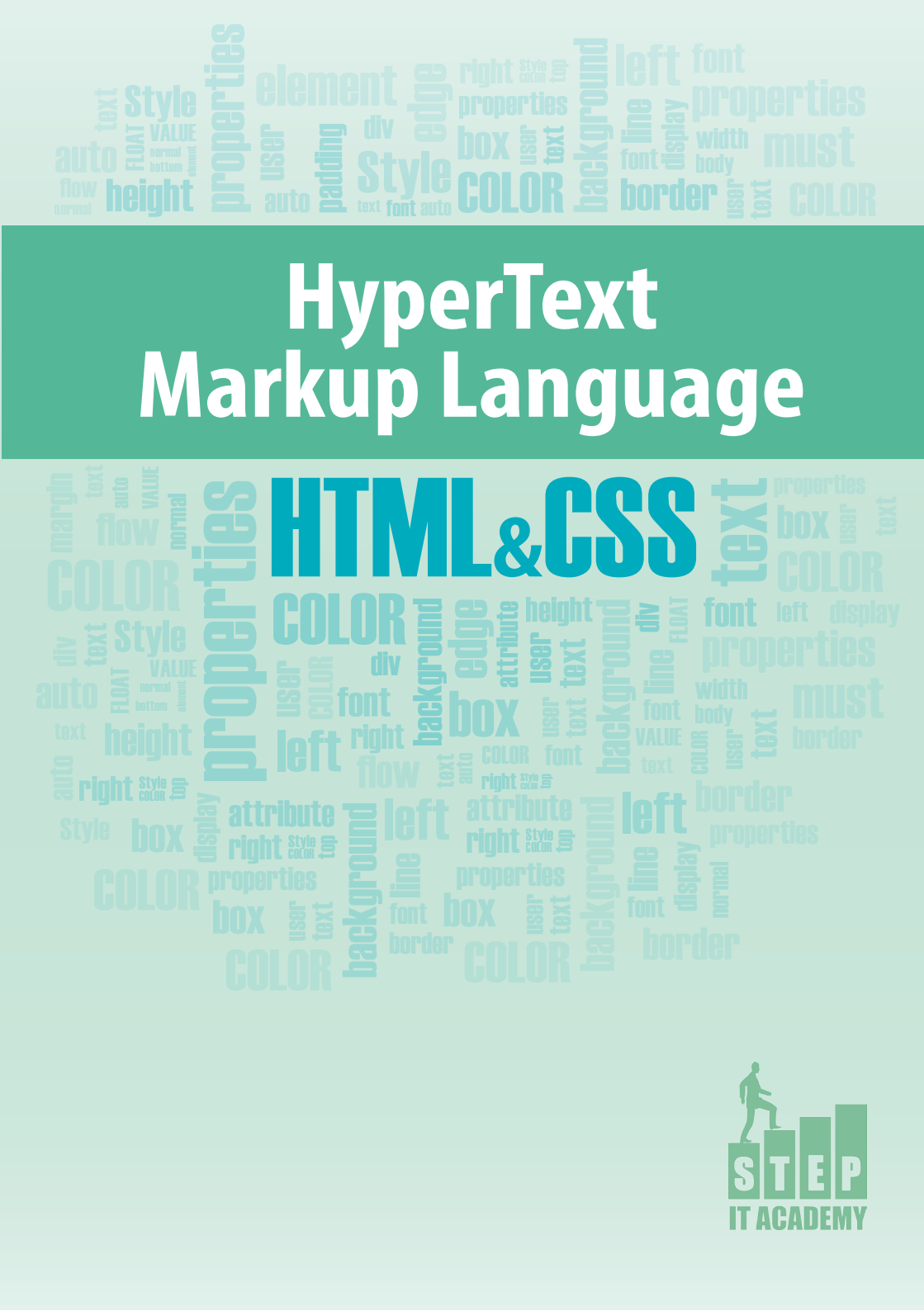
HyperText Markup Language



STEP IT ACADEMY

Unit 4

Creating tables

Contents

Creating tables

Basic elements

Tabular representation of information is used in those cases when it is necessary to display a large amount of data on the page, characterized by the same type of data set. Examples of such data can be a CD library, or a book catalog.

