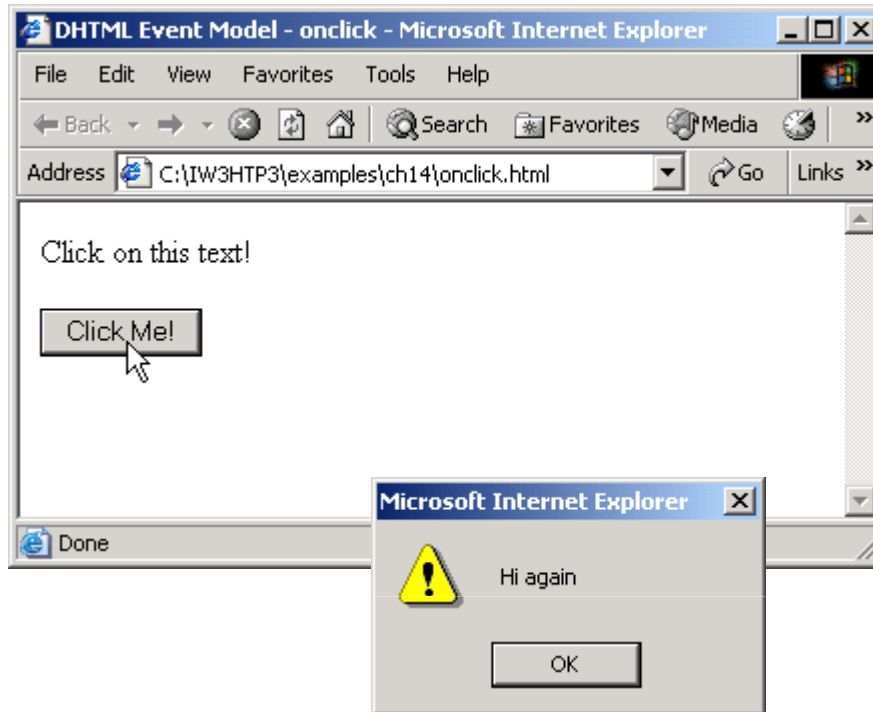
```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5 <!-- Fig 14.1: onclick.html      -->
6 <!-- Demonstrating the onclick event -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>DHTML Event Model - onclick</title>
11
12    <!-- The for attribute declares the script for -->
13    <!-- a certain element, and the event for a -->
14    <!-- certain event. -->
15    <script type = "text/javascript" for = "para"
16      event = "onclick">
17      <!--
18      alert( "Hi there" );
19      // -->
20    </script>
21  </head>
22
23  <body>
24
```

**onclick.html**  
**(1 of 2)**

```
25 <!-- The id attribute gives a unique identifier -->
26 <p id = "para">Click on this text!</p>
27
28 <!-- You can specify event handlers inline -->
29 <input type = "button" value = "Click Me!"
30     onclick = "alert( 'Hi again' )" />
31
32 </body>
33 </html>
```

**onclick.html**  
(2 of 2)





## 14.3 Event onLoad

- onLoad event
  - Fires when an element finishes loading
  - Used in the body element
  - Initiates a script after the page loads into the client

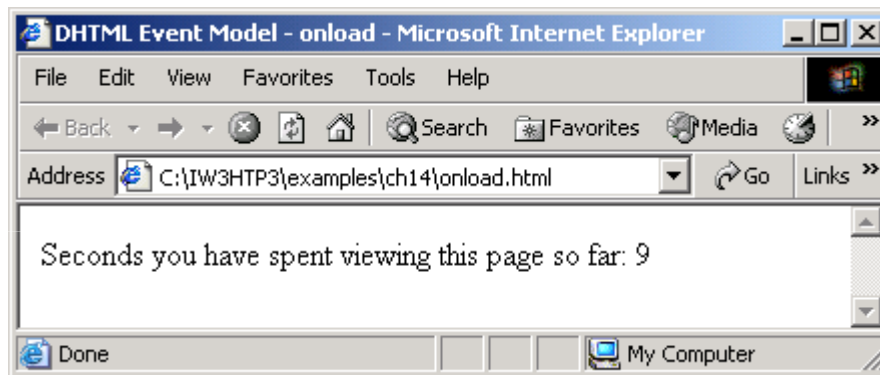


```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig. 14.2: onload.html      -->
6 <!-- Demonstrating the onload event -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10     <title>DHTML Event Model - onload</title>
11     <script type = "text/javascript">
12       <!--
13       var seconds = 0;
14
15       function startTimer() {
16         // 1000 milliseconds = 1 second
17         window.setInterval( "updateTime()", 1000 );
18       }
19
20       function updateTime() {
21         seconds++;
22         soFar.innerHTML = seconds;
23       }
24       // -->
25     </script>
```

**onload.html**  
**(1 of 2)**

```
26 </head>
27
28 <body onload = "startTimer()">
29
30     <p>Seconds you have spent viewing this page so far:
31     <strong id = "soFar">0</strong></p>
32
33 </body>
34 </html>
```

