transentia

Developing with
JavaScript Technologies

MSIE Colour Selector

Exercise: Handling Mouse Clicks

Introduction
This exercise will show how to distinguish and handle mouse clicks in Internet Explorer.

The Exercise
In this exercise, you will .

Set Up
Make a new directory for this exercise. Call this directory **cs**:

```
C:\> mkdir cs
C:\> cd cs
```

All the files that you subsequently create as you do this exercise should be contained in this directory.

Create the HTML/Script Source
Create the following in a file called **ColorSelect.html**:

```
<html>
  <head>
    <script language='JavaScript'>
      // Define initial states
      var myOldSrcElement;
      var myRed   = "00";
      var myGreen = "00";
      var myBlue  = "00";

      // Handle clicks in the table
      function clicked()
        {
        var myValue;
        var myColumn;
        var myRow;

        myEvent = event;

        // Indicate what was clicked on
        if(myOldSrcElement)
          myOldSrcElement.bgColor = "white";

        // Work out click column and row
        myColumn = (myEvent.srcElement.cellIndex) + 1;
        myRow    = myEvent.srcElement.parentElement.rowIndex;

        // Set color intensity
```

```
        switch(myRow)
           {
           case 1: myValue = "FF"; break;
           case 2: myValue = "CC"; break;
           case 3: myValue = "99"; break;
           case 4: myValue = "66"; break;
           case 5: myValue = "33"; break;
           case 6: myValue = "00"; break;
           default: myValue = ""; break;
           }

        // Set colour to change
        if(myValue != "")
           {
           myEvent.srcElement.bgColor = "red";

           switch(myColumn)
              {
              case 1: myRed = myValue; break;
              case 2: myGreen = myValue; break;
              case 3: myBlue = myValue; break;
              }

           myColour = "#" + myRed + myGreen + myBlue;

           myTarget = document.getElementById("COLORCELL");
           myTarget.bgColor = myColour;

           myResult = document.getElementById("OUTPUT");
           myResult.innerHTML = myColour;

           myOldSrcElement = myEvent.srcElement;
           }
        }
    </script>
    <style>
      td {text-align: right; }
    </style>
</head>
<body>
   <table border='0' width="75%" align="center">
     <tr>
       <td valign="top" align="right">Select color:</td>
       <td>
         <table border='1' onClick='clicked()'>
           <tr>
             <th width="33%">Red</th>
             <th width="33%">Green</th>
             <th width="33%">Blue</th>
           </tr>
           <tr><td>100</td> <td>100</td> <td>100</td></tr>
           <tr><td>80</td> <td>80</td> <td>80</td></tr>
           <tr><td>60</td> <td>60</td> <td>60</td></tr>
```

```
            <tr><td>40</td> <td>40</td> <td>40</td></tr>
            <tr><td>20</td> <td>20</td> <td>20</td></tr>
            <tr><td>0</td> <td>0</td> <td>0</td></tr>
          </table>
        </td>
      <td id="COLORCELL" width="50%" valign="middle">
        <p align="center"><span id="OUTPUT">#FFFFFF</span></p>
      </td>
      </tr>
    </table>
  </body>
</html>
```

Points of interest

- This simple example works by detecting DHTML mousedown events and correlating the position of the mouse to the row/column in the enclosing table and thence to the correct colour to use.

- Notice the use of the getElementById() method to isolate a specific element.

Time to Play!

Double-click the ColourSelect.html file to open it in InternetExplorer.

Simply click within the values cells in the table and note how the area to the right of the table changes colour appropriately. You may also note how the last clicked cell is highlighted.