Tables are built using the following types of elements:

- *table* — the actual container that contains the table. This element can only contain row grouping elements (thead, tfoot and tbody).
- *tbody* — the body of the table. Within this element, only the rows of the table should be placed.
- *tr* — serves for creating table rows. The row can contain table data items (td) or header elements (th).
- *td* is the data element of the table.
- *th* is the element of the table header.

This is not all the elements of the tables, however, we will analyze everything in order. So, below is the code for the simplest table:

```
<table>
    <tbody>
        <tr>
          <td>ASP.NET</td>
          <td>Matthew MacDonald</td>
          <td>2003</td>
      </tr>
      <tr>
          <td>XHTML</td>
```

```
            <td>Donald MacMatthew </td>
            <td>2004</td>
        </tr>
        <tr>
            <td>Using PHP 5</td>
            <td>Mac Matthew</td>
            <td>2005</td>
        </tr>
        ...
    </tbody>
</table>
```

As you can see, the resulting table does not look exactly as we would like. Habitual to us of tables have at least frames! To achieve a "normal" appearance, the attributes of the table element will help us:

- *border* — this attribute determines the presence and thickness of the table frame. If border has a value greater than 0, then the table, as well as each of its non-empty cells, will acquire a frame. In addition, the thickness of the outer frame of the table is determined by the assigned value.
- *width* — the width of the entire table. In this case, the width of each column of the table is selected automatically based on the length of the values of the cells that make up this column. The width of the table can be specified in pixels or in percent. In the latter case, the "%" sign is put immediately after the number.
- *cellspacing* — the size of the outer indent from the cell borders to adjacent cells or the table frame.
- *cellpadding* — the size of the inner indent from the cell borders to the content.

- *align* — alignment of the table. Canceled! To align the entire table, use SS.

In view of the above, we modify the code of the example 5.1.1 as follows:

```
<table border="1" width="500"cellspacing="0"
       cellpadding="3">
    <tbody>
        <tr>
           <td>ASP.NET</td>
           <td>Matthew MacDonald</td>
           <td>2003</td>
        </tr>
        <tr>
           <td>XHTML</td>
           <td>Donald MacMatthew</td>
           <td>2004</td>
        </tr>
        <tr>
            <td>Using PHP 5</td>
            <td>Mac Matthew</td>
            <td>2005</td>
        </tr>
         ...
    </tbody>
</table>
```

Now we will equip our table with a decent "top". The labels for the columns of the table are placed on a separate line containing the th elements — the elements of the header:

```
<table border="1" width="500"cellspacing="0"
       cellpadding="3">
    <tbody>
        <tr>
```

```
            <th>Name</th>
            <th>Author</th>
            <th>Year of publication</th>
        </tr>
         ...
   </tbody>
</table>
```

In addition, you can select a title bar by assigning a color or an image to it as a background. For example:

```
<table border="1" width="500"cellspacing="0"
       cellpadding="3">
    <tbody>
        <tr style="background-image:url(bg01.jpg);">
        ...
    </tbody>
</table>
```

In fact, you can assign the background color or image, as well as other style properties (text color, font, style, ...) to the entire table, to its individual line and even to a single cell. To do this, add the style attribute to the corresponding element.

In addition to other features described, a table can be given a caption. This is implemented using the caption element, which is usually located immediately after the table opening tag:

```
<table border="1" width="500"cellspacing="0"
       cellpadding="3">
<caption>Web-master library</caption>
    <tbody>
        <tr style="background-image:url(bg01.jpg);">
```

```
        ...
    </tbody>
</table>
```

The caption is usually centered on the table. To change the alignment, use the text-align style rule.

More fine customization of the table appearance can be done using the attributes of the elements: <tr>, <td> and <th>.

Aligning table data

You can align table data for each specific cell, for the entire row, and for a row or column group (<tbody>, <thead>, <tfoot>, etc., see below). In addition, the data can be aligned horizontally and vertically. For horizontal alignment, the align attribute is used, which can take the following values:

- *left* — along the left side of the cell
- *center* — along the right side of the cell
- *right* — in the center of the cell
- *justify* — breadthwise of the cell.

For vertical alignment, the valign attribute is used. The values for this attribute can be:

- *top* — at the top of the cell
- *middle* — in the center of the cell
- *bottom* — at the bottom of the cell.

For whatever element you specify the align or valign attributes (<td>, <th>, <tr>, or group), alignment occurs relative to the boundaries of each cell. Specifying the alignment for the <td> element, we place only its contents accordingly.

