Динамическое создание таблиц

В настоящем разделе рассмотрим динамические способы создания таблиц, в которых элемент <TABLE>, а также элементы строк и ячеек создаются инструкциями сценария.

Построение таблицы

При программировании некоторых интерактивных игр, создании пользовательских диалогов часто возникает задача создания таблиц. Динамическую таблицу можно ввести на страницу путем генерации нужного количества тегов <TR> и <TD>, оформив эту процедуру с помощью операторов цикла for. Ниже записан код страницы, на которой генерируется таблица заданного размера mxn (m – число строк, n– число столбцов):

```
<HTML>
 <HEAD>
   <TITLE>Построение таблицы</TITLE>
   <STYLE type="text/css">
     #tabElement TD{width:60px; height:60px}
   </STYLE>
   <SCRIPT language="javascript">
   //Функция построения таблицы mxn в документе
   function drawTable(m,n){
    var tab="<TABLE border id=tabElement>";
    //Задание тегов строк
    for (var indR=0; indR<m; indR++) {
     tab+="<TR>";
     //Задание тегов ячеек
     for (var indC=0; indC<n; indC++)
       tab+="<TD> </TD>";
       tab+="</TR>";
       }
     tab+="</TABLE>";
   //Возвращается значение переменной,
   //описывающей теговую структуру таблицы
   return tab;
   }
  </SCRIPT>
 </HEAD>
 <BODY>
  <CENTER>
  <H2>Построение таблицы методом write()</H2>
  <SCRIPT language="javascript">
    document.write(drawTable(4,5));
  </SCRIPT>
  </CENTER>
 </BODY>
</HTML>
```

Внешний вид таблицы 4x5, построенной описанным способом, показан на рис. 1. Фактическое число строк и столбцов в этом коде определяется значениями аргументов в

методе `document.write(drawTable())`. Также можно задавать размер таблицы динамически с помощью диалоговых окон запроса `prompt`.

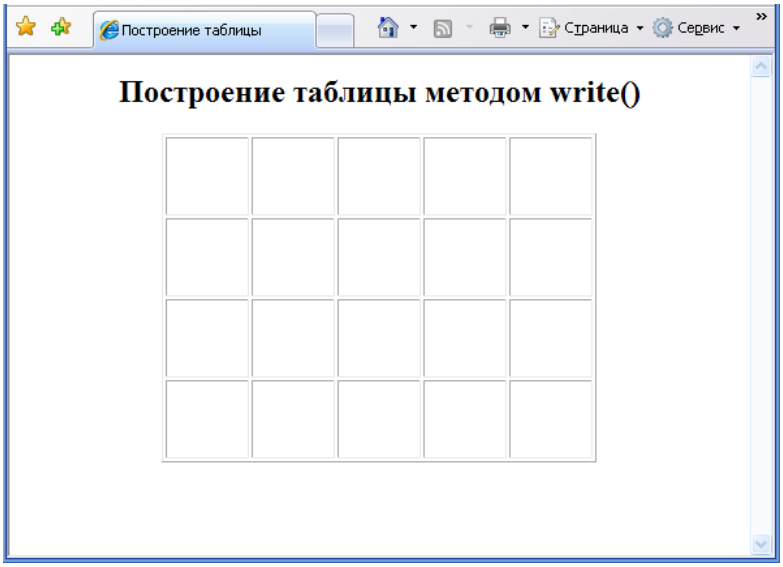

Рис.1. Пример таблицы, построенной с помощью сценария

Свойства и методы элемента <TABLE>

Каждый элемент `<TABLE>` в HTML-документе является объектом, который обладает наборами свойств и методов. Нам будут интересны те из них, которые можно применить при построении таблиц.

Приведем основные сведения о массивах, образованных компонентами таблиц. Строки таблицы (например, `myTable`) представляются массивом `rows`, который имеет размер, определяемый свойством `length`:

```
var nrows = document.all("myTable").rows.length
```

Здесь переменная `nrows` возвращает количество строк в таблице. Индекс массива `rows[k]` пробегает значения от 0 до (`nrows-1`).

Каждой строке `rows[k]` соответствует массив ячеек `cells`, который, в свою очередь обладает размером

```
var ncells = document.all("myTable").rows[k].cells.length
```

Эта переменная возвращает количество ячеек в строке `k`. Заметим, что даже в случае прямоугольной таблицы, где строки имеют одинаковую длину, необходимо указывать в определении свойства `rows[k].cells.length` индекс строки. Индекс массива `cells[m]` пробегает значения от 0 до (`ncells-1`). Элемент вида

```
myTable.rows[k].cells[m]
```

представляет ячейку с координатами `[k,m]`, то есть m-ую ячейку k-ой строки.

Индекс строки возвращается свойством `rowIndex`, а индекс ячейки в строке – свойством `cellIndex`. Например, фрагмент кода

```
var z = myTable.rows(1);
alert(z.rowIndex);
var y = = myTable.rows(1).cells(4);
alert(y.cellIndex);
```

выведет на экран сначала сообщение «1», а затем – «4».

Из методов элемента `<TABLE>` назовем те, которые могут быть использованы при динамической генерации либо редактировании таблицы.

- `insertRow(k)` – вставляет в таблицу новую строку с номером `k`. При этом новый элемент автоматически включается в массив `rows`;
- `insertCell(m)` – вставляет ячейку с номером m таким образом, что ячейки, расположенные справа от m, смещаются вправо. Вставленная ячейка сразу включается в массив `cells`;
- `deleteRow(k)` – удаляет из таблицы строку `k`;
- `deleteCell(m)` – удаляет из таблицы ячейку m. При этом ячейки, расположенные справа от m, смещаются влево.

