Chapter two

2. Hyper Text Markup Language (HTML)

2.1. HTML Introduction

What is HTML?

HTML is an acronym which stands for Hyper Text Markup Language.

Hyper Text: Hyper Text simply means "Text within Text". A text has a link within it, is a hypertext. Every time when you click on a word which brings you to a new webpage, you have clicked on a hypertext.

Markup language: A markup language is a programming language that is used make text more interactive and dynamic. It can turn a text into images, tables, links etc.

- HTML is used to create web pages.
- HTML is widely used language on the web.
- We can create static website by HTML only.

What do you need to get started?

To get started HTML you need a computer installed with HTML editor and web browser.

- **HTML Editor:** HTML can be written and edited using many different editors like Dreamweaver and Visual Studio. However, in this course we use a plain text editor (like Notepad, Note pad ++) to edit HTML.
- **Web Browser:** such as Mozilla Firefox, Google Chrome, Opera etc.

2.2. HTML Basics, Elements and Attributes

An HTML document is made of many HTML tags and each HTML tag contains different content.

HTML Tags

HTML tags contain three main parts: opening tag, content and closing tag. But some HTML tags are unclosed tags. When a web browser reads an HTML document, browser reads it from top to bottom and left to right. HTML tags are used to create HTML documents and render their properties. Each HTML tags have different properties.

Syntax: <tag> content </tag>

- HTML tags are keywords surrounded by **angle brackets** like <html>
- HTML tags normally **come in pairs** like and
- The first tag in a pair is the **start tag,** the second tag is the **end tag**
- Start and end tags are also called **opening tags** and **closing tags**

Unclosed HTML Tags

Some HTML tags are not closed, for example br, hr, img etc.

- **
 Tag**: br stands for break line, it breaks the line of the code.
- **<hr> Tag**: hr stands for Horizontal Rule. This tag is used to put a line across the webpage.

HTML Documents = Web Pages

- HTML documents **describe web pages**
- HTML documents **contain HTML tags** and plain text
- HTML documents are also **called web pages**

The purpose of a web browser (like Internet Explorer or Firefox) is to read HTML documents and display them as web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page:

```
<html><body><h1>My First Heading</h1> <p>My first paragraph.</p>
</body></html>
```

- The text between <html> and </html> describes the web page
- The text between <body> and </body> is the visible page content
- The text between <h1> and </h1> is displayed as a heading
- The text between and is displayed as a paragraph

HTML head Elements

Tag	Description
<head>	Defines information about the document
<title>	Defines the title of a document
<base>	Defines a default address or a default target for all links on a page
<link>	Defines the relationship between a document and an external resource
<meta>	Defines metadata about an HTML document
<script>	Defines a client-side script
<style>	Defines style information for a document

The HTML `<head>` element has nothing to do with HTML headings. The `<head>` element is a container for metadata. HTML metadata is data about the HTML document. Metadata is not displayed. The `<head>` element is placed between the `<html>` tag and the `<body>` tag:

Example

```
<!DOCTYPE html>
<html><head> <title>My First HTML</title>  <meta charset="UTF-
8"></head><body>…
```

Note: Metadata typically define the document title, character set, styles, links, scripts, and other meta information.

How to View HTML Source?

Have you ever seen a Web page and wondered "Hey! How did they do that?"

View HTML Source Code:

- Right-click in an HTML page and select "View Page Source" (in Chrome) or "View Source" (in IE), or similar in other browsers. This will open a window containing the HTML source code of the page.

HTML Elements

An HTML element usually consists of a **start** tag and **end** tag, with the content inserted in between:

```
<tagname>Content goes here...</tagname>
```

The HTML **element** is everything from the start tag to the end tag:

```
<p>My first paragraph.</p>
```

Start tag	Element content	End tag
<h1>	My First Heading	</h1>
	My first paragraph.	

HTML elements with no content are called empty elements. Empty elements do not have an end tag, such as the
 element (which indicates a line break).

Nested HTML Elements

HTML elements can be nested (elements can contain elements).

All HTML documents consist of nested HTML elements.

This example contains four HTML elements:

Example

```
<!DOCTYPE html> <html><body><h1>My First Heading</h1><p>My first paragraph.</p></body></html>
```

The `<html>` element defines the **whole document**. It has a **start** tag <html> and an **end** tag </html>. The element **content** is another HTML element (the `<body>` element). The `<body>` element defines the **document body**. It has a **start** tag <body> and an **end** tag </body>. The element **content** is two other HTML elements (`<h1>` and `<p>`). The `<h1>` element defines a **heading**. It has a **start** tag <h1> and an **end** tag </h1>. The element **content** is: My First Heading. The `<p>` element defines a **paragraph**. It has a **start** tag and an **end** tag .

The element **content** is: My first paragraph.