**onload.html**  
**(2 of 2)**



## 14.4 Error Handling with onerror

- onerror event
  - Execute specialized error-handling code

```

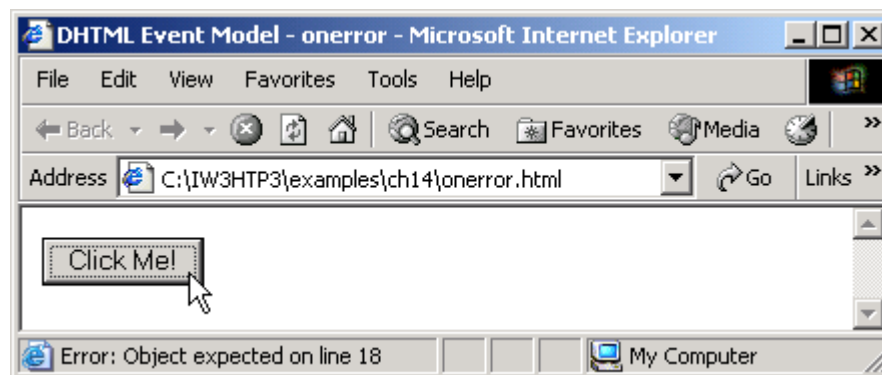
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5 <!-- Fig 14.3: onerror.html      -->
6 <!-- Demonstrating the onerror event -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>DHTML Event Model - onerror</title>
11    <script type = "text/javascript">
12      <!--
13      // Specify that if an onerror event is triggered
14      // in the window function handleError should execute
15      window.onerror = handleError;
16
17      function doThis() {
18        alrrt( "hi" ); // alert misspelled, creates an error
19      }
20
21      // The ONERROR event passes three values to the
22      // function: the name of the error, the url of
23      // the file, and the line number.
24      function handleError( errType, errURL, errLineNum )
25      {

```

**onerror.html**  
(1 of 2)

```
26 // writes to the status bar at the
27 // bottom of the window.
28 window.status = "Error: " + errType + " on line " +
29     errLineNum;
30
31 // Returning a value of true cancels the
32 // browser's reaction.
33 return true;
34 }
35 // -->
36 </script>
37 </head>
38
39 <body>
40
41     <input id = "mybutton" type = "button" value = "Click Me!"
42         onclick = "doThis()" />
43
44 </body>
45 </html>
```

**onerror.html**  
(2 of 2)



## 14.5 Tracking the Mouse with Event `onmousemove`

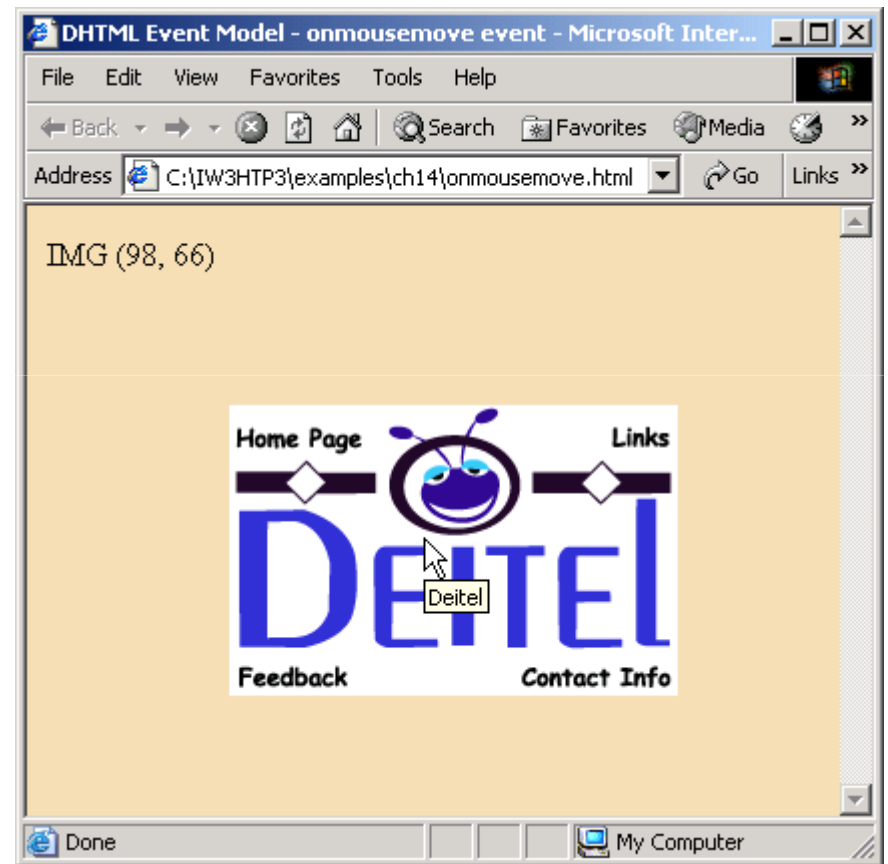
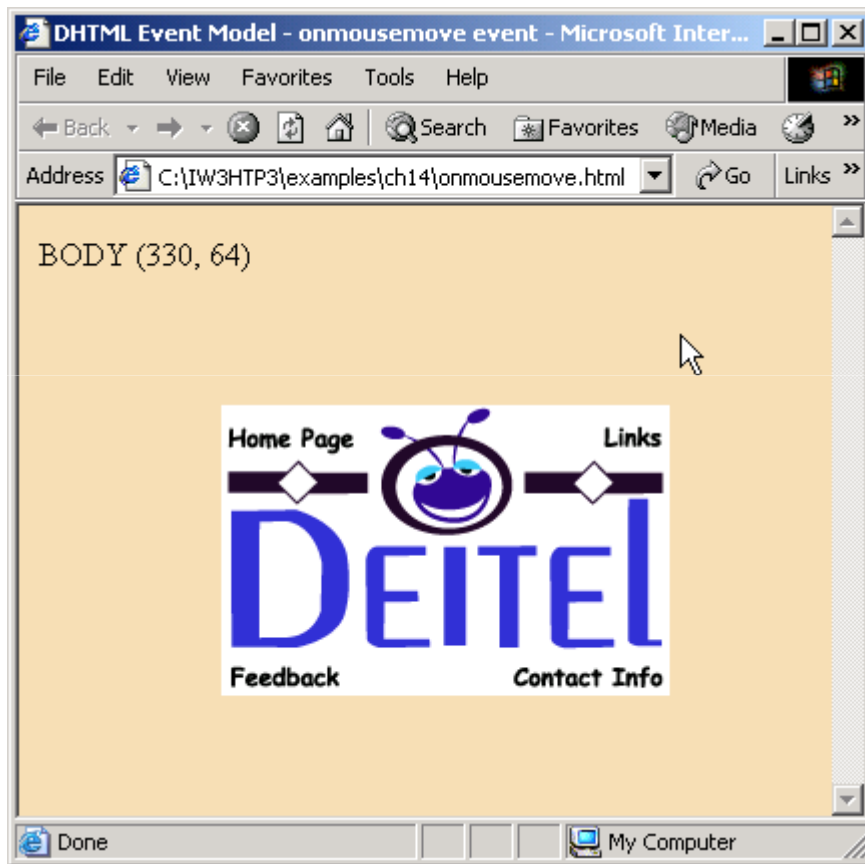
- `onmousemove`
  - Fires repeatedly when the user moves the mouse over the Web page
  - Gives position of the mouse

