Dr. SNS RAJALAKSHMI COLLEGE OF ARTS AND SCIENCE
(Autonomous)
Coimbatore -641049
Accredited by NAAC (Cycle- III) with ‘Á+’ Grade.

COMPUTER APPLICATIONS

WEB DESIGNING

(16UCA301)

UNIT-II

Class: III BCA
Regulation: 2020-2023
Academic year: 2022-2023
Subject In charge: Ms.S.Lalitha

UNIT-II

HTML Table

HTML table tag is used to display data in tabular form (row * column). There can be many columns in a row.

We can create a table to display data in tabular form, using <table> element, with the help of <tr> , <td>, and <th> elements.

In Each table, table row is defined by <tr> tag, table header is defined by <th>, and table data is defined by <td> tags.

HTML tables are used to manage the layout of the page e.g. header section, navigation bar, body content, footer section etc. But it is recommended to use div tag over table to manage the layout of the page .

HTML Table Tags

Tag	Description
<table>	It defines a table.
<tr>	It defines a row in a table.
<th>	It defines a header cell in a table.
<td>	It defines a cell in a table.
<caption>	It defines the table caption.
<colgroup>	It specifies a group of one or more columns in a table for formatting.
<col>	It is used with <colgroup> element to specify column properties for each column.
<tbody>	It is used to group the body content in a table.
<thead>	It is used to group the header content in a table.
<tfooter>	It is used to group the footer content in a table.

HTML Table Example

```
<!DOCTYPE>

<html>
```

```
<body>

<table>

<tr><th>First_Name</th><th>Last_Name</th><th>Marks</th></tr>

<tr><td>Sonoo</td><td>Jaiswal</td><td>60</td></tr>

<tr><td>James</td><td>William</td><td>80</td></tr>

<tr><td>Swati</td><td>Sironi</td><td>82</td></tr>

<tr><td>Chetna</td><td>Singh</td><td>72</td></tr>

</table>

</body>

</html>
```

OUTPUT:

```
First_Name    Last_Name    Marks

Sonoo         Jaiswal      60

James         William            80

Swati         Sironi       82

Chetna        Singh        72
```

Table Heading

Table heading can be defined using **<th>** tag. This tag will be put to replace <td> tag, which is used to represent actual data cell. Normally you will put your top row as table heading as shown below, otherwise you can use <th> element in any row. Headings, which are defined in <th> tag are centered and bold by default.

Example

```
<!DOCTYPE html>
```

```
<html>

  <head>
    <title>HTML Table Header</title>
  </head>

  <body>
```

```
      <table border = "1">
         <tr>
            <th>Name</th>
            <th>Salary</th>
         </tr>
         <tr>
            <td>Ramesh Raman</td>
            <td>5000</td>
         </tr>

         <tr>
            <td>Shabbir Hussein</td>
            <td>7000</td>
         </tr>
      </table>
   </body>

</html>
```

This will produce the following result −

Cellpadding and Cellspacing Attributes

There are two attributes called *cellpadding* and *cellspacing* which you will use to adjust the white space in your table cells. The cellspacing attribute defines space between table cells, while cellpadding represents the distance between cell borders and the content within a cell.

Example

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML Table Cellpadding</title>
   </head>

   <body>
      <table border = "1" cellpadding = "5" cellspacing = "5">
         <tr>
            <th>Name</th>
            <th>Salary</th>
         </tr>
         <tr>
            <td>Ramesh Raman</td>
            <td>5000</td>
         </tr>
         <tr>
            <td>Shabbir Hussein</td>
            <td>7000</td>
         </tr>
      </table>
   </body>
   
</html>
```

This will produce the following result −

Colspan and Rowspan Attributes

You will use **colspan** attribute if you want to merge two or more columns into a single column. Similar way you will use **rowspan** if you want to merge two or more rows.

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML Table Colspan/Rowspan</title>
   </head>
	
   <body>
      <table border = "1">
         <tr>
            <th>Column 1</th>
            <th>Column 2</th>
            <th>Column 3</th>
         </tr>
         <tr>
            <td rowspan = "2">Row 1 Cell 1</td>
            <td>Row 1 Cell 2</td>
            <td>Row 1 Cell 3</td>
         </tr>
         <tr>
            <td>Row 2 Cell 2</td>
            <td>Row 2 Cell 3</td>
         </tr>
         <tr>
            <td colspan = "3">Row 3 Cell 1</td>
         </tr>
      </table>
   </body>
	
</html>
```

This will produce the following result −

Tables Backgrounds

You can set table background using one of the following two ways −

- **bgcolor** attribute − You can set background color for whole table or just for one cell.
- **background** attribute − You can set background image for whole table or just for one cell.