Note:

- **Do Not Forget the End Tag-** Some HTML elements will display correctly, even if you forget the end tag. But some browsers might produce unexpected results and/or errors if you forget the end tag.
- HTML elements with no content are called empty elements. Empty elements can be "closed" in the opening tag like this:
. HTML5 does not require empty elements to be closed. But if you want stricter validation, or if you need to make your document readable by XML parsers, you must close all HTML elements properly.
- Use Lowercase Tags - HTML tags are not case sensitive: <P> means the same as . The HTML5 standard does not require lowercase tags, but W3C **recommends** lowercase in HTML, and **demands** lowercase for stricter document types like XHTML.

HTML Attributes

- All HTML elements can have **attributes**
- Attributes provide **additional information** about an element
- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like: **name="value"**

Example

- <a **href**="https://www.google.come">This is a link - href is used to specify a link address.
- <img **src**="img.jpg"> - src is used to specify filename of an image source.
- <img src="img.jpg" width="500" height="600"> - Images in HTML have a set of **size** attributes, which specifies the width and height of the image
- I am a paragraph - style attribute is used to specify the styling of an element. Such as color, font, size etc.

2.3. HTML Heading, Paragraphs, Styles, Formatting, Quotations, Computer Codes, Comments and Colors

HTML Headings

Headings are defined with the `<h1>` to `<h6>` tags. `<h1>` defines the most important heading. `<h6>` defines the least important heading.

Example

```
<h1>Heading 1</h1> <h2>Heading 2</h2> <h3>Heading 3</h3>
<h4>Heading 4</h4> <h5>Heading 5</h5> <h6>Heading 6</h6>
```

Note: Browsers automatically add some white space (a margin) before and after a heading.

HTML Horizontal Rules

The `<hr>` tag defines a thematic break in an HTML page, and is most often displayed as a horizontal rule.

The `<hr>` element is used to separate content (or define a change) in an HTML page:

Example

```
<h1>This is heading 1</h1><p>This is some text.</p> <hr>
<h2>This is heading 2</h2><p>This is some other text.</p> <hr>
```

HTML Paragraphs

The HTML `<p>` element defines a **paragraph**:

Example

```
<p>This is a paragraph.</p> <p>This is another paragraph.</p>
```

Note: Browsers automatically add some white space (a margin) before and after a paragraph.

HTML Display

You cannot be sure how HTML will be displayed. Large or small screens, and resized windows will create different results. With HTML, you cannot change the output by adding extra spaces or extra lines in your HTML code. The browser will remove any extra spaces and extra lines when the page is displayed:

Example

```
<p>This paragraph contains a lot of lines in the source code, but the browser ignores it.</p> <p>This paragraph contains a lot of spaces in the source code, but the  browser ignores it. </p>
```

Don't Forget the End Tag

Most browsers will display HTML correctly even if you forget the end tag:

Example

```
<p>This is a paragraph.   <p>This is another paragraph.
```

The example above will work in most browsers, but do not rely on it.

Note: Dropping the end tag can produce unexpected results or errors.

HTML Line Breaks

The HTML `<br>` element defines a **line break**.

Use `<br>` if you want a line break (a new line) without starting a new paragraph:

Example

```
<p>This is<br>a paragraph<br>with line breaks.</p>
```

The `<br>` tag is an empty tag, which means that it has no end tag.

The HTML Style Attribute

Setting the style of an HTML element, can be done with the `style` attribute.

The HTML `style` attribute has the following **syntax**:

```
<tagname style="property:value;">
```

Example

```
<body style="background-color:powderblue;color:yellow;font-family:verdana;">
<h1>This is a heading</h1><p>This is a paragraph.</p>
</body>
```

The ***property*** is a CSS property. The ***value*** is a CSS value.

HTML Formatting Elements

Like Style attribute, HTML also defines special **elements** for defining text with a special **meaning**. HTML uses elements like `<b>` and `<i>` for formatting output, like **bold** or *italic* text.

Formatting elements were designed to display special types of text:

HTML <b> and Elements

The HTML `<b>` element defines **bold** text, without any extra importance.

Example

This text is bold

The HTML `<strong>` element defines **strong** text, with added semantic "strong" importance.

Example

This text is strong

HTML <i> and Elements

The HTML `<i>` element defines *italic* text, without any extra importance.

Example

```
<i>This text is italic</i>
```

The HTML `<em>` element defines *emphasized* text, with added semantic importance.

Example

```
<em>This text is emphasized</em>
```

Note: Browsers display `<strong>` as `<b>`, and `<em>` as `<i>`. However, there is a difference in the meaning of these tags: `<b>` and `<i>` defines bold and italic text, but `<strong>` and `<em>` means that the text is "important".

HTML <small> Element

The HTML `<small>` element defines smaller text:

Example

```
<h2>HTML <small>Small</small> Formatting</h2>
```

HTML <mark> Element

The HTML `<mark>` element defines marked or highlighted text:

Example

```
<h2>HTML <mark>Marked</mark> Formatting</h2>
```

HTML <del> Element

The HTML `<del>` element defines ~~deleted~~ (removed) text.

Example

```
<p>My favorite color is <del>blue</del> red.</p>
```

HTML <ins> Element

The HTML `<ins>` element defines inserted (added) text.