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5 <!-- Fig. 14.4: onmousemove.html      -->
6 <!-- Demonstrating the onmousemove event -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>DHTML Event Model - onmousemove event</title>
11    <script type = "text/javascript">
12      <!--
13      function updateMouseCoordinates()
14      {
15        coordinates.innerHTML = event.srcElement.tagName +
16          " (" + event.offsetX + ", " + event.offsetY + ")";
17      }
18      // -->
19    </script>
20  </head>
21
22  <body style = "back-groundcolor: wheat"
23    onmousemove = "updateMouseCoordinates()">
24
```

## onmousemove.html (1 of 2)

```
25 <span id = "coordinates">(0, 0)</span><br />
26 <img src = "deitel.gif" style = "position: absolute;
27     top: 100; left: 100" alt = "Deitel" />
28
29 </body>
30 </html>
```

**onmousemove.html**  
**(2 of 2)**





## 14.6 Rollovers with onmouseover and onmouseout

- Two more events fired by mouse movements
  - onmouseover
    - Mouse cursor moves over element
  - Onmouseout
    - Mouse cursor leaves element

## 14.6 Rollovers with onmouseover and onmouseout

Property of event	Description
altkey	This value is <b>true</b> if <i>Alt</i> key was pressed when event fired.
button	Returns which mouse button was pressed by user (1: left-mouse button, 2: right-mouse button, 3: left and right buttons, 4: middle button, 5: left and middle buttons, 6: right and middle buttons, 7: all three buttons).
cancelBubble	Set to <b>false</b> to prevent this event from bubbling (see Section 14.9, “Event Bubbling”).
clientX / clientY	The coordinates of the mouse cursor inside the client area (i.e., the active area where the Web page is displayed, excluding scrollbars, navigation buttons, etc.).
ctrlKey	This value is <b>true</b> if <i>Ctrl</i> key was pressed when event fired.
offsetX / offsetY	The coordinates of the mouse cursor relative to the object that fired the event.
propertyName	The name of the property that changed in this event.
recordset	A reference to a data field’s recordset (see Chapter 16, “Data Binding”).
returnValue	Set to <b>false</b> to cancel the default browser action.
screenX / screenY	The coordinates of the mouse cursor on the screen coordinate system.
shiftKey	This value is <b>true</b> if <i>Shift</i> key was pressed when event fired.
srcElement	A reference to the object that fired the event.
type	The name of the event that fired.
x / y	The coordinates of the mouse cursor relative to this element’s parent element.

Fig. 14.5 Some event object properties.

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig 14.6: onmouseoverout.html -->
6 <!-- Events onmouseover and onmouseout -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>
11      DHTML Event Model - onmouseover and onmouseout
12    </title>
13    <script type = "text/javascript">
14      <!--
15      captionImage1 = new Image();
16      captionImage1.src = "caption1.gif";
17      captionImage2 = new Image();
18      captionImage2.src = "caption2.gif";
19
20      function mOver()
21      {
22        if ( event.srcElement.id == "tableCaption" ) {
23          event.srcElement.src = captionImage2.src;
24          return;
25        }

```

**onmouseoverout  
.html  
(1 of 4)**

```
26
27     // If the element which triggered onmouseover has
28     // an id, change its color to its id.
29     if ( event.srcElement.id )
30         event.srcElement.style.color =
31         event.srcElement.id;
32 }
33
34 function mOut()
35 {
36     if ( event.srcElement.id == "tableCaption" ) {
37         event.srcElement.src = captionImage1.src;
38         return;
39     }
40
41     // If it has an id, change the text inside to the
42     // text of the id.
43     if ( event.srcElement.id )
44         event.srcElement.innerText = event.srcElement.id;
45 }
46
47 document.onmouseover = mOver;
48 document.onmouseout = mOut;
49 // -->
50 </script>
```

