COMP11008 Web Development

Section 13. JavaScript: Event Handling

13.9. Exercises

Exercise 13.1.

Add an erase feature to the drawing program in Figure 13.3. Try setting the background color of the table cell over which the mouse moved to white when the Alt key is pressed.

Demonstration

```
<!DOCTYPE html>

<!-- Exercise 13.1: exercise-13-01.html -->
<html>
   <head>
      <meta charset = "utf-8">
      <title>Web Development</title>
      <link rel = "stylesheet" type = "text/css" href = "style.css">
      <script src = "exercise-13-01.js"></script>
   </head>
   <body>
      <table id = "canvas">
         <caption>Hold <em>Ctrl</em> (or <em>Control</em>) to draw blue.
            Hold <em>Shift</em> to draw red. Hold <em>Alt</em> to erase.</caption>
         <tbody id = "tablebody"></tbody>
      </table>
   </body>
</html>
```

exercise-13-01.html

```
/* style.css */
#canvas { width: 400px;
   border: 1px solid #999999;
   border-collapse: collapse }
td { width: 4px;
   height: 4px;
   margin: 0px;
   padding: 0px; }
.blue { background-color: blue; }
.red { background-color: red; }
.white { background-color: white; }
```

```
1.  // Exercise 13.1: exercise-13-01.js
2.  // A simple drawing program
3.
4.  //initialization function to insert cells into the table
5.  function createCanvas()
6.  {
7.     var side = 100;
8.     var tbody = document.getElementById( "tablebody" );
9.
10.    for ( var i = 0; i < side; ++i )
11.    {
12.       var row = document.createElement( "tr" );
13.
14.       for ( var j = 0; j < side; ++j )
15.       {
16.          var cell = document.createElement( "td" );
17.          row.appendChild( cell );
18.       } // end for
19.
20.       tbody.appendChild( row );
21.    } // end for
22.
23.    // register mousemove listener for the table
24.    document.getElementById( "canvas" ).addEventListener(
25.       "mousemove", processMouseMove, false );
26. } // end function createCanvas
27.
28. // processes the onmousemove event
29. function processMouseMove( e )
30. {
31.    if ( e.target.tagName.toLowerCase() == "td" )
32.    {
33.       // turn the cell blue if the Ctrl key is pressed
34.       if ( e.ctrlKey )
35.       {
36.          e.target.setAttribute( "class", "blue" );
37.       } // end if
38.
39.       // turn the cell red if the Shift key is pressed
40.       if ( e.shiftKey )
41.       {
42.          e.target.setAttribute( "class", "red" );
43.       } // end if
44.
45.       if ( e.altKey )
46.       {
47.          e.target.setAttribute( "class", "white" );
48.       } // end if
49.    } // end if
50. } // end function processMouseMove
51.
52. window.addEventListener( "load", createCanvas, false );
```

Exercise 13.2.

Add a button to your program from Exercise 3 to erase the entire drawing window.

Demonstration

```
<!DOCTYPE html>

<!-- Exercise 13.2: exercise-13-02.html -->
<html>
   <head>
      <meta charset = "utf-8">
      <title>Web Development</title>
      <link rel = "stylesheet" type = "text/css" href = "style.css">
      <script src = "exercise-13-02.js"></script>
   </head>
   <body>
      <table id = "canvas">
         <caption>Hold <em>Ctrl</em> (or <em>Control</em>) to draw blue.
            Hold <em>Shift</em> to draw red. Hold <em>Alt</em> to erase.</caption>
         <tbody id = "tablebody"></tbody>
      </table>
      <form action = "">
         <p><input id = "clearButton" type = "button" value = "Clear"></p>
      </form>
   </body>
</html>
```