By setting the alignment for the entire <tr> line, we set the alignment for all the cells in this row. Similarly, for groups of rows or columns.

Below is an example demonstrating the alignment of data in table cells:

```
...
<tr>
    <td>XHTML</td>
   <td align="center">Donald MacMatthew</td>
   <td align="right">2004</td>
</tr>
<tr>
   <td>Using PHP 5</td>
   <td align="center">Mac Matthew</td>
   <td align="right">2005</td> </tr>
...
```

Managing cell sizes

For <td> and <th> cells, you can specify dimensions using the width and height style properties:

- width: width;
- height: height;

In principle, these style properties can be used not only to specify the size of the cells in the tables. With their help, you can specify the size of any block element of the page (for example — a paragraph or headers).

Example:

```
...
<tr>
    <td style="width: 220px;">XHTML</td>
```

```
    <td style="width: 150px;" align="center">
        Donald MacMatthew </td>
    <td style="width: 60px;" align="right">2004</td>
</tr>

<tr>
    <td style="width: 220px;">Using PHP 5</td>
    <td style="width: 150px;" align="center">
        MAc Matthew</td>
    <td style="width: 60px;" align="right">2005</td>
</tr>
...
```

It is obvious that when you specify the dimensions of each table cell, you form columns of a given width. This means that the total width of the entire table will be equal to the total width of the columns. Therefore, one should not specify the total width of the table in this case. In our example, the first column has a width of 220 pixels, the second — 150 pixels and the third — 60 pixels. The total width of the table will be 430 pixels (without regard to the width of the frame).

Merging cells

In some tasks, you have to build tables whose cells "cover" several columns or rows. Most often this applies to the cells of the headers, less often to the data cells of the table. Below is an example of such a table:

<table>
<tr><td rowspan="2"></td><td colspan="2"></td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
</table>

Pay attention to the "top" of the table. It consists of two lines. This should be taken into account when designing the table.

Managing the merging of cells occurs using the attributes of the <td> and <th> elements:

- **colspan** = "**N**" — combines N columns;
- **rowspan** = "**N**" — combines N lines.

The code for this table is shown below:

```
<table border="1" cellpadding="2" cellspacing="0">
    <tbody>
        <tr style="background-color: #dddddd;">
            <td rowspan="2" style="width: 200px;">
                 </td>
            <td colspan="2" style="width: 100px;">
                 </td>
        </tr>

        <tr style="background-color: #dddddd;">
            <td style="width: 40px;"> </td>
            <td style="width: 60px;"> </td>
        </tr>

        <tr>
            <td style="width: 200px;"> </td>
            <td style="width: 40px;"> </td>
            <td style="width: 60px;"> </td>
        </tr>
            ...
    </tbody>
</table>
```

The "top" lines in the example are assigned the same background color.

When building complex tables that contain multiple merged cells, it is important to remember the following rule: ***the row or column of the table must contain at least one ungrouped cell!*** If this rule is violated, the table will not be displayed correctly.

Row and column groups

In this section we will discuss several new elements that allow you to logically merge the table cells into columns and column groups, and rows into groups of rows. For what may it be necessary? Performing a number of previous examples, you probably thought: "is it really so necessary to set the alignment, width and other attributes for each cell in the column?". Is there such a mechanism that would simplify the tasks of formatting these tables?

Indeed, such a mechanism exists. By combining cells into columns, and columns into groups of rows according to a certain feature, you can set the desired style and attribute to the entire column or even the entire group.

Columns and Column Groups

Columns and column groups are created in tables using two types of elements:

- <col /> — merges the cells in a column;
- <colgroup> ... </ colgroup> — merges the columns in a group.

The <col /> and <colgroup> elements are specified at the beginning of the table before the <tbody>, <thead>, or <tfoot> elements (see below).

The <col /> element supports the following special attributes:

- align="left|right|center|justify" — alignment of data in the cells of the column;
- valign="top|middle|bottom" — vertical alignment of data;
- width="width" — specifies the width of the cells in the column.

In addition, all common attributes are supported. The <col /> element is empty and cannot contain nested elements. Can be used inside the <table> and <colgroup> elements.

The <colgroup> element allows you to merge several columns into a group. This can be useful if several columns have some identical formatting features: alignment, background color, width, etc. Then, common characteristics can be indicated in the <colgroup> element, and list only individual characteristics of individual columns in the nested <col /> elements.