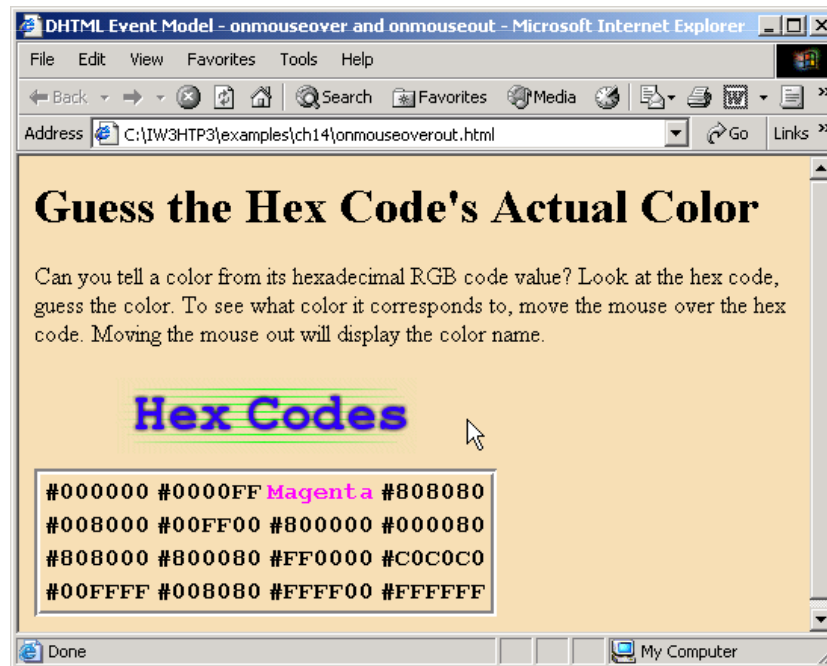
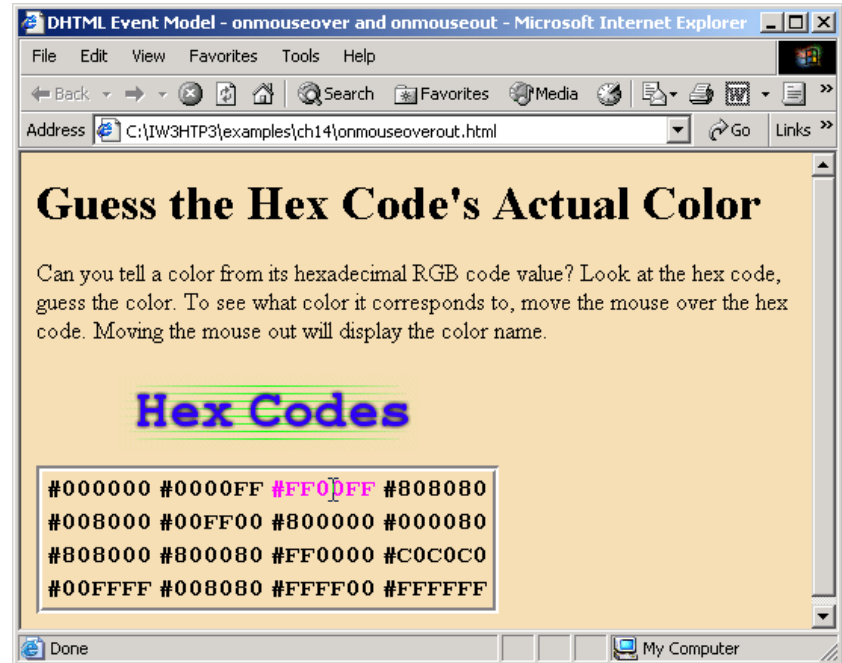
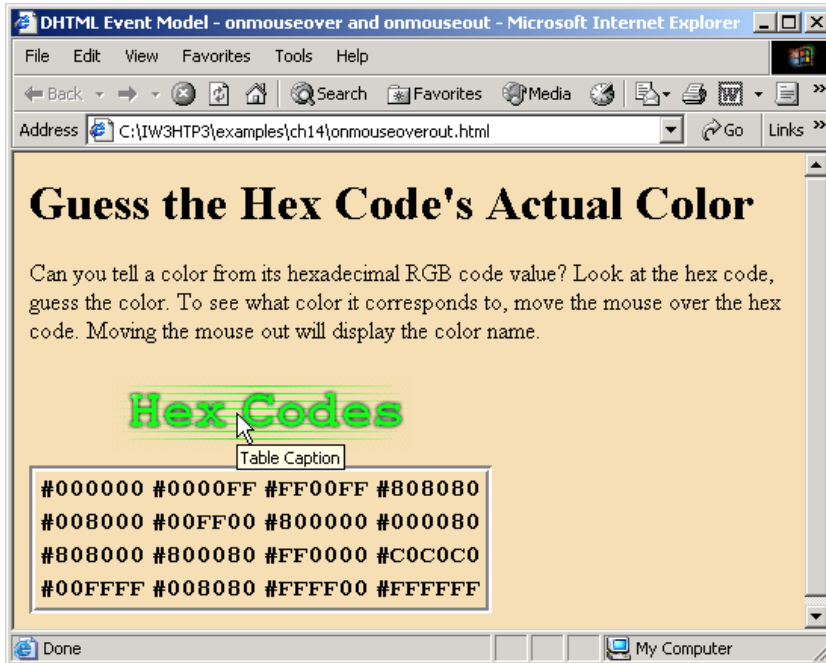
**onmouseoverout  
.html  
(2 of 4)**

```
51 </head>
52
53 <body style = "background-color: wheat">
54
55     <h1>Guess the Hex Code's Actual Color</h1>
56
57     <p>Can you tell a color from its hexadecimal RGB code
58     value? Look at the hex code, guess the color. To see
59     what color it corresponds to, move the mouse over the
60     hex code. Moving the mouse out will display the color
61     name.</p>
62
63     <table style = "width: 50%; border-style: groove;
64         text-align: center; font-family: monospace;
65         font-weight: bold">
66
67         <caption>
68             <img src = "caption1.gif" id = "tableCaption"
69                 alt = "Table Caption" />
70         </caption>
71
72         <tr>
73             <td><a id = "Black">#000000</a></td>
74             <td><a id = "Blue">#0000FF</a></td>
75             <td><a id = "Magenta">#FF00FF</a></td>
```

**onmouseoverout  
.html  
(3 of 4)**

```
76     <td><a id = "Gray">#808080</a></td>
77 </tr>
78 <tr>
79     <td><a id = "Green">#008000</a></td>
80     <td><a id = "Lime">#00FF00</a></td>
81     <td><a id = "Maroon">#800000</a></td>
82     <td><a id = "Navy">#000080</a></td>
83 </tr>
84 <tr>
85     <td><a id = "olive">#808000</a></td>
86     <td><a id = "Purple">#800080</a></td>
87     <td><a id = "Red">#FF0000</a></td>
88     <td><a id = "Silver">#C0C0C0</a></td>
89 </tr>
90 <tr>
91     <td><a id = "Cyan">#00FFFF</a></td>
92     <td><a id = "Teal">#008080</a></td>
93     <td><a id = "Yellow">#FFFF00</a></td>
94     <td><a id = "White">#FFFFFF</a></td>
95 </tr>
96 </table>
97
98 </body>
99 </html>
```