You can also set border color also using **bordercolor** attribute.

Note − The *bgcolor*, *background*, and *bordercolor* attributes deprecated in HTML5. Do not use these attributes.

Example

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML Table Background</title>
   </head>

   <body>
      <table border = "1" bordercolor = "green" bgcolor = "yellow">
         <tr>
            <th>Column 1</th>
            <th>Column 2</th>
            <th>Column 3</th>
         </tr>
         <tr>
            <td rowspan = "2">Row 1 Cell 1</td>
            <td>Row 1 Cell 2</td>
            <td>Row 1 Cell 3</td>
         </tr>
         <tr>
            <td>Row 2 Cell 2</td>
            <td>Row 2 Cell 3</td>
         </tr>
         <tr>
            <td colspan = "3">Row 3 Cell 1</td>
         </tr>
      </table>
   </body>

</html>
```

This will produce the following result −

Here is an example of using **background** attribute. Here we will use an image available in /images directory.

```
<!DOCTYPE html>
```

```
<html>

  <head>
    <title>HTML Table Background</title>
  </head>

  <body>
    <table border = "1" bordercolor = "green" background = "/images/test.png">
      <tr>
        <th>Column 1</th>
        <th>Column 2</th>
        <th>Column 3</th>
      </tr>
      <tr>
        <td rowspan = "2">Row 1 Cell 1</td>
        <td>Row 1 Cell 2</td><td>Row 1 Cell 3</td>
      </tr>
      <tr>
        <td>Row 2 Cell 2</td>
        <td>Row 2 Cell 3</td>
      </tr>
      <tr>
        <td colspan = "3">Row 3 Cell 1</td>
      </tr>
    </table>
  </body>

</html>
```

This will produce the following result. Here background image did not apply to table's header.

Table Height and Width

You can set a table width and height using **width** and **height** attributes. You can specify table width or height in terms of pixels or in terms of percentage of available screen area.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Table Width/Height</title>
  </head>

  <body>
    <table border = "1" width = "400" height = "150">
      <tr>
```

```
        <td>Row 1, Column 1</td>
        <td>Row 1, Column 2</td>
      </tr>

      <tr>
        <td>Row 2, Column 1</td>
        <td>Row 2, Column 2</td>
      </tr>
    </table>
  </body>

</html>
```

This will produce the following result −

Table Caption

The **caption** tag will serve as a title or explanation for the table and it shows up at the top of the table. This tag is deprecated in newer version of HTML/XHTML.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Table Caption</title>
  </head>

  <body>
    <table border = "1" width = "100%">
      <caption>This is the caption</caption>

      <tr>
        <td>row 1, column 1</td><td>row 1, columnn 2</td>
      </tr>

      <tr>
        <td>row 2, column 1</td><td>row 2, columnn 2</td>
      </tr>
    </table>
  </body>

</html>
```

This will produce the following result −

Table Header, Body, and Footer

Tables can be divided into three portions − a header, a body, and a foot. The head and foot are rather similar to headers and footers in a word-processed document that remain the same for every page, while the body is the main content holder of the table.

The three elements for separating the head, body, and foot of a table are −

- **<thead>** − to create a separate table header.
- **<tbody>** − to indicate the main body of the table.
- **<tfoot>** − to create a separate table footer.

A table may contain several <tbody> elements to indicate *different pages* or groups of data. But it is notable that <thead> and <tfoot> tags should appear before <tbody>

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Table</title>
  </head>

  <body>
    <table border = "1" width = "100%">
      <thead>
        <tr>
          <td colspan = "4">This is the head of the table</td>
        </tr>
      </thead>

      <tfoot>
        <tr>
          <td colspan = "4">This is the foot of the table</td>
        </tr>
      </tfoot>

      <tbody>
        <tr>
          <td>Cell 1</td>
          <td>Cell 2</td>
          <td>Cell 3</td>
          <td>Cell 4</td>
        </tr>
      </tbody>

    </table>
  </body>

</html>
```

This will produce the following result −

Nested Tables

You can use one table inside another table. Not only tables you can use almost all the tags inside table data tag <td>.