Example

```
<p>My favorite <ins>color</ins> is red.</p>
```

HTML <sub> Element

The HTML `<sub>` element defines _{subscripted} text.

Example

```
<p>This is <sub>subscripted</sub> text.</p>
```

HTML <sup> Element

The HTML `<sup>` element defines superscripted text.

Example

```
<p>This is <sup>superscripted</sup> text.</p>
```

HTML <q> for Short Quotations

The HTML `<q>` element defines a short quotation.

Browsers usually insert quotation marks around the `<q>` element.

Example

```
<p>WWF's goal is to: <q>Build a future where people live in harmony with nature.</q></p>
```

HTML <blockquote> for Quotations

The HTML `<blockquote>` element defines a section that is quoted from another source.

Browsers usually indent `<blockquote>` elements.

Example

```
<p>Here is a quote from WWF's website:</p> <blockquote>
content</blockquote>
```

HTML <abbr> for Abbreviations

The HTML `<abbr>` element defines an abbreviation or an acronym.

Marking abbreviations can give useful information to browsers, translation systems and search-engines.

Example

```
<p>The <abbr title="World Health Organization">WHO</abbr> was founded in 1948.</p>
```

HTML <address> for Contact Information

The HTML `<address>` element defines contact information (author/owner) of a document or an article.

The `<address>` element is usually displayed in italic. Most browsers will add a line break before and after the element.

Example

```
<address> Written by John Doe.<br> Visit us at:<br> Example.com<br>
Box 564, Disneyland<br> USA </address>
```

HTML <cite> for Work Title

The HTML `<cite>` element defines the title of a work. Browsers usually display `<cite>` elements in italic.

Example: <cite>The Scream</cite> by Edvard Munch. Painted in 1893.

HTML <bdo> for Bi-Directional Override

The HTML `<bdo>` element defines bi-directional override. The `<bdo>` element is used to override the current text direction:

Example: <bdo dir="rtl">This text will be written from right to left</bdo>

HTML Comments

Comment tags are used to insert comments in the HTML source code. You can add comments to your HTML source by using the following syntax:

<!-- Write your comments here -->

Notice that there is an exclamation point (!) in the opening tag, but not in the closing tag.

Note: Comments are not displayed by the browser, but they can help document your HTML source code.

With comments you can place notifications and reminders in your HTML:

Example

<!-- This is a comment --> This is a paragraph. <!-- Remember to add more information here -->

Comments are also great for debugging HTML, because you can comment out HTML lines of code, one at a time, to search for errors:

Example

<!-- Do not display this at the moment <img border="0" src="pic_trulli.jpg" alt="Trulli"> -->

HTML Computer Code Elements

Computer Code

<code> x = 5;
 y = 6;
 z = x + y; </code>

HTML <kbd> For Keyboard Input

The HTML `<kbd>` element represents user input, like keyboard input or voice commands.

Text surrounded by `<kbd>` tags is typically displayed in the browser's default monospace font:

Example

Save the document by pressing <kbd>Ctrl + S</kbd>

HTML <samp> For Program Output

The HTML `<samp>` element represents output from a program or computing system.

Text surrounded by `<samp>` tags is typically displayed in the browser's default monospace font:

Example

If you input wrong value, the program will return <samp>Error!</samp>

HTML <code> For Computer Code

The HTML `<code>` element defines a fragment of computer code.

Text surrounded by `<code>` tags is typically displayed in the browser's default monospace font:

Example

<code> x = 5; y = 6; z = x + y; </code>

Notice that the `<code>` element does not preserve extra whitespace and line-breaks. To fix this, you can put the `<code>` element inside a `<pre>` element:

Example

<pre><code>x = 5; y = 6; z = x + y; </code> </pre>

HTML <var> For Variables

The HTML `<var>` element defines a variable.

The variable could be a variable in a mathematical expression or a variable in programming context:

Example

Einstein wrote: <var>E</var> = <var>mc</var>².

HTML Entities

Reserved characters in HTML must be replaced with character entities. Characters that are not present on your keyboard can also be replaced by entities. Some characters are reserved in HTML. If you use the less than (<) or greater than (>) signs in your text, the browser might mix them with tags. Character entities are used to display reserved characters in HTML.

A character entity looks like this: &*entity_name*; OR &#*entity_number*;

To display a less than sign (<) we must write: **<** or **<**

Advantage of using an entity name: An entity name is easy to remember. **Disadvantage of using an entity name:** Browsers may not support all entity names, but the support for numbers is good.

Non-breaking Space

A common character entity used in HTML is the non-breaking space: ** **

A non-breaking space is a space that will not break into a new line.

Two words separated by a non-breaking space will stick together (not break into a new line). This is handy when breaking the words might be disruptive.

Examples: § 10, 10 km/h and 10 PM

Another common use of the non-breaking space is to prevent browsers from truncating spaces in HTML pages. If you write 10 spaces in your text, the browser will remove 9 of them. To add real spaces to your text, you can use the ** ** character entity. The non-breaking hyphen (‑) lets you use a hyphen character (-) that won't break.