**onmouseoverout  
.html  
(4 of 4)**



## 14.7 Form Processing with `onfocus` and `onblur`

- `onfocus` event fires when element gains focus
- `onblur` event fires when element loses focus



```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5 <!-- Fig. 14.7: onfocusblur.html          -->
6 <!-- Demonstrating the onfocus and onBlur events -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10     <title>DHTML Event Model - onfocus and onBlur</title>
11     <script type = "text/javascript">
12       <!--
13       var helpArray =
14         [ "Enter your name in this input box.",
15           "Enter your email address in this input box, " +
16           "in the format user@domain.",
17           "Check this box if you liked our site.",
18           "In this box, enter any comments you would " +
19           "like us to read.",
20           "This button submits the form to the " +
21           "server-side script",
22           "This button clears the form",
23           "This textarea provides context-sensitive " +
24           "help. Click on any input field or use the TAB " +
25           "key to get more information about the " +
```

## onfocusblur.html (1 of 3)

```

26         "input field." ];
27
28     function helpText( messageNum )
29     {
30         myForm.helpBox.value = helpArray[ messageNum ];
31     }
32     // -->
33 </script>
34 </head>
35
36 <body>
37
38     <form id = "myForm" action = "">
39     Name: <input type = "text" name = "name"
40         onfocus = "helpText(0)" onblur = "helpText(6)" /><br />
41     Email: <input type = "text" name = "email"
42         onfocus = "helpText(1)" onblur = "helpText(6)" /><br />
43     Click here if you like this site
44     <input type = "checkbox" name = "like" onfocus =
45         "helpText(2)" onblur = "helpText(6)" /><br /><hr />
46
47     Any comments?<br />
48     <textarea name = "comments" rows = "5" cols = "45"
49         onfocus = "helpText(3)" onblur = "helpText(6)">
50 </textarea><br />

```

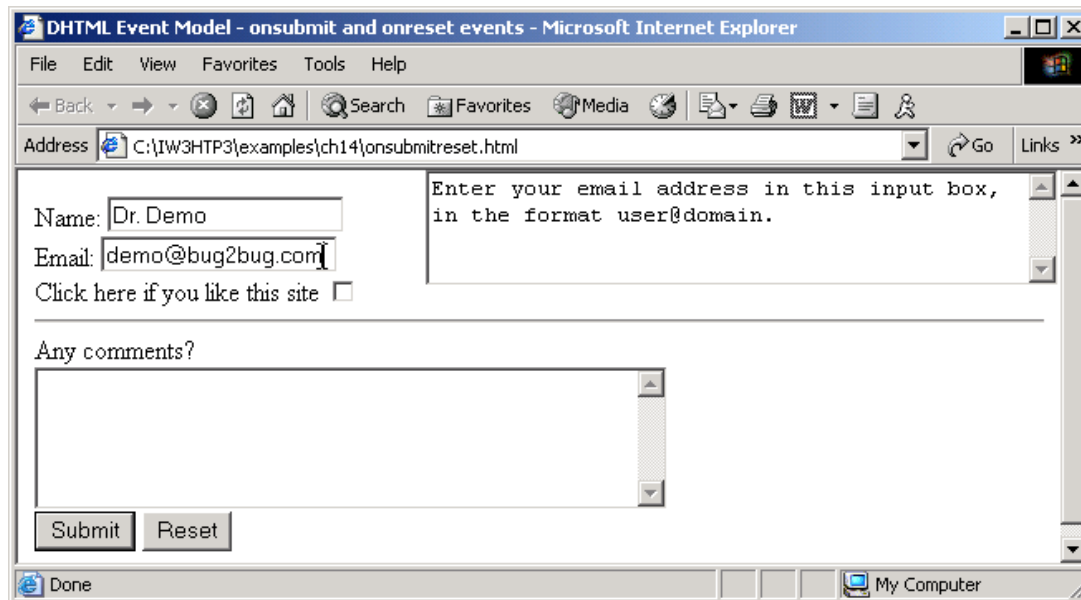
**onfocusblur.html**  
(2 of 3)

```

51 <input type = "submit" value = "Submit" onfocus =
52     "helpText(4)" onblur = "helpText(6)" />
53 <input type = "reset" value = "Reset" onfocus =
54     "helpText(5)" onblur = "helpText(6)" />
55
56 <textarea name = "helpBox" style = "position: absolute;
57     right: 0; top: 0" readonly = "true" rows = "4" cols = "45">
58 This textarea provides context-sensitive help.
59 Click on any input field or use the Tab key
60 to get more information about the input field.</textarea>
61 </form>
62
63 </body>
64 </html>

```

**onfocusblur.html**  
(3 of 3)



## 14.8 More Form Processing with `onsubmit` and `onreset`

- `onsubmit` and `onreset` are useful events for processing forms

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5 <!-- Fig 14.8: onsubmitreset.html -->
6 <!-- Demonstrating the onsubmit and onreset events -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10     <title>
11       DHTML Event Model - onsubmit and onreset events
12     </title>
13     <script type = "text/javascript">
14       <!--
15       var helpArray =
16         [ "Enter your name in this input box.",
17           "Enter your email address in this input box, " +
18           "in the format user@domain.",
19           "Check this box if you liked our site.",
20           "In this box, enter any comments you would " +
21           "like us to read.",
22           "This button submits the form to the " +
23           "server-side script",
24           "This button clears the form",
25           "This textarea provides context-sensitive " +
```

## onsubmitreset.html (1 of 4)

