

WEB TECHNOLOGY

MODULE - II

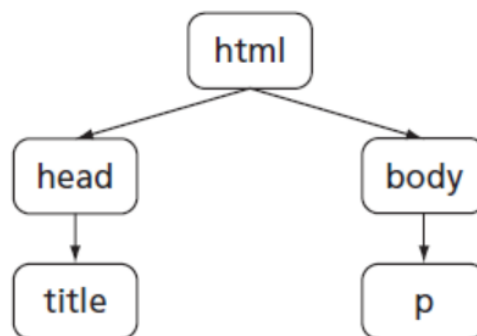
Introduction to HTML – Elementary tags in HTML – List in HTML – Displaying Text in Lists – Using Ordered List – Using Unordered Lists- HTML Description Lists - Nested HTML Lists, Control List – Combining List Types – Graphics and Image Formats – Graphics and HTML document- image and hyperlink anchors – Image maps – Tables – Frames – Forms.

An Introduction to HTML

Before saying any more about the Hypertext Markup Language, let's briefly look at a small example file to gain a more concrete understanding of HTML syntax and semantics. The following HTML example presents an HTML "Hello World!" document. Figure 2.2 shows how this document would appear if it were opened using the Mozilla browser as discussed in the previous chapter (the figure may not look much like a browser window, because toolbars and menus

```
<HTML>
<head>
  <title>    HelloWorld.html </title>
</head>
<body>
  <p>
    Hello World!
  </p>
</body>
</HTML>
```

The body element contains the information that is to be displayed in the client area of the browser. This document's body contains a single paragraph (p) element. Notice that only the content of this element is displayed; the p start and end tags are used to inform the browser about the content and are not displayed themselves. A p element in particular tells the browser that its content represents a single paragraph of text (and possibly other elements) and should be displayed accordingly.



HTML Tags

As told earlier, HTML is a markup language and makes use of various tags to format the content. These tags are enclosed within angle braces **<Tag Name>**. Except few tags, most of the tags have their corresponding closing tags. For example, **<html>** has its closing tag **</html>** and **<body>** tag has its closing tag **</body>** tag etc.

Above example of HTML document uses the following tags:

Tag	Description
<!DOCTYPE...>	This tag defines the document type and HTML version.
<html>	This tag encloses the complete HTML document and mainly comprises of document header which is represented by <head>...</head> and document body which is represented by <body>...</body> tags.
<head>	This tag represents the document's header which can keep other HTML tags like <title>, <link> etc.
<title>	The <title> tag is used inside the <head> tag to mention the document title.
<body>	This tag represents the document's body which keeps other HTML tags like <h1>, <div>, <p> etc.
<h1>	This tag represents the heading.

Lists

HTML offers web authors three ways for specifying lists of information. All lists must contain one or more list elements. Lists may contain –

- **** – An unordered list. This will list items using plain bullets.
- **** – An ordered list. This will use different schemes of numbers to list your items.
- **<dl>** – A definition list. This arranges your items in the same way as they are arranged in a dictionary.

OL-Ordered List

If we are required to put our items in a numbered list instead of bulleted, then HTML ordered list will be used. This list is created by using **** tag. The numbering starts at one and is incremented by one for each successive ordered list element tagged with ****.

Coffee

Tea

Milk

Boost

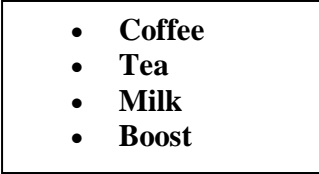
- | |
|---|
| <ol style="list-style-type: none">1. Coffee2. Tea3. Milk4. Boost |
|---|

UL-Unordered List

An unordered list is a collection of related items that have no special order or sequence. This list is created by using HTML `` tag. Each item in the list is marked with a bullet.

Example

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
  <li>Boost</li>
</ul>
```



- Coffee
- Tea
- Milk
- Boost

Definition List

HTML and XHTML supports a list style which is called **definition lists** where entries are listed like in a dictionary or encyclopedia. The definition list is the ideal way to present a glossary, list of terms, or other name/value list.

```
<dl>
<dt>RAM</dt>
<dd>Random Access Memory</dd>
<dt>ROM</dt>
<dd>Read Only Memory</dd>
</dl>
```

RAM
Random Access Memory

ROM
Read Only Memory

Nested List

A nested list or a sublist is a list within a list. The trick to marking nested lists up correctly in HTML is to recognize that the sublist is actually a child of a list item and not of a list.

1. Start by creating a list. It can be ordered or unordered:

```
<ul>
  <li>Fruit</li>
  <li>Vegetables</li>
  <li>Meat</li>
</ul>
```

2. Now add a nested list to the first list item:

```
<ul>
```

```
<li>Fruit
  <ul>
    <li>Bananas</li>
    <li>Apples</li>
    <li>Pears</li>
  </ul>
</li>
<li>Vegetables</li>
<li>Meat</li>
</ul>
```

Notice that the sublist is a child and not a sibling of an `` tag.

Graphics and HTML

HTML Canvas

The HTML `<canvas>` element is used to draw graphics, on the fly, via JavaScript.

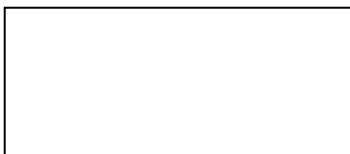
The `<canvas>` element is only a container for graphics. You must use JavaScript to actually draw the graphics.

Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

Example 1

A canvas is a rectangular area on an HTML page. By default, a canvas has no border and no content. The markup looks like this:

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

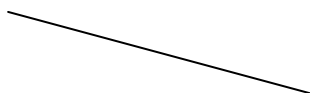


Example 2

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.moveTo(0, 0);
ctx.lineTo(200, 100);
ctx.stroke();
</script>
```

Output



SVG-Graphics

SVG defines vector-based graphics in XML format.

