M8 - Containers – Tables and Divs

Tables

•**Tables are very useful things in Web sites. They perform several functions:**

–Present tabular information in a traditional table format

–Allow precise placement of images and text on an HTML page. To the browser viewer, it usually isn't obvious that a table is even being used

–Special purpose things like putting frames around text and pictures

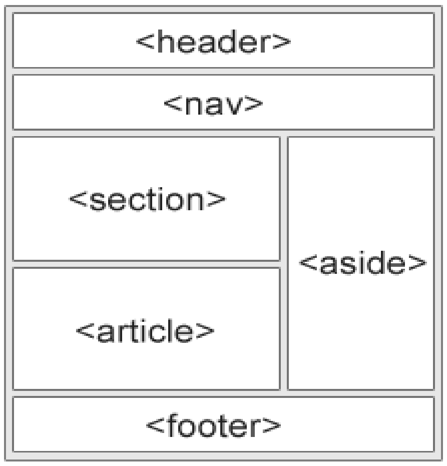
•**The table tags are:**

Tag	Description
<table> and </table>	Defines the beginning and end of the table
<caption> and </caption>	Provides a caption for a table, centered and bold
<tr> and </tr>	Defines a row in the table
<th> and </th>	Defines a bold, centered, column heading
<td> and </td>	Defines a cell in the table

Check out **sample-table.html**.

DIVs and the development of other containers.

Earlier versions of HTML featured an element called a **div** which is basically a rectangular box with a default width of the screen which could be used to structure the layout of a web page. HTML5 recognized that there are usually distinct areas of a page for which div areas were being used and so devised a number of "divs" for specific purposes as the diagram below shows:



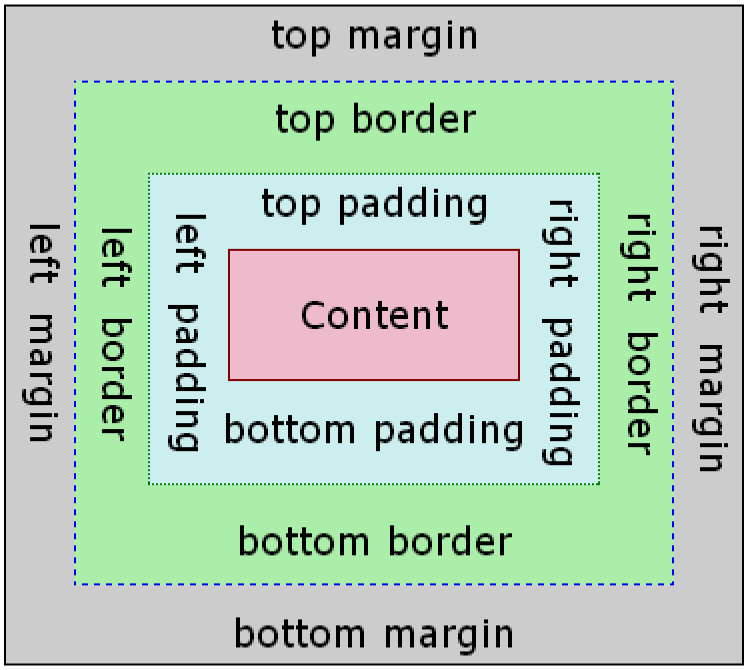
Header	- Defines a header for a document or a section
Nav	- Defines a container for navigation links
Section	- Defines a section in a document
Article	- Defines an independent self-contained article
Aside	- Defines content aside from the content
Footer	- Defines a footer for a document or a section

Each of the above containers can be individually styled, sized, colored and positioned using css. This enables unique web site layouts to be achieved.

Check out **City_of_London.html** in both your browser and in the text editor. Note that the styling is included rather than in a separate css file solely to make it easier to understand the page structure.

Look at **Borders_of_London.html** to see the borders of each of the containers.

The actual content in each of the containers can be separated from the borders and the adjacent containers by setting each of the parameters shown in the following diagram:



Each of the parameters can be set separately for each of the four sides of the container if required. By default they are all zero. Examples are:-

`Padding: 5px;` = 5 pixels on each side

`Padding: 5px 9px;` = 5 pixels top and bottom, 9 pixels on each side

`Padding: 5px 9px 11px;` = 5 top, 9 left and right, 11 at the bottom

`Padding: 5px 9px 11px 7px;` = 5 top, 9 right, 11 bottom,7 left (Clockwise)

See an example of the effects of padding, borders and margins by looking at **edgework.html**.

Note that in every case the content size is the same at 300 x 300 pixels.

Change the size of the viewable area and note the movement that results. What happens if the "**float: left;**" style is deleted?

Container Positioning

Each of the containers can be positioned in a number of ways. Without positioning or use of the "**float: left;**" or "**float: right;**" instruction each container positions its top left corner adjacent to the bottom right corner of the preceding element as based on the order of appearance in the code file, or on the next line if there is not adequate space in the viewport. This default positioning is known as "**Static**" positioning.

"**Relative**" positioning can be used to position the top left corner so many pixels right or left and up or down relative to the position it would normally be if the positioning requirement is not stated.

"**Absolute**" positioning is used to position the top left corner normally relative to the top left corner of the page.

If a page is scrolled up, down or sideways, containers positioned by any of the above 3 methods will move with the detail on the rest of the page. However, if a "**Fixed**" positioning method is used, the container if positioned relative to the browser window and does NOT move if the page is scrolled.