exercise-13-02.html

```
/* style.css */
#canvas { width: 400px;
   border: 1px solid #999999;
   border-collapse: collapse }
td{ width: 4px;
   height: 4px;
   margin: 0px;
   padding: 0px; }
.blue { background-color: blue; }
.red { background-color: red; }
.white  { background-color: white; }
```

```
// Exercise 13.2: exercise-13-02.js
// A simple drawing program

//initialization function to insert cells into the table
function createCanvas()
{
   var side = 100;
```

```
   var tbody = document.getElementById( "tablebody" );

   for ( var i = 0; i < side; ++i )
   {
      var row = document.createElement( "tr" );

      for ( var j = 0; j < side; ++j )
      {
         var cell = document.createElement( "td" );
         row.appendChild( cell );
      } // end for

      tbody.appendChild( row );
   } // end for

   // register mousemove listener for the table
   document.getElementById( "canvas" ).addEventListener(
      "mousemove", processMouseMove, false );

   // register Clear button listener
   document.getElementById( "clearButton" ).addEventListener(
      "click", clearCanvas, false );
} // end function createCanvas

// processes the onmousemove event
function processMouseMove( e )
{
   if ( e.target.tagName.toLowerCase() == "td" )
   {
      // turn the cell blue if the Ctrl key is pressed
      if ( e.ctrlKey )
      {
         e.target.setAttribute( "class", "blue" );
      } // end if

      // turn the cell red if the Shift key is pressed
      if ( e.shiftKey )
      {
         e.target.setAttribute( "class", "red" );
      } // end if

      if ( e.altKey )
      {
         e.target.setAttribute( "class", "white" );
      } // end if
   } // end if
} // end function processMouseMove

function clearCanvas()
{
   var tablebody = document.getElementById( "tablebody" );
   var tableParent = tablebody.parentNode;
   tableParent.removeChild( tablebody );
   var newTableBody = document.createElement( "tbody" );
   newTableBody.id = "tablebody";
   tableParent.appendChild( newTableBody );
```

```
64.     createCanvas();
65. } // end function clearCanvas
66.
67. window.addEventListener( "load", createCanvas, false );
```

Exercise 13.3.

Write a function that responds to a click anywhere on the page by displaying an alert dialog. Display the event name if the user held Shift during the mouse click. Display the element name that triggered the event if the user held Ctrl during the mouse click.

Demonstration

```
1.  <!DOCTYPE html>
2.
3.  <!-- Exercise 13.3: exercise-13-03.html -->
4.  <html>
5.     <head>
6.        <meta charset = "utf-8">
7.        <title>Web Development</title>
8.        <script src = "exercise-13-03.js"></script>
9.     </head>
10.    <body>
11.       <p>Wikipedia, the free encyclopedia that anyone can edit.
12.          <img src = "images/wikipedia.jpg" alt = "wikipedia logo"></p>
13.       <p><a href =
      "http://en.wikipedia.org/wiki/Main_Page">http://en.wikipedia.org/wiki/Main_Page</a>
      </p>
14.    </body>
15. </html>
```

exercise-13-03.html

```
1.  // Exercise 13.3: exercise-13-03.js
2.  function start()
3.  {
4.     document.addEventListener( "click", processClick, false );
5.  } // end function start
6.
7.  function processClick( e )
8.  {
9.     if ( e.shiftKey )
10.       alert( e.type );
11.
12.    if ( e.ctrlKey )
13.       alert( e.target.tagName );
14. } // end function processClick
15.
16. window.addEventListener( "load", start, false );
```

Exercise 13.4.

Use CSS absolute positioning, mousedown, mousemove, mouseup and the clientX/clientY properties of the event object to create a program that allows you to drag and drop an image. When the user clicks the image, it should follow the cursor until the mouse button is released.

Demonstration

```
<!DOCTYPE html>

<!-- Exercise 13.4: exercise-13-04.html -->
<html>
   <head>
      <meta charset = "utf-8">
      <title>Web Development</title>
      <link rel = "stylesheet" type = "text/css" href = "style.css">
      <script src = "exercise-13-04.js"></script>
   </head>
   <body>
      <div class = "draggable" id = "wiki"></div>
   </body>
</html>
```

exercise-13-04.html

```
/* style.css */
div#wiki{ position: absolute;
   top: 0;
   left: 0;
   height: 200px;
   width: 226px;
   background-image: url("images/wikipedia.jpg") }
```

```
// Exercise 13.4: exercise-13-04.js
var theImg;
var mouseX;
var mouseY;
var zIndex;

function start()
{
   document.addEventListener( "mousedown", startDrag, false );
   document.addEventListener( "mouseup", stopDrag, false );
   theImg = document.getElementById( "wiki" );
   zIndex = 0;
   mouseX = 0;
   mouseY = 0;
} // end function start

function startDrag( e )
{
```

```
    mouseX = e.clientX - theImg.offsetLeft;
    mouseY = e.clientY - theImg.offsetTop;
    if ( e.target == theImg )
    {
        document.addEventListener( "mousemove", updateImageLocation, false
);
    } // end if
} // end function startDrag

function stopDrag()
{
    document.removeEventListener( "mousemove", updateImageLocation, false
);
} // end function stopDrag

function updateImageLocation( e )
{
    theImg.setAttribute( "style", "top: " + ( e.clientY - mouseY ) + "px; " +
        "left: " + ( e.clientX - mouseX ) + "px" );
} // end function updateImageLocation

window.addEventListener( "load", start, false );
```

< Back | Index | Next >