Some Other Useful HTML Character Entities

Result	Description	Entity Name	Entity Number
	non-breaking space		
<	less than	<	<
>	greater than	>	>
&	ampersand	&	&
"	double quotation mark	"	"
'	single quotation mark (apostrophe)	'	'
¢	cent	¢	¢

£	pound	£	£
¥	yen	¥	¥
€	euro	€	€
©	copyright	©	©
®	registered trademark	®	®

Note: Entity names are case sensitive.

2.4. HTML links, images, tables and lists

HTML Links

Links are found in nearly all web pages. Links allow users to click their way from page to page. HTML links are hyperlinks. You can click on a link and jump to another document. When you move the mouse over a link, the mouse arrow will turn into a little hand.

Note: A link does not have to be text. It can be an image or any other HTML element.

HTML Links - Syntax

In HTML, links are defined with the `<a>` tag: *link text*

Example: Visit our website

The `href` attribute specifies the destination address (https://www.dmu.edu.et) of the link.

The **link text** is the visible part (Visit our website). Clicking on the link text will send you to the specified address.

Note: Without a forward slash at the end of subfolder addresses, you might generate two requests to the server. Many servers will automatically add a forward slash to the end of the address, and then create a new request.

Local Links

The example above used an absolute URL (a full web address). A local link (link to the same web site) is specified with a relative URL (without https://www....).

Example: HTML Images

HTML Link Colors

By default, a link will appear like this (in all browsers):

- An unvisited link is **underlined and blue**
- A visited link is **underlined and purple**
- An active link is **underlined and red**

You can change the default colors, by using CSS.

HTML Links - The target Attribute

The `target` attribute specifies where to open the linked document. The target attribute can have one of the following values:

- _blank - Opens the linked document in a new window or tab
- _self - Opens the linked document in the same window/tab as it was clicked (this is default)
- _parent - Opens the linked document in the parent frame
- _top - Opens the linked document in the full body of the window
- *framename* - Opens the linked document in a named frame

Tip: If your webpage is locked in a frame, you can use `target="_top"` to break out of the frame:

Link Titles

The `title` attribute specifies extra information about an element. The information is most often shown as a tooltip text when the mouse moves over the element.

Example

Visit our website

HTML Links - Create a Bookmark

HTML bookmarks are used to allow readers to jump to specific parts of a Web page. Bookmarks can be useful if your webpage is very long. To make a bookmark, you must first create the bookmark, and then add a link to it. When the link is clicked, the page will scroll to the location with the bookmark.

Example

First, create a bookmark with the `id` attribute: <h2 id="C4">Chapter 4</h2>

Then, add a link to the bookmark ("Jump to Chapter 4"), from within the same page:

Jump to Chapter 4

Or, add a link to the bookmark ("Jump to Chapter 4"), from another page:

Example: Jump to Chapter 4

External Paths

External pages can be referenced with a full URL or with a path relative to the current web page. This example uses a full URL to link to a web page:

Example: HTML tutorial

This example links to a page located in the html folder on the current web site:

Example: HTML tutorial

This example links to a page located in the same folder as the current page:

Example: HTML tutorial

HTML Images

Images can improve the design and the appearance of a web page. In HTML, images are defined with the `<img>` tag. The `<img>` tag is empty, it contains attributes only, and does not have a closing tag. The `src` attribute specifies the URL (web address) of the image:

<img src="*url*">

The alt Attribute

The `alt` attribute provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the src attribute, or if the user uses a screen reader). The value of the `alt` attribute should describe the image.

Image Size - Width and Height

You can use the `style` attribute to specify the width and height of an image. The `width` and `height` attributes always defines the width and height of the image in pixels.

Example: <img src="img_girl.jpg" alt="Girl in a jacket" style="width:500px;height:600px;">

Alternatively, you can use the `width` and `height` attributes:

<img src="img_girl.jpg" alt="Girl in a jacket" width="500" height="600">

Image Maps

The `<map>` tag defines an image-map. An image-map is an image with clickable areas. In the image below, click on the computer, the phone, or the cup of coffee:

Example

```
<img src="workplace.jpg" alt="Workplace" usemap="#workmap">
  <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>
```

The `name` attribute of the `<map>` tag is associated with the `<img>`'s usemap attribute and creates a relationship between the image and the map.

The <map> element contains a number of <area> tags, that define the clickable areas in the image-map.

Note: Loading images takes time. Large images can slow down your page. Use images carefully.

HTML Tables

An HTML table is defined with the <table> tag. Each table row is defined with the <tr> tag. A table header is defined with the <th> tag. By default, table headings are bold and centered. A table data/cell is defined with the <td> tag.

Example

```
<table style="width:100%">
  <tr>    <th>Firstname</th>     <th>Lastname</th>      <th>Age</th>   </tr>
  <tr>     <td>Jill</td>    <td>Smith</td>     <td>50</td>  </tr>
  <tr>    <td>Eve</td>    <td>Jackson</td>     <td>94</td>  </tr>
</table>
```

Note: The <td> elements are the data containers of the table. They can contain all sorts of HTML elements; text, images, lists, other tables, etc.