Example

Following is the example of using another table and other tags inside a table cell.

!DOCTYPE html>

```
<html>

  <head>
    <title>HTML Table</title>
  </head>

  <body>
    <table border = "1" width = "100%">

      <tr>
        <td>
          <table border = "1" width = "100%">
            <tr>
              <th>Name</th>
              <th>Salary</th>
            </tr>
            <tr>
              <td>Ramesh Raman</td>
              <td>5000</td>
            </tr>
            <tr>
              <td>Shabbir Hussein</td>
              <td>7000</td>
            </tr>
          </table>
        </td>
      </tr>

    </table>
  </body>

</html>
```

HTML table advanced features and accessibility

we look at some more advanced features of HTML tables — such as captions/summaries and grouping your rows into table head, body and footer sections — as well as looking at the accessibility of tables for visually impaired users.

HTML Forms

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

The <form> Element

The HTML <form> element is used to create an HTML form for user input:

```
<form>
.
form elements
.
</form>
```

The <form> element is a container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc.

All the different form elements are covered in this chapter: HTML Form Elements.

The <input> Element

The HTML <input> element is the most used form element.

An <input> element can be displayed in many ways, depending on the type attribute.

Here are some examples:

Type	Description

<input type="text">	Displays a single-line text input field
<input type="radio">	Displays a radio button (for selecting one of many choices)
<input type="checkbox">	Displays a checkbox (for selecting zero or more of many choices)
<input type="submit">	Displays a submit button (for submitting the form)
<input type="button">	Displays a clickable button

Example:

```
<!DOCTYPE html>
<html>
<body>

<h2>Text input fields</h2>

<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe">
```

```
</form>


<p>Note that the form itself is not visible.</p>


<p>Also note that the default width of text input fields is 20 characters.</p>


</body>

</html>
```

Output:

Text input fields

First name:

John

Last name:

Doe

Note that the form itself is not visible.

Also note that the default width of text input fields is 20 characters.

The HTML <form> Elements

The HTML <form> element can contain one or more of the following form elements:

- <input>
- <label>
- <select>
- <textarea>
- <button>
- <fieldset>
- <legend>
- <datalist>
- <output>
- <option>
- <optgroup>

The <input> Element

One of the most used form element is the <input> element.

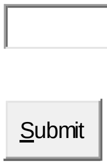
The <input> element can be displayed in several ways, depending on the type attribute.

```
<!DOCTYPE html>

<html>

<body>

<h2>The input Element</h2>

<form action="/action_page.php">

  <label for="fname">First name:</label><br>

  <input type="text" id="fname" name="fname"><br><br>

  <input type="submit" value="Submit">

</form>

</body>

</html>
```

Output:

The input Element

First name:



The <label> Element

The <label> element defines a label for several form elements.

The <label> element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focus on the input element.

The <label> element also help users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the <label> element, it toggles the radio button/checkbox.

The for attribute of the <label> tag should be equal to the id attribute of the <input> element to bind them together.

The <select> Element

The <select> element defines a drop-down list:

Example

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

The <option> elements defines an option that can be selected.

By default, the first item in the drop-down list is selected.

To define a pre-selected option, add the selected attribute to the option:

Example

```
<option value="fiat" selected>Fiat</option>
```

Visible Values:

Use the size attribute to specify the number of visible values:

Example

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars" size="3">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

Allow Multiple Selections:

Use the multiple attribute to allow the user to select more than one value:

Example

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars" size="4" multiple>
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

The <textarea> Element

The <textarea> element defines a multi-line input field (a text area):

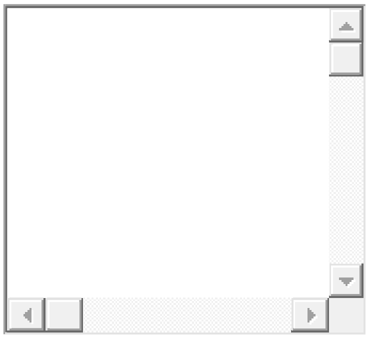
Example

```
<textarea name="message" rows="10" cols="30">
The cat was playing in the garden.
</textarea>
```

The rows attribute specifies the visible number of lines in a text area.

The cols attribute specifies the visible width of a text area.