Рассмотрим, как пример, вставку в существующую таблицу новой строки с номером 2. Для этого в сценарий необходимо сначала записать инструкцию:

```
myTable.insertRow(2);
```

Новую строку нужно заполнить ячейками, а в ячейки – ввести содержание. Это можно сделать, например, с помощью следующих инструкций:

```
for(var j=0; j<nrows; j++) {
   x=myTable.rows(2).insertCell(j);
   x.innerText=" ";
}
```

Здесь `nrows` – количество ячеек в строке (в общем случае оно может не совпадать с количеством столбцов в таблице). Свойству динамического содержания `innerText` присвоен обычный пробел, чтобы ячейка вообще отображалась на экране.

Пользуясь методами `insertRow()` и `insertCell()` нетрудно реализовать динамическую генерацию всей таблицы.

Заполнение таблицы данными

Рассмотрим задачу создания таблицы и заполнения ее простым содержимым (крестиками, галочками, единицами и т.д.), причем содержимое задается одиночным щелчком мыши. Пусть нужно заполнить квадратную таблицу крестиками (аналогично тому, как это делается при игре в «Морской бой»). Приведем листинг соответствующей Web-страницы.

```
<HTML>
  <HEAD>
   <TITLE>Карта выстрелов</TITLE>
   <STYLE type="text/css">
     TD {font-weight:bold}
     #tabElement TD{width:30px; height:30px;
     text-align:center; cursor:hand; font-size:12pt}
   </STYLE>
   <SCRIPT language="javascript">
   //Определение функции для построения таблицы
   function drawTable(){
   var tab="<TABLE border id=tabElement onClick='doCross();'>";
   for (var indR=0; indR<10; indR++) {
     tab+="<TR>";
     for (var indC=0; indC<10; indC++)
       tab+="<TD> </TD>";
       tab+="</TR>";
   }
   tab+="</TABLE>";
   return tab;
   }
   //Ввод в ячейку символа
   function doCross(){
   //проверка события щелчка в ячейке таблицы
   if("TD"==event.srcElement.tagName){
     var shot=event.srcElement;
     //Проверка того, что ячейка пуста
     if(" "==shot.innerHTML)
     shot.innerText="X";
     }
   }
   </SCRIPT>
  </HEAD>
 <BODY>
  <CENTER>
  <H2>Морской бой</H2>
  <H3>Карта выстрелов</H3>
  <SCRIPT language="javascript">
   document.write(drawTable());
  </SCRIPT>
  </CENTER>
 </BODY>
</HTML>
```

С помощью щелчков мыши вы можете заполнить ячейки таблицы, как показано на рис. 2.

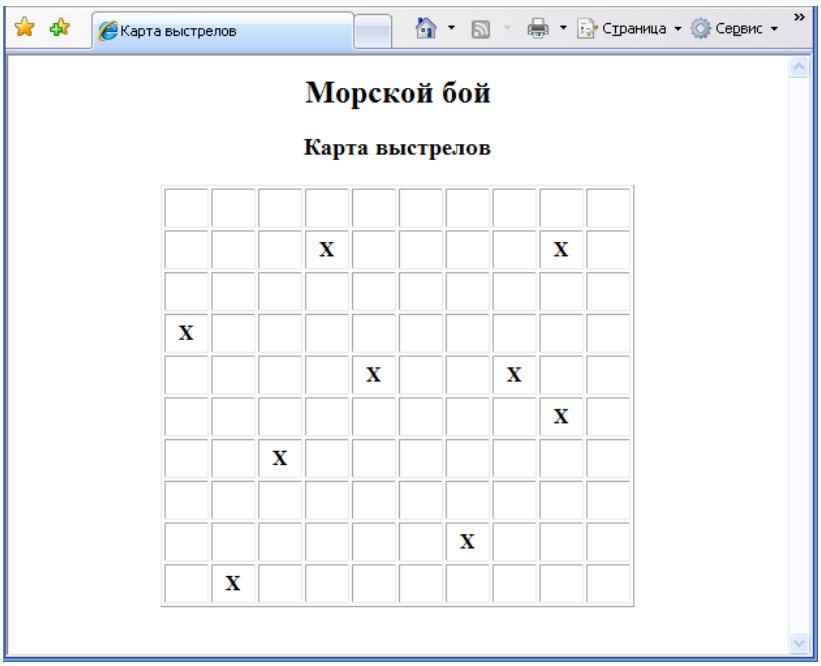

Рис.2. Пример заполнения таблицы с помощью щелчков мыши

Содержимое ячейки с координатами `(i,j)` будет возвращаться свойством:

```
rows[i].cell[j].innerText
```

где `j` – номер ячейки в строке, который совпадает с номером столбцы. Например, если вы хотите подсчитать общее количество ячеек, заполненных символом «X», можете ввести в сценарий функцию:

```
function quant(){
  var n=0;
  for (i=0; i<10; i++){
    x=tabElement.rows(i);
    for (j=0; j<10; j++){
      if(x/cells[j].innerText=="X")
      n+=1;
      }
    }
  document.write("количество заполненных ячеек: "+n);
   }
```

Для вызова этой функции можно предусмотреть кнопку, которая нажимается после заполнения таблицы.