HTML Table - Cells that Span Many Columns

To make a cell span more than one column, use the colspan attribute:

Example

```
<table style="width:100%">  <tr>    <th>Name</th>   <th colspan="2">Telephone</th>  </tr>
  <tr>    <td>Bill Gates</td>    <td>55577854</td>    <td>55577855</td>  </tr> </table>
```

HTML Table - Cells that Span Many Rows

To make a cell span more than one row, use the rowspan attribute:

Example

```
<table style="width:100%">  <tr>    <th>Name:</th>    <td>Bill Gates</td>  </tr>  <tr>    <th
rowspan="2">Telephone:</th>    <td>55577854</td>  </tr> <tr>    <td>55577855</td>  </tr></table>
```

HTML Table - Adding a Caption

To add a caption to a table, use the <caption> tag:

Example

```
<table style="width:100%">  <caption>Monthly savings</caption>
  <tr>    <th>Month</th>    <th>Savings</th>  </tr>  <tr>    <td>January</td>    <td>$100</td>  </tr>
<tr>    <td>February</td>    <td>$50</td>  </tr></table>
```

Note: The <caption> tag must be inserted immediately after the <table> tag.

HTML Lists

Unordered HTML List

An unordered list starts with the `<ul>` tag. Each list item starts with the `<li>` tag. The list items will be marked with bullets (small black circles) by default:

Example

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

Unordered HTML List - Choose List Item Marker

The CSS `list-style-type` property is used to define the style of the list item marker:

Value	Description
disc	Sets the list item marker to a bullet (default)
circle	Sets the list item marker to a circle
square	Sets the list item marker to a square
none	The list items will not be marked

Example - Disc

```
<ul style="list-style-type:disc">  <li>Coffee</li>  <li>Tea</li>  <li>Milk</li></ul>
```

Example - Circle

```
<ul style="list-style-type:circle">  <li>Coffee</li>  <li>Tea</li>  <li>Milk</li></ul>
```

Example - Square

```
<ul style="list-style-type:square">  <li>Coffee</li>  <li>Tea</li>  <li>Milk</li></ul>
```

Example - None

```
<ul style="list-style-type:none">  <li>Coffee</li>  <li>Tea</li>  <li>Milk</li></ul>
```

Ordered HTML List

An ordered list starts with the `<ol>` tag. Each list item starts with the `<li>` tag. The list items will be marked with numbers by default:

Example

```
<ol>  <li>Coffee</li>  <li>Tea</li>  <li>Milk</li> </ol>
```

Ordered HTML List - The Type Attribute

The `type` attribute of the `<ol>` tag, defines the type of the list item marker:

Type	Description
type="1"	The list items will be numbered with numbers (default)
type="A"	The list items will be numbered with uppercase letters
type="a"	The list items will be numbered with lowercase letters

type="I"	The list items will be numbered with uppercase roman numbers
type="i"	The list items will be numbered with lowercase roman numbers

Numbers:

<ol type="1"> Coffee Tea Milk

Uppercase Letters:

<ol type="A"> Coffee Tea Milk

Lowercase Letters:

<ol type="a"> Coffee Tea Milk

Uppercase Roman Numbers:

<ol type="I"> Coffee Tea Milk

Lowercase Roman Numbers:

<ol type="i"> Coffee Tea Milk

HTML Description Lists

HTML also supports description lists. A description list is a list of terms, with a description of each term. The `<dl>` tag defines the description list, the `<dt>` tag defines the term (name), and the `<dd>` tag describes each term:

Example

<dl> <dt>Coffee</dt> <dd>- black hot drink</dd> <dt>Milk</dt> <dd>- white cold drink</dd></dl>

2.5. HTML Iframe and Forms

HTML Iframes

An iframe is used to display a web page within a web page.

Iframe Syntax

An HTML iframe is defined with the `<iframe>` tag: <iframe src="*URL*"></iframe>

The `src` attribute specifies the URL (web address) of the inline frame page.

Iframe - Set Height and Width

Use the `height` and `width` attributes to specify the size of the iframe. The attribute values are specified in pixels by default, but they can also be in percent (like "80%").

Example

<iframe src="demo_iframe.htm" height="200" width="300"></iframe>

Or you can use CSS to set the height and width of the iframe:

Example

<iframe src="demo_iframe.htm" style="height:200px;width:300px;"></iframe>

Iframe - Remove the Border

By default, an iframe has a border around it. To remove the border, add the `style` attribute and use the CSS `border` property:

Example

<iframe src="demo_iframe.htm" style="border:none;"></iframe>

With CSS, you can also change the size, style and color of the iframe's border:

Example

<iframe src="demo_iframe.htm" style="border:2px solid red;"></iframe>

Iframe - Target for a Link

An iframe can be used as the target frame for a link.