This is how the HTML code above will be displayed in a browser:



You can also define the size of the text area by using CSS:

Example

```
<textarea name="message" style="width:200px; height:600px;">
The cat was playing in the garden.
</textarea>
```

The <fieldset> and <legend> Elements

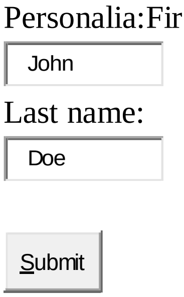
The <fieldset> element is used to group related data in a form.

The <legend> element defines a caption for the <fieldset> element.

Example

```
<form action="/action_page.php">
  <fieldset>
    <legend>Personalia:</legend>
    <label for="fname">First name:</label><br>
    <input type="text" id="fname" name="fname" value="John"><br>
    <label for="lname">Last name:</label><br>
    <input type="text" id="lname" name="lname" value="Doe"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

This is how the HTML code above will be displayed in a browser:



Personalia:First name:

John

Last name:

Doe

Submit

The <datalist> Element

The <datalist> element specifies a list of pre-defined options for an <input> element.

Users will see a drop-down list of the pre-defined options as they input data.

The list attribute of the <input> element, must refer to the id attribute of the <datalist> element.

Example

```
<form action="/action_page.php">
  <input list="browsers">
  <datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
</form>
```

The <output> Element

The <output> element represents the result of a calculation (like one performed by a script).

Example

Perform a calculation and show the result in an <output> element:

```
<form action="/action_page.php"
  oninput="x.value=parseInt(a.value)+parseInt(b.value)">
  0
  <input type="range"  id="a" name="a" value="50">
  100 +
  <input type="number" id="b" name="b" value="50">
  =
  <output name="x" for="a b"></output>
  <br><br>
  <input type="submit">
</form>
```

Using HTML form controls

The objective of this technique is to use standard HTML form controls and link elements to provide keyboard operation and assistive technology interoperability of interactive user interface elements.

User agents provide the keyboard operation of HTML form controls and links. In addition, the user agent maps the form controls and links to an accessibility API. Assistive technologies use the accessibility API to extract appropriate accessibility information, such as role, name, state, and value, and present them to users. The role is provided by the HTML element, and the name is provided by the text associated with that element. Elements for which values and states are appropriate also expose the values and states via multiple mechanisms.

In some cases, the text is already associated with the control through a required attribute. For example, submit buttons use the button element text or image 'alt' attribute as the name. In the case of form controls, label elements or 'title' attributes are used. The following table describes how the role, name, value, and state are determined for HTML links and form controls.

HTML element	Role	Name	Value	State
<a>	link	'title' attribute, text within <a> element or 'alt' attribute if image link. Concatenated if both text and image 'alt' attribute are provided	'href' attribute	
<button>	push button	text inside <button> element or 'title' attribute		
<fieldset>	grouping	text inside <legend> element within fieldset element		
<input type = "button", "submit", or "reset">	push button	'value' attribute		
<input type = "image">	push button	'alt' attribute or 'title' attribute		
<input type = "text">	editable text	<label> element associated with it or 'title' attribute	'value' attribute	
<input type = "password">	editable text	<label> element associated with it or 'title' attribute	value is purposefully hidden	
<input type="file">	editable text	<label> element associated with it or 'title' attribute	'value' attribute	
<input type="checkbox">	checkbox	<label> element associated with it or 'title' attribute		'checked' attribute
<input type="radio">	radio button	<label> element associated with it or 'title' attribute		'checked' attribute
<select>	list box	<label> element associated with it or 'title' attribute	<option> element with 'selected' attribute set to "selected"	

HTML element	Role	Name	Value	State
<textarea>	editable text	<label> element associated with it or 'title' attribute	text within <textarea> element	

HTML – Frames

HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document. A collection of frames in the browser window is known as a frameset. The window is divided into frames in a similar way the tables are organized: into rows and columns.

Creating Frames

To use frames on a page we use <frameset> tag instead of <body> tag. The <frameset> tag defines, how to divide the window into frames. The **rows** attribute of <frameset> tag defines horizontal frames and **cols** attribute defines vertical frames. Each frame is indicated by <frame> tag and it defines which HTML document shall open into the frame.

Note − The <frame> tag deprecated in HTML5. Do not use this element.

Example

Following is the example to create three horizontal frames −

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Frames</title>
  </head>

  <frameset rows = "10%,80%,10%">
    <frame name = "top" src = "/html/top_frame.htm" />
    <frame name = "main" src = "/html/main_frame.htm" />
    <frame name = "bottom" src = "/html/bottom_frame.htm" />

    <noframes>
      <body>Your browser does not support frames.</body>
    </noframes>

  </frameset>

</html>
```