```
26     "help. Click on any input field or use the Tab " +
27     "key to get more information about " +
28     "the input field." ];
29
30     function helpText( messageNum )
31     {
32         myForm.helpBox.value = helpArray[ messageNum ];
33     }
34
35     function formSubmit() {
36         window.event.returnValue = false;
37
38         if ( confirm ( "Are you sure you want to submit?" ) )
39             window.event.returnValue = true;
40     }
41
42     function formReset() {
43         window.event.returnValue = false;
44
45         if ( confirm( "Are you sure you want to reset?" ) )
46             window.event.returnValue = true;
47     }
48     // -->
49 </script>
50 </head>
```

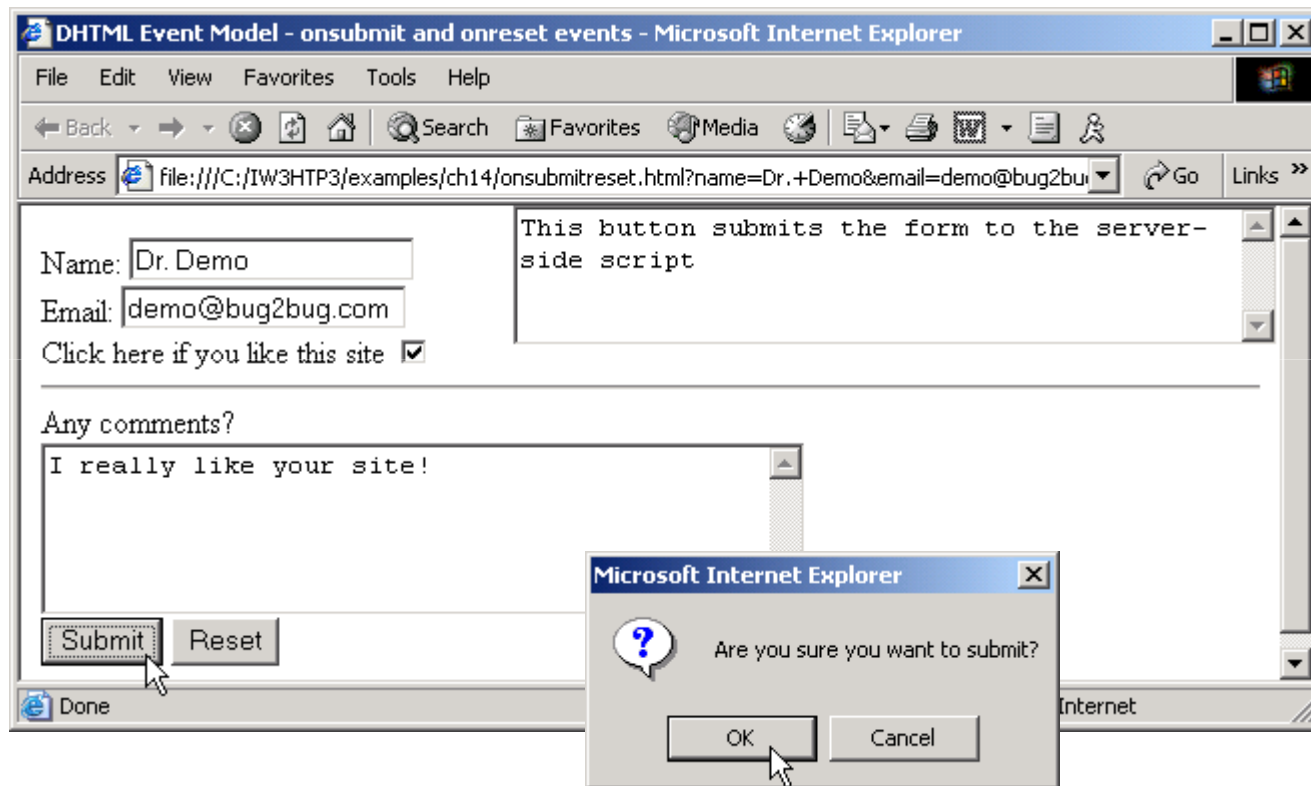
**onsubmitreset.html**  
**(2 of 4)**

```
51
52 <body>
53
54 <form id = "myForm" onsubmit = "formSubmit()"
55   onreset = "formReset()" action = "">
56   Name: <input type = "text" name = "name"
57     onfocus = "helpText(0)" onblur = "helpText(6)" /><br />
58   Email: <input type = "text" name = "email"
59     onfocus = "helpText(1)" onblur = "helpText(6)" /><br />
60   Click here if you like this site
61   <input type = "checkbox" name = "like" onfocus =
62     "helpText(2)" onblur = "helpText(6)" /><hr />
63
64   Any comments?<br />
65   <textarea name = "comments" rows = "5" cols = "45"
66     onfocus = "helpText(3)" onblur = "helpText(6)">
67 </textarea><br />
68   <input type = "submit" value = "Submit" onfocus =
69     "helpText(4)" onblur = "helpText(6)" />
70   <input type = "reset" value = "Reset" onfocus =
71     "helpText(5)" onblur = "helpText(6)" />
72
73   <textarea name = "helpBox" style = "position: absolute;
74     right:0; top: 0" rows = "4" cols = "45">
75   This textarea provides context-sensitive help. Click on
```

onsubmitreset.html  
(3 of 4)

