

```

1  <!DOCTYPE html>
2
3  <html lang="en">
4  <head>
5      <title>Dice Objects Exercise</title>
6      <meta charset="utf-8">
7      <style type="text/css">
8          /* #tableBody, #tableHeadings, #diceTable, #showDice { visibility: hidden; } */
9      </style>
10     <script type="text/javascript">
11         /* Class Definition(s) */
12         /*
13          * Example of a Dice class:
14          * This class has several properties:
15          *   numberOfFaces: which must be a positive integer (> 1 ) to be interesting.
16          *   count : a non-negative integer that records how many times this particular
17          *   Dice has been "rolled."
18          *   face : current face, a non-negative integer 1 .. numberOfFaces.
19          *
20          * In order to use this function, say
21          *   var myDice = new Dice( 6 ); // makes a six-faced dice.
22          */
23         function Dice( numFaces ) {
24             this.numberOfFaces = numFaces;
25             this.count=0;
26             this.face = Math.floor( Math.random() * this.numberOfFaces ) + 1;
27             this.toString = function () {
28                 return "[Showing: " + this.face + " : rolled " + this.count + " time(s)]";
29             };
30             this.roll = function () {
31                 // maybe change the current face ...
32                 this.face= Math.floor( Math.random() * this.numberOfFaces ) + 1;
33                 // increment the number of times that this Dice has been "rolled"
34                 this.count++;
35             };
36             this.reset = function () {
37                 this.count=0;
38             };
39         }
40         /* ____ end class definitions ____ */
41         /* Variables, constants, functions, procedures and event-handling code */
42         /* We depend upon two "global" variables to maintain state between
43          * events.
44          */
45         var numberOfDice=0;
46         var diceArray=[];
47         /* _____end constants, variables section_____ */
48
49         /* Functions, procedures, etc. */
50         /* event handlers */
51         function inputNumberOfDice() {
52             numberOfDice = parseInt( document.getElementById( "numberOfDice" ).value );
53             document.getElementById( "showDice" ).innerHTML= "Show the results of creating/rolling " + numberOfDice + " dice";
54             document.getElementById( "showDice" ).style.visibility="visible";
55         }
56         /*
57          *Tricky: this function actually populates the global variable,
58          *diceArray with the required number of Dice with the specified number of
59          *faces.
60          */
61         function inputDiceFaces () {
62             var numberOfFaces = parseInt( document.getElementById( "numberOfFaces" ).value );
63             diceArray = Array( numberOfDice ); // notice: I do NOT create a local var here!
64             for( var index=0; index < numberOfDice; index++ ) {
65                 diceArray[ index ] = new Dice( numberOfFaces );
66             }
67         }
68         /*
69          * This procedure creates the <thead>, i.e., the "headers" for the
70          * dice table. It does this by examining the length of the diceTable

```

```

71      * and generating a label, such as Dice #1, Dice#2, etc., for each Dice.
72      */
73      function generateTableHeadings () {
74          var headings = "<tr>";
75          for( var diceNumber=0; diceNumber < diceArray.length; diceNumber++ ) {
76              headings += "<th>Dice #" + (diceNumber + 1) + "</th>";
77          }
78          return headings + "</tr>";
79      }
80
81      /*
82      * This procedure populates the Dice Table based upon how many
83      * Dice are found in the diceArray ...
84      */
85      function generateTableRows() {
86          /* for each Dice object found in the diceTable,
87           * we need to generate some text that will fill in the
88           * <td> element for that particular Dice.
89           * Note that the 'headings' for each Dice is handled by
90           * the generateTableHeadings procedure, above.
91           */
92          var row="<tr>";
93          for( var index=0; index < diceArray.length; index++ ) {
94              row += "<td>" + diceArray[ index ].toString() + "</td>";
95          }
96          return row + "</tr>";
97      }
98      function generateDiceTable () {
99          document.getElementById("tableHeadings").innerHTML = generateTableHeadings();
100         document.getElementById("tableBody").innerHTML = generateTableRows();
101         document.getElementById("diceTable").style.visibility="visible";
102     }
103     /*
104     * Event Handler for createBtn---on click run this.
105     */
106     function showTable() {
107         document.getElementById("showDice").style.visibility="visible";
108         generateDiceTable();
109     }
110     /*
111     * Event Handler: rolls every dice in the table, and then regenerates the table to
112     * show the changes.
113     */
114     function rollAll() {
115         for( var i = 0; i < diceArray.length; i++ ) {
116             diceArray[ i ].roll();
117         }
118         document.getElementById("tableBody").innerHTML = generateTableRows();
119     }
120     /*
121     * Event Handler: roll one pops up a window asking which one (for now),
122     * I had to think of a reason for using a prompt somewhere?
123     */
124     function rollOne () {
125         var whichOne = parseInt( window.prompt("Type a number between 1 and " + diceArray.length + ": "));
126         while( whichOne < 1 || whichOne > diceArray.length ) {
127             whichOne = parseInt( window.prompt("Try again: a number between 1 and " + diceArray.length + ": "));
128         }
129         /* note the number that the user provides needs to be translated into a valid
130          * array index. Because arrays are 0-indexed, this means subtracting one.
131          */
132         whichOne--;
133         diceArray[ whichOne ].roll();
134         document.getElementById("tableBody").innerHTML = generateTableRows();
135     }
136     </script>
137
138 </head>
139
140 <body>
141
142 <h1>Playing with Dice</h1>

```

```
143 <p>How many dice shall we create? <input type="number" min="2" max="5" id="numberOfDice" onchange="inputNumberOfDice();" > </p>
144 <p>How many faces should each Dice have? <input type="number" min="2" max="12" id="numberOfFaces" onchange="inputDiceFaces();" > </p>
145 <br>
146 <button id="createBtn" onclick="showTable();">Create Them!</button>
147 <br>
148 <h2 id="showDice"></h2>
149 <table id="diceTable">
150   <thead id = "tableHeadings">
151     <!-- Heading text will go here -->
152   </thead>
153   <tbody id="tableBody">
154     <!-- Body text will go here -->
155   </tbody>
156 </table>
157 <br>
158 <br>
159 <button id="rollAll" onclick="rollAll();">Roll All of the Dice</button>&nbsp;
160 <button id="rollOne" onclick="rollOne();">Choose a dice (by number) to roll</button>
161 </body>
162 </html>
```