This will produce the following result −

Example

Let's put the above example as follows, here we replaced rows attribute by cols and changed their width. This will create all the three frames vertically −

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Frames</title>
  </head>

  <frameset cols = "25%,50%,25%">
    <frame name = "left" src = "/html/top_frame.htm" />
    <frame name = "center" src = "/html/main_frame.htm" />
    <frame name = "right" src = "/html/bottom_frame.htm" />

    <noframes>
      <body>Your browser does not support frames.</body>
    </noframes>
  </frameset>

</html>
```

This will produce the following result −

The <frameset> Tag Attributes

Following are important attributes of the <frameset> tag −

Sr.No	Attribute & Description
1	**cols** Specifies how many columns are contained in the frameset and the size of each column. You can specify the width of each column in one of the four ways − Absolute values in pixels. For example, to create three vertical frames, use *cols = "100, 500, 100"*. A percentage of the browser window. For example, to create three vertical frames, use *cols = "10%, 80%, 10%"*. Using a wildcard symbol. For example, to create three vertical frames, use *cols = "10%, *, 10%"*. In this case wildcard takes remainder of the window. As relative widths of the browser window. For example, to create three vertical frames, use *cols = "3*, 2*, 1*"*. This is an alternative to percentages. You can use relative

	widths of the browser window. Here the window is divided into sixths: the first column takes up half of the window, the second takes one third, and the third takes one sixth.
2	**rows** This attribute works just like the cols attribute and takes the same values, but it is used to specify the rows in the frameset. For example, to create two horizontal frames, use *rows = "10%, 90%"*. You can specify the height of each row in the same way as explained above for columns.
3	**border** This attribute specifies the width of the border of each frame in pixels. For example, border = "5". A value of zero means no border.
4	**frameborder** This attribute specifies whether a three-dimensional border should be displayed between frames. This attribute takes value either 1 (yes) or 0 (no). For example frameborder = "0" specifies no border.
5	**framespacing** This attribute specifies the amount of space between frames in a frameset. This can take any integer value. For example framespacing = "10" means there should be 10 pixels spacing between each frames.

The <frame> Tag Attributes

Following are the important attributes of <frame> tag −

Sr.No	Attribute & Description
1	**src** This attribute is used to give the file name that should be loaded in the frame. Its value can be any URL. For example, src = "/html/top_frame.htm" will load an HTML file available in html directory.
2	**name** This attribute allows you to give a name to a frame. It is used to indicate which frame a

	document should be loaded into. This is especially important when you want to create links in one frame that load pages into an another frame, in which case the second frame needs a name to identify itself as the target of the link.
3	**frameborder** This attribute specifies whether or not the borders of that frame are shown; it overrides the value given in the frameborder attribute on the <frameset> tag if one is given, and this can take values either 1 (yes) or 0 (no).
4	**marginwidth** This attribute allows you to specify the width of the space between the left and right of the frame's borders and the frame's content. The value is given in pixels. For example marginwidth = "10".
5	**marginheight** This attribute allows you to specify the height of the space between the top and bottom of the frame's borders and its contents. The value is given in pixels. For example marginheight = "10".
6	**noresize** By default, you can resize any frame by clicking and dragging on the borders of a frame. The noresize attribute prevents a user from being able to resize the frame. For example noresize = "noresize".
7	**scrolling** This attribute controls the appearance of the scrollbars that appear on the frame. This takes values either "yes", "no" or "auto". For example scrolling = "no" means it should not have scroll bars.
8	**longdesc** This attribute allows you to provide a link to another page containing a long description of the contents of the frame. For example longdesc = "framedescription.htm"

Browser Support for Frames

If a user is using any old browser or any browser, which does not support frames then <noframes> element should be displayed to the user.

So you must place a <body> element inside the <noframes> element because the <frameset> element is supposed to replace the <body> element, but if a browser does not understand <frameset> element then it should understand what is inside the <body> element which is contained in a <noframes> element.

You can put some nice message for your user having old browsers. For example, *Sorry!! your browser does not support frames.* as shown in the above example.

Frame's name and target attributes

One of the most popular uses of frames is to place navigation bars in one frame and then load main pages into a separate frame.

Let's see following example where a test.htm file has following code −

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML Target Frames</title>
   </head>

   <frameset cols = "200, *">
      <frame src = "/html/menu.htm" name = "menu_page" />
      <frame src = "/html/main.htm" name = "main_page" />

      <noframes>
         <body>Your browser does not support frames.</body>
      </noframes>
   </frameset>

</html>
```