The `target` attribute of the link must refer to the `name` attribute of the iframe:

Example

<iframe src="demo_iframe.htm" name="iframe_a"></iframe>

W3Schools.com

HTML Forms

The HTML `<form>` element defines a form that is used to collect user input:

<form> *form elements* </form>

An HTML form contains **form elements**. Form elements are different types of input elements, like text fields, checkboxes, radio buttons, submit buttons, and more.

Attributes of Form Tag

The Action Attribute

The `action` attribute defines the action to be performed when the form is submitted.

Normally, the form data is sent to a web page on the server when the user clicks on the submit button. In the example above, the form data is sent to a page on the server called "/action_page.php". This page contains a server-side script that handles the form data:

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

If the `action` attribute is omitted, the action is set to the current page.

The Target Attribute

The `target` attribute specifies if the submitted result will open in a new browser tab, a frame, or in the current window.

The default value is "`_self`" which means the form will be submitted in the current window.

To make the form result open in a new browser tab, use the value "`_blank`":

Example

<form action="/action_page.php" **target="_blank"**>

Other legal values are "`_parent`", "`_top`", or a name representing the name of an iframe.

The Method Attribute

The `method` attribute specifies the HTTP method (**GET** or **POST**) to be used when submitting the form data:

Example: <form action="/action_page.php" **method="get"**> or

<form action="/action_page.php" **method="post"**>

When to Use GET?

The default method when submitting form data is GET. However, when GET is used, the submitted form data will be **visible in the page address field**:

/action_page.php?firstname=Mickey&lastname=Mouse

Notes on GET:

- Appends form-data into the URL in name/value pairs
- The length of a URL is limited (about 3000 characters)
- Never use GET to send sensitive data! (will be visible in the URL)
- Useful for form submissions where a user wants to bookmark the result
- GET is better for non-secure data, like query strings in Google

When to Use POST?

Always use POST if the form data contains sensitive or personal information. The POST method does not display the submitted form data in the page address field.

Notes on POST:

POST has no size limitations, and can be used to send large amounts of data.

Form submissions with POST cannot be bookmarked

The Name Attribute

Each input field must have a `name` attribute to be submitted. If the `name` attribute is omitted, the data of that input field will not be sent at all. This example will only submit the "Last name" input field:

Example

```
<form action="/action_page.php">
  First name:<br>  <input type="text" value="Mickey"><br>
  Last name:<br>  <input type="text" name="lastname" value="Mouse"><br><br>
  <input type="submit" value="Submit">
</form>
```

HTML Form Elements

The <input> Element

The most important form element is the `<input>` element. The `<input>` element can be displayed in several ways, depending on the `type` attribute.

Example

```
<input name="firstname" type="text">
```

If the `type` attribute is omitted, the input field gets the default type: "text".

HTML Input types

The `<input>` element is the most important form element. The `<input>` element can be displayed in several ways, depending on the **type** attribute

Input Type Text

`<input type="text">` defines a **one-line text input field**:

Input Type Password

`<input type="password">` defines a **password field**:

The characters in a password field are masked (shown as asterisks or circles).

Input Type Submit

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

The form-handler is typically a server page with a script for processing input data. The form-handler is specified in the form's `action` attribute:

Example

```
<form action="/action_page.php">
  First name:<br>  <input type="text" name="firstname" value="Mickey"><br>
  Last name:<br>  <input type="text" name="lastname" value="Mouse"><br><br>
  <input type="submit" value="Submit">
</form>
```

If you omit the submit button's value attribute, the button will get a default text.

Input Type Reset

`<input type="reset">` defines a **reset button** that will reset all form values to their default values.

If you change the input values and then click the "Reset" button, the form-data will be reset to the default values.

Input Type Radio

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

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

Input Type Checkbox

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

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

Input Type Button

`<input type="button">` defines a **button**:

Example

```
<input type="button" onclick="alert('Hello World!')" value="Click Me!">
```

HTML5 Input Types

HTML5 added several new input types:

- Color, date, datetime-local, email, month, number, range, search, tel, time, url and week

New input types that are not supported by older web browsers, will behave as `<input type="text">`.

Input Type Color

The `<input type="color">` is used for input fields that should contain a color. Depending on browser support, a color picker can show up in the input field.

Input Type Date

The `<input type="date">` is used for input fields that should contain a date. Depending on browser support, a date picker can show up in the input field. You can also use the `min` and `max` attributes to add restrictions to dates:

Example

```
<form>
  Enter a date before 1980-01-01:  <input type="date" name="bday" max="1979-12-31"><br>
  Enter a date after 2000-01-01:  <input type="date" name="bday" min="2000-01-02"><br>
</form>
```

Input Type Datetime-local

The `<input type="datetime-local">` specifies a date and time input field, with no time zone. Depending on browser support, a date picker can show up in the input field.

Example

```
<form>   Birthday (date and time):   <input type="datetime-local" name="bdaytime"> </form>
```

Input Type Email

The `<input type="email">` is used for input fields that should contain an e-mail address. Depending on browser support, the e-mail address can be automatically validated when submitted. Some smartphones recognize the email type, and adds ".com" to the keyboard to match email input.