SVG is used to define graphics for the Web

SVG is a W3C recommendation

The HTML <svg> Element

The HTML <svg> element is a container for SVG graphics.

SVG has several methods for drawing paths, boxes, circles, text, and graphic images.

Example

```
<html>
<body>
  <svg width="100" height="100">
    <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
  </svg>
</body>
</html>
```



HTML Images

Images can improve the design and the appearance of a web page.

The HTML tag is used to embed an image in a web page.

Images are not technically inserted into a web page; images are linked to web pages.

The tag creates a holding space for the referenced image.

The tag is empty, it contains attributes only, and does not have a closing tag.

The tag has two required attributes:

src - Specifies the path to the image

alt - Specifies an alternate text for the image

The required src attribute specifies the path (URL) to the image

HTML – Hyper Links

Links are found in nearly all web pages. Links allow users to click their way from page to page.

HTML links are hyperlinks.

We can click on a link and jump to another document.

When we move the mouse over a link, the mouse arrow will turn into a little hand.

HTML Links - Syntax

The HTML <a> tag defines a hyperlink. It has the following syntax:

```
<a href="url">link text</a>
```

The most important attribute of the <a> element is the href attribute, which indicates the link's destination.

The link text is the part that will be visible to the reader.

Clicking on the link text, will send the reader to the specified URL address.

Example

This example shows how to create a link to W3Schools.com:

```
<a href="https://www.w3schools.com/">Visit W3Schools.com!</a>
```

By default, links will appear as follows in all browsers:

An unvisited link is underlined and blue

A visited link is underlined and purple

An active link is underlined and red

Image Maps

The HTML `<map>` tag defines an image map. An image map is an image with clickable areas. The areas are defined with one or more `<area>` tags.

Try to click on the computer, phone, or the cup of coffee in the image below:



Example

Here is the HTML source code for the image map above: The image is inserted using the `` tag. The only difference from other images is that you must add a `usemap` attribute: The `usemap` value starts with a hash tag `#` followed by the name of the image map, and is used to create a relationship between the image and the image map.

Then, add a `<map>` element.

The `<map>` element is used to create an image map, and is linked to the image by using the required `name` attribute:

Then, add the clickable areas.

A clickable area is defined using an `<area>` element.

Shape

We must define the shape of the clickable area, and you can choose one of these values:

`rect` - defines a rectangular region

circle - defines a circular region
poly - defines a polygonal region
default - defines the entire region

We must also define some coordinates to be able to place the clickable area onto the image.

```

<map name="workmap">
  <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm">
  <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">
  <area shape="circle" coords="337,300,44" alt="Coffee" href="coffee.htm">
</map>
```

HTML Tables

HTML tables allow web developers to arrange data into rows and columns.

- **The <table> tag defines an HTML table.**
- **Each table row is defined with a <tr> tag.**
- **Each table header is defined with a <th> tag.**
- **Each table data/cell is defined with a <td> tag.**
- **By default, the text in <th> elements are bold and centered.**
- **By default, the text in <td> elements are regular and left-aligned.**

```
<h2>An unordered HTML list</h2>
```

```
<TABLE BORDER=1>
  <TR><TH>SLNO</TH><TH>Product</TH><TH>PRICE</TH></TR>
  <TR><TD>01</TD><TD>HAMMAM</TD><TD>50</TD></TR>
  <TR>
    <TD>02</TD>
    <TD>LUX</TD>
    <TD>35</TD>
  </TR>
</TABLE>
</body>
</html>
```

Output

SLNO	Product	PRICE
01	HAMMAM	50
02	LUX	35

The <form> Element

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

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

elements

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

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

Text Fields

The <input type="text"> defines a single-line input field for text input.

Example

A form with input fields for text:

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

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

First name:

Last name:

The <label> Element

Notice the use of the <label> element in the example above.

The <label> tag defines a label for many 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.

Radio Buttons

The `<input type="radio">` defines a radio button.

Radio buttons let a user select ONE of a limited number of choices.

Example

A form with radio buttons:

```
<p>Choose your favorite Web language:</p>

<form>
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language" value="JavaScript">
  <label for="javascript">JavaScript</label>
</form>
```

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

Choose your favorite Web language:

- HTML
- CSS
- JavaScript

Checkboxes

The `<input type="checkbox">` defines a **checkbox**.

Checkboxes let a user select ZERO or MORE options of a limited number of choices.

Example

A form with checkboxes:

```
<form>
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> I have a boat</label>
</form>
```

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

- I have a bike
- I have a car
- I have a boat

The Submit Button

The `<input type="submit">` defines a button for submitting the form data to a form-handler.

The form-handler is typically a file on the server with a script for processing input data.

The form-handler is specified in the form's `action` attribute.

Example

A form with a submit button:

```
<form action="/action_page.php">
  <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">
</form>
```

The Name Attribute for <input>

Notice that each input field must have a `name` attribute to be submitted.

If the `name` attribute is omitted, the value of the input field will not be sent at all.

Example

This example will not submit the value of the "First name" input field:

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