Here, we have created two columns to fill with two frames. The first frame is 200 pixels wide and will contain the navigation menu bar implemented by **menu.htm** file. The second column fills in remaining space and will contain the main part of the page and it is implemented by **main.htm** file. For all the three links available in menu bar, we have mentioned target frame as **main_page**, so whenever you click any of the links in menu bar, available link will open in main page.

Following is the content of menu.htm file

```
<!DOCTYPE html>
<html>

   <body bgcolor = "#4a7d49">
      <a href = "http://www.google.com" target = "main_page">Google</a>
      <br />
      <br />

      <a href = "http://www.microsoft.com" target = "main_page">Microsoft</a>
      <br />
      <br />
      
      <a href = "http://news.bbc.co.uk" target = "main_page">BBC News</a>
   </body>
   
</html>
```

Following is the content of main.htm file −

```
<!DOCTYPE html>
<html>

   <body bgcolor = "#b5dcb3">
      <h3>This is main page and content from any link will be displayed here.</h3>
      <p>So now click any link and see the result.</p>
   </body>
   
</html>
```

When we load **test.htm** file, it produces following result −

Now you can try to click links available in the left panel and see the result. The *targetattribute* can also take one of the following values −

Sr.No	Option & Description
1	**_self** Loads the page into the current frame.
2	**_blank** Loads a page into a new browser window. Opening a new window.
3	**_parent** Loads the page into the parent window, which in the case of a single frameset is the main browser window.
4	**_top** Loads the page into the browser window, replacing any current frames.

5	**targetframe** Loads the page into a named targetframe.

<frame> tag – It defines each frame of the frameset. It is an empty element, and has no content but it indicates the web page to be shown in the frame given by the 'src' attribute like

<frame src="top.htm" />

<frame src="bottom.htm" />

<noframes> tag – It is used for browsers which does not support frames, then <noframes> tag shows content enclosed in it to the user. The <body> element is placed in <noframes> tag as <frameset> tag replaces it. Usually some message indicating old browser is shown to the user, for example

<noframes>

<body>

<p>Your browser does not support frames. </p>

</body>

</noframes>

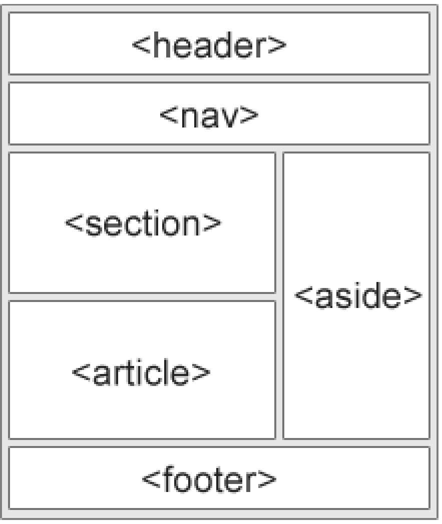
Frame layout elements:

The task is to create frames using HTML. HTML is **H**yper**T**ext **M**arkup **L**anguage, and it is the standard markup language for creating web pages. It describes the shape of a web page and includes a sequence of elements.

Frames in HTML are used to divide your browser window into more than one section in which every phase can load a separate HTML report and a group of frames within the browser window is referred to as a frameset.

It is never recommended to use frames on your web page because of the following reasons.

-
- Smaller gadgets can't deal with frames frequently due to the fact that their display isn't always sufficient enough to be divided up.
- Due to different screen resolutions, your page will be displayed differently on different computers.
- It happens sometimes that your browser's *back* button doesn't work as the user hopes.
- There are few browsers that do not supports frame.



- <header> - Defines a header for a document or a section
- <nav> - Defines a set of navigation links
- <section> - Defines a section in a document
- <article> - Defines an independent, self-contained content
- <aside> - Defines content aside from the content (like a sidebar)
- <footer> - Defines a footer for a document or a section
- <details> - Defines additional details that the user can open and close on demand
- <summary> - Defines a heading for the <details> element