Input Type File

The `<input type="file">` defines a file-select field and a "Browse" button for file uploads.

Input Type Month

The `<input type="month">` allows the user to select a month and year. Depending on browser support, a date picker can show up in the input field.

Input Type Number

The `<input type="number" min="1" max="5">>` defines a **numeric** input field. You can also set restrictions on what numbers are accepted. The following example displays a numeric input field, where you can enter a value from 1 to 5:

Input Restrictions

Here is a list of some common input restrictions (some are new in HTML5):

Attribute	Description

disabled	Specifies that an input field should be disabled
max	Specifies the maximum value for an input field
maxlength	Specifies the maximum number of character for an input field
min	Specifies the minimum value for an input field
pattern	Specifies a regular expression to check the input value against
readonly	Specifies that an input field is read only (cannot be changed)
required	Specifies that an input field is required (must be filled out)
size	Specifies the width (in characters) of an input field
step	Specifies the legal number intervals for an input field
value	Specifies the default value for an input field

The following example displays a numeric input field, where you can enter a value from 0 to 100, in steps of 10. The default value is 30:

Example

<form>Quantity: <input type="number" name="points" min="0" max="100" step="10" value="30"></form>

Input Type Range

The `<input type="range">` defines a control for entering a number whose exact value is not important (like a slider control). Default range is 0 to 100. However, you can set restrictions on what numbers are accepted with the `min`, `max`, and `step` attributes:

Example

<form><input type="range" name="points" min="0" max="10"></form>

Input Type Search

The `<input type="search">` is used for search fields (a search field behaves like a regular text field).

Input Type Tel

The `<input type="tel">` is used for input fields that should contain a telephone number.

Note: The tel type is currently only supported in Safari 8.

Input Type Time

The `<input type="time">` allows the user to select a time (no time zone). Depending on browser support, a time picker can show up in the input field.

Input Type Url

The `<input type="url">` is used for input fields that should contain a URL address. Depending on browser support, the url field can be automatically validated when submitted.

Some smartphones recognize the url type, and adds ".com" to the keyboard to match url input.

Input Type Week

The `<input type="week">` allows the user to select a week and year. Depending on browser support, a date picker can show up in the input field.

HTML Input Attributes

- The `value` attribute specifies the initial value for an input field:
- The `readonly` attribute specifies that the input field is read only (cannot be changed):
- The `disabled` attribute specifies that the input field is disabled.
- The `size` attribute specifies the size (in characters) for the input field:
- The `maxlength` attribute specifies the maximum allowed length for the input field: With a `maxlength` attribute, the input field will not accept more than the allowed number of characters.

Note: Input restrictions are not foolproof, and JavaScript provides many ways to add illegal input. To safely restrict input, it must be checked by the receiver (the server) as well!

The <select> Element

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

Example

```
<select 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

```
<select 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

```
<select 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" style="width:200px; height:600px"> 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.

The <button> Element

The `<button>` element defines a clickable **button**:

Example

<button type="button" onclick="alert('Hello World!')">Click Me!</button>

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

Note: Always specify the **type** attribute for the button element. Different browsers may use different default types for the button element.

Grouping Form Data with <fieldset>

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>Personal information:</legend>
   First name:<br>    <input type="text" name="firstname" value="Mickey"><br>
   Last name:<br>    <input type="text" name="lastname" value="Mouse"><br><br>
   <input type="submit" value="Submit">
 </fieldset></form>
```

HTML5 Form Elements

HTML5 added the following form elements:

- `<datalist>`
- `<output>`

Note: Browsers do not display unknown elements. New elements that are not supported in older browsers will not "destroy" your web page.

HTML5 <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>
```

HTML5 <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>
```

HTML5 Attributes

HTML5 added the following attributes for `<input>`:

- Autocomplete, autofocus, form, formaction, formenctype, formmethod, formnovalidate, formtarget, height and width, list, min and max, multiple, pattern (regexp), placeholder, required, step

and the following attributes for `<form>`:

- Autocomplete, novalidate

The form Attribute

The `form` attribute specifies one or more forms an `<input>` element belongs to.

- ✓ The `formaction` attribute specifies the URL of a file that will process the input control when the form is submitted. The formaction attribute overrides the action attribute of the `<form>` element. The formaction attribute is used with `type="submit"` and `type="image"`.
- ✓ The `formenctype` attribute specifies how the form data should be encoded when submitted (only for forms with method="post"). The `formenctype` attribute overrides the enctype attribute of the `<form>` element. The `formenctype` attribute is used with `type="submit"` and `type="image"`.
- ✓ The `formmethod` attribute defines the HTTP method for sending form-data to the action URL. The `formmethod` attribute overrides the method attribute of the `<form>` element. The `formmethod` attribute can be used with `type="submit"` and `type="image"`.
- ✓ The `formnovalidate` attribute overrides the novalidate attribute of the `<form>` element. The `formnovalidate` attribute can be used with `type="submit"`.
- ✓ The `formtarget` attribute specifies a name or a keyword that indicates where to display the response that is received after submitting the form. The `formtarget` attribute overrides the target attribute of the `<form>` element. The `formtarget` attribute can be used with `type="submit"` and `type="image"`.