```
76     any input field or use the Tab key to get more
77     information about the input field.</textarea>
78     </form>
79
80 </body>
81 </html>
```

**onsubmitreset.html**  
**(4 of 4)**





## 14.9 Event Bubbling

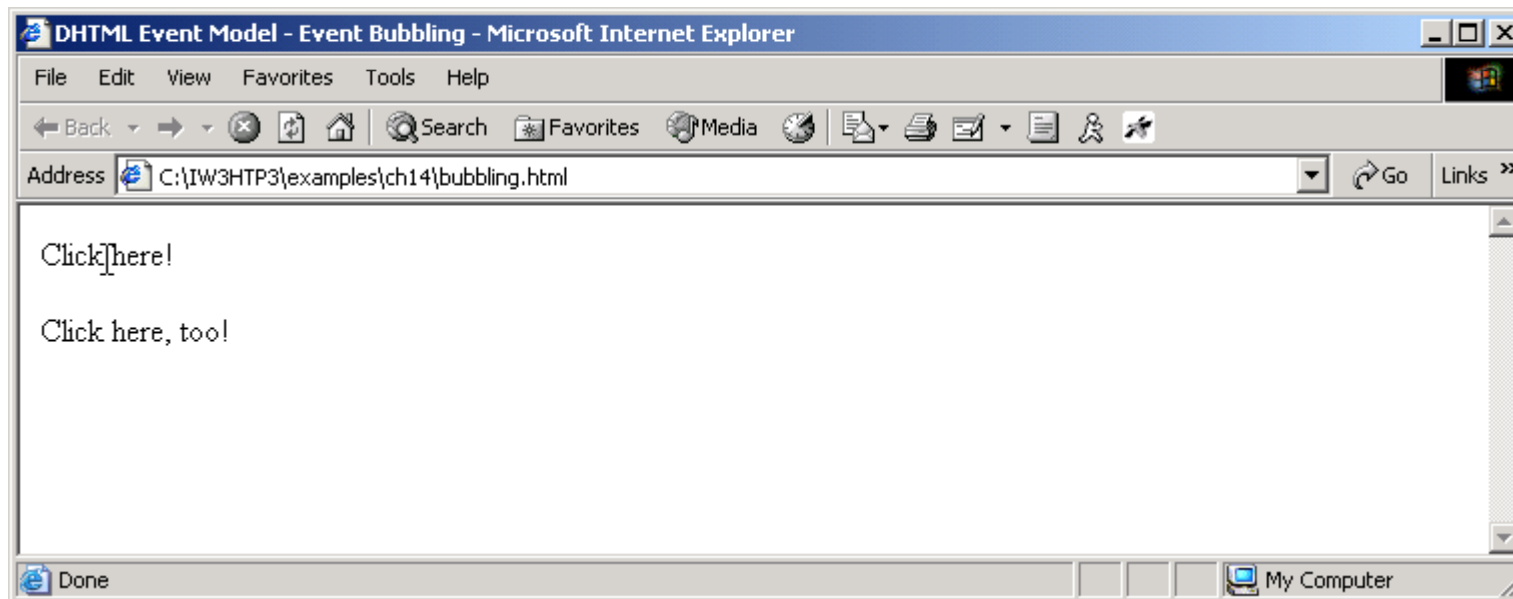
- Crucial part of the event model
- Process whereby events fired in child elements “bubble” up to their parent elements

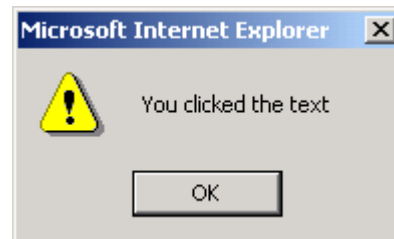
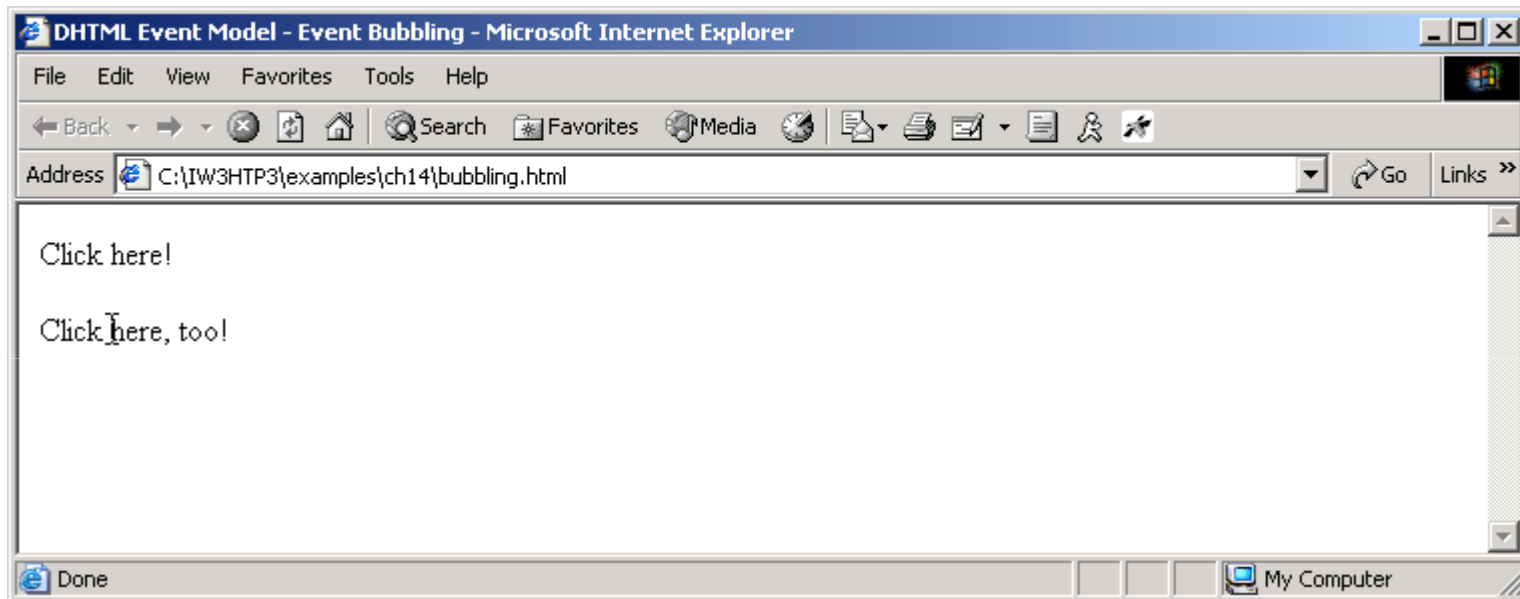
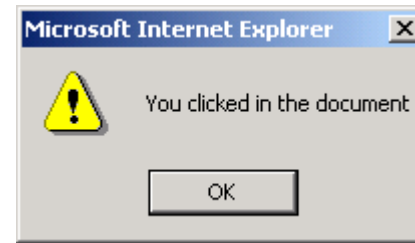
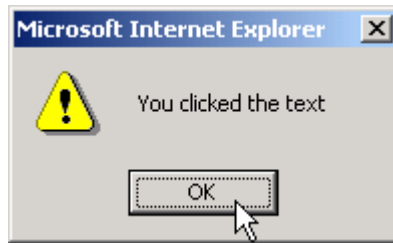
```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig 14.9: bubbling.html -->
6 <!-- Disabling event bubbling -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10     <title>DHTML Event Model - Event Bubbling</title>
11
12     <script type = "text/javascript">
13       <!--
14       function documentClick()
15       {
16         alert( "You clicked in the document" );
17       }
18
19       function paragraphClick( value )
20       {
21         alert( "You clicked the text" );
22
23         if ( value )
24           event.cancelBubble = true;
25       }
```

**bubbling.html**  
**(1 of 2)**