The <colgroup> element supports the following special attributes:

- span="N" — the number of columns to be grouped into a group;
- align="left|right|center|justify" — alignment of data in the cells of the column;
- valign="top|middle|bottom" — vertical alignment of data;
- width="width" — specifies the width of the cells in the column.

The most interesting is the span attribute, which allows you to specify the number of columns to merge. It would

seem that this is a completely useless task. Why to specify the number of columns? The browser can determine it by the number of nested <col /> elements. However, in reality everything is a little different. The <colgroup> element does not need to contain nested <col /> elements. After all, the column is a "virtual" concept, an abstract one. Physically, it does not exist. This is simply a collection of cells located at the same place in each row of the table. Thus, we come to the conclusion that <colgroup> can combine not only real, but virtual columns! :)

Example:

```
<table border="1" cellspacing="0" cellpadding="2">
    <colgroup span="4" width="250" align="center"
    style="background-color: #ddeeff;">
    <col width="100" align="left" />
    <col/>
    <col width="100" align="right"
     style="background-color: #ddffee;" />
    <col />
    </colgroup>
    <col width="120" align="center" style="color:red;" />

    <tbody>
        <tr>
           <th>Header</th>
           <th>Header</th>
           <th>Header</th>
           <th>Header</th>
           <th>Header</th>
       </tr>
       <tr>
           <td>Data</td>
           <td>Data</td>
           <td>Data</td>
```

```
            <td>Data</td>
            <td>Data</td>
        </tr>
         ...
    </tbody>
</table>
```

In the example above, a group of 4 columns is created first. Their default properties are:

- background color style="background-color: #ddeeff;",
- width of cells width="250",
- horizontal align align="center".

Within the group, there are 4 <col /> elements. Some of them apply the individual formatting of a particular column, the rest (those without attributes) are used simply to skip the column. Skipped columns use formatting defined for the entire group. Next, outside the group there is another column with its own individual parameters.

Note that the table cells no longer list any attributes or styles. Thus, using columns and column groups, the table code becomes much more vivid.

Row Groups

In general, table rows can be logically divided into three groups:

- the rows that make up the "top" or the header lines of the table,
- lines of the final part of the table,
- the rest of the lines that make up the contents, the body of the table.

For the logical allocation of these groups of rows, the following elements are used respectively for the title, the final part and the body:

- <thead>;
- <tfoot>;
- <tbody>.

And, the <tbody> element can occur repeatedly in the table.

When using the <thead> and <tfoot> elements, they should be placed before the first of the <tbody> elements at the beginning of the table. This will allow the browser to display the header and totals before all the cells in the table are loaded. Concerning the display order, there is no need to worry: browsers place the contents of the <thead> element at the beginning of the table, and the contents of the <tfoot> element at the very end of the table, after the rows of the table body.

The <thead>, <tbody>, and <tfoot> elements support the following special attributes:

- *align* — horizontal alignment of cells for a group of rows
- *valign* — vertical alignment.

Web Page Layout Using Tables

Web designers rarely use tables as intended; instead they use it to set page structure as a whole. For example, how can one arrange information in several columns (like in a newspaper) on a page? The solution is to create a table of one row and several cells. Arrange information in cells (paragraphs, headers, images, lists, other tables…)

Let's look at the shorthand example code of dividing into columns:

```
<html>
<head>
    <title>Table layout of a website</title>
</head>

<body>
    <table width="100%" border="1" cellpadding="4">
        <thead>
            <tr>
                <td colspan="2"> General Header </td>
            </tr>
        </thead>

        <tbody>
            <tr>
               <td width="50%"> Column 1 </td>
               <td width="50%"> Column 2 </td>
            </tr>
        </tbody>

        <tfoot>
            <tr>
               <td colspan="2" > Concluding Part </td>
            </tr>
        </tfoot>
    </table>

</body>
</html>
```

If you run the example, you will see an article assembled by a template typical for newspapers. Header and preambular part are centered on the top, then there is information part split to columns and finally, concluding part that contains

information about author. I replicated this result using the following table:

<table>
<tr><td colspan="2">General Header</td></tr>
<tr><td>Column 1</td><td>Column 2</td></tr>
<tr><td colspan="2">Concluding Part</td></tr>
</table>

One more example, where you can apply tables to set page structure, is a photo gallery. Using tables, you can arrange thumbnails on pages of gallery sections.

Unit 4.
Creating tables