```
26
27     document.onclick = documentClick;
28     // -->
29 </script>
30 </head>
31
32 <body>
33
34     <p onclick = "paragraphClick( false )">Click here!</p>
35     <p onclick = "paragraphClick( true )">Click here, too!</p>
36 </body>
37 </html>
```

**bubbling.html**  
**(2 of 2)**





## 14.10 More DHTML Events

- Remaining DHTML events and their descriptions

## 14.10 More DHTML Events

Event	Description
<i>Clipboard events</i>	
onbeforecut	Fires before a selection is cut to the clipboard.
onbeforecopy	Fires before a selection is copied to the clipboard.
onbeforepaste	Fires before a selection is pasted from the clipboard.
oncopy	Fires when a selection is copied to the clipboard.
oncut	Fires when a selection is cut to the clipboard.
onabort	Fires if image transfer has been interrupted by user.
onpaste	Fires when a selection is pasted from the clipboard.
<i>Data binding events</i>	
onafterupdate	Fires immediately after a databound object has been updated.
onbeforeupdate	Fires before a data source is updated.
oncellchange	Fires when a data source has changed.
ondataavailable	Fires when new data from a data source become available.
ondatachanged	Fires when content at a data source has changed.
ondatasetcomplete	Fires when transfer of data from the data source has completed.
onerrorupdate	Fires if an error occurs while updating a data field.
onrowenter	Fires when a new row of data from the data source is available.
onrowexit	Fires when a row of data from the data source has just finished.
onrowsdelete	Fires when a row of data from the data source is deleted.
onrowsinserted	Fires when a row of data from the data source is inserted.
Fig. 14.10 Dynamic HTML events.	

## 14.10 More DHTML Events

Event	Description
<i>Keyboard events</i>	
onhelp	Fires when the user initiates help (i.e., by pressing the <i>F1</i> key).
onkeydown	Fires when the user pushes down a key.
onkeypress	Fires when the user presses a key.
onkeyup	Fires when the user ends a key press.
<i>Marquee events</i>	
onbounce	Fires when a scrolling <b>marquee</b> bounces back in the other direction.
onfinish	Fires when a <b>marquee</b> finishes its scrolling.
onstart	Fires when a <b>marquee</b> begins a new loop.
<i>Mouse events</i>	
oncontextmenu	Fires when the context menu is shown (right-click).
ondblclick	Fires when the mouse is double clicked.
ondrag	Fires during a mouse drag.
ondragend	Fires when a mouse drag ends.
ondragenter	Fires when something is dragged onto an area.
ondragleave	Fires when something is dragged out of an area.
ondragover	Fires when a drag is held over an area.
ondragstart	Fires when a mouse drag begins.
ondrop	Fires when a mouse button is released over a valid target during a drag.
onmousedown	Fires when a mouse button is pressed down.

Fig. 14.10 Dynamic HTML events.

## 14.10 More DHTML Events

Event	Description
onmouseup	Fires when a mouse button is released.
<i>Miscellaneous events</i>	
onafterprint	Fires immediately after the document prints.
onbeforeeditfocus	Fires before an element gains focus for editing.
onbeforeprint	Fires before a document is printed.
onbeforeunload	Fires before a document is unloaded (i.e., the window was closed or a link was clicked).
onchange	Fires when a new choice is made in a <code>select</code> element, or when a text input is changed and the element loses focus.
onfilterchange	Fires when a filter changes properties or finishes a transition (see Chapter 15, <i>Dynamic HTML: Filters and Transitions</i> ).
onlosecapture	Fires when the <code>releaseCapture</code> method is invoked.
onpropertychange	Fires when the property of an object is changed.
onreadystatechange	Fires when the <code>readyState</code> property of an element changes.
onreset	Fires when a form resets (i.e., the user clicks a reset button).
onresize	Fires when the size of an object changes (i.e., the user resizes a window or frame).
onscroll	Fires when a window or frame is scrolled.
onselect	Fires when a text selection begins (applies to <code>input</code> or <code>textarea</code> ).
onselectstart	Fires when the object is selected.
onstop	Fires when the user stops loading the object.
onunload	Fires when a page is about to unload.

Fig. 14.10 Dynamic HTML events.