- ✓ The `height` and `width` attributes specify the height and width of an `<input type="image">` element. Always specify the size of images. If the browser does not know the size, the page will flicker while images load.
- ✓ <input type="image" src="img_submit.gif" alt="Submit" width="48" height="48">
- ✓ The `list` attribute refers to a `<datalist>` element that contains pre-defined options for an <input> element.
- ✓ The `min` and `max` attributes specify the minimum and maximum values for an `<input>` element. The `min` and `max` attributes work with the following input types: number, range, date, datetime-local, month, time and week.
- ✓ The `multiple` attribute specifies that the user is allowed to enter more than one value in the `<input>` element. The `multiple` attribute works with the following input types: email, and file. Example: Select images: <input type="file" name="img" multiple>
- ✓ The `pattern` attribute specifies a regular expression that the `<input>` element's value is checked against. The `pattern` attribute works with the following input types: text, search, url, tel, email, and password. Example: <input type="text" name="country_code" pattern="[A-Za-z]{3}" title="Three letter country code">
- ✓ The `placeholder` attribute specifies a hint that describes the expected value of an input field (a sample value or a short description of the format). The hint is displayed in the input field before the user enters a value. The `placeholder` attribute works with the following input types: text, search, url, tel, email, and password.
- ✓ The `required` attribute specifies that an input field must be filled out before submitting the form. The `required` attribute works with the following input types: text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file.
- ✓ The `step` attribute specifies the legal number intervals for an `<input>` element. Example: if step="3", legal numbers could be -3, 0, 3, 6, etc.

Tip: The step attribute can be used together with the max and min attributes to create a range of legal values. The `step` attribute works with the following input types: number, range, date, datetime-local, month, time and week.

2.6. HTML 5

HTML5 is a next version of HTML. Here, you will get some brand new features which will make HTML much easier. These new introducing features make your website layout clearer to both website designers and users. There are some elements like <header>, <footer>, <nav> and <article> that define the layout of a website.

Why use HTML5

- It is enriched with advance features which makes it easy and interactive for designer/developer and users.
- It allows you to play a video and audio file.

- It allows you to draw on a canvas.
- It facilitates you to design better forms and build web applications that work offline.
- It provides you advanced features for that you would normally have to write JavaScript to do.

The most important reason to use HTML 5 is, we believe it is not going anywhere. It will be here to serve for a long time according to W3C recommendation.

Example:

```
<!DOCTYPE>  <html>  <body>
<h1>Write Your First Heading</h1> <p>Write Your First Paragraph.</p>
</body>  </html>
```

HTML 5 Tags

There is a list of newly included tags in HTML 5. These HTML 5 tags (elements) provide a better document structure. This list shows all HTML 5 tags in alphabetical order with description.

List of HTML 5 Tags

Tag	Description
<article>	This element is used to define an independent piece of content in a document, that may be a blog, a magazine or a newspaper article.
<aside>	It specifies that article is slightly related to the rest of the whole page.
<audio>	It is used to play audio file in HTML.
<bdi>	The bdi stands for bi-directional isolation. It isolates a part of text that is formatted in other direction from the outside text document.
<canvas>	It is used to draw canvas.
<data>	It provides machine readable version of its data.
<datalist>	It provides auto complete feature for textfield.
<details>	It specifies the additional information or controls required by user.
<dialog>	It defines a window or a dialog box.
<figcaption>	It is used to define a caption for a <figure> element.
<figure>	It defines a self-contained content like photos, diagrams etc.
<footer>	It defines a footer for a section.
<header>	It defines a header for a section.
<main>	It defines the main content of a document.
<mark>	It specifies the marked or highlighted content.

<menuitem>	It defines a command that the user can invoke from a popup menu.
<meter>	It is used to measure the scalar value within a given range.
<nav>	It is used to define the navigation link in the document.
<progress>	It specifies the progress of the task.
<rp>	It defines what to show in browser that don't support ruby annotation.
<rt>	It defines an explanation/pronunciation of characters.
<ruby>	It defines ruby annotation along with <rp> and <rt>.
<section>	It defines a section in the document.
<summary>	It specifies a visible heading for <detailed> element.
<svg>	It is used to display shapes.
<time>	It is used to define a date/time.
<video>	It is used to play video file in HTML.
<wbr>	It defines a possible line break.

HTML Audio Tag

HTML audio tag is used to define sounds such as music and other audio clips. Audio tag supports some file formats such as : mp3, wav and ogg.

HTML5 supports <video> and <audio> controls. The Flash, Silverlight and similar technologies are used to play the multimedia items.

Example:

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
Your browser does not support the audio element.
</audio>
```

HTML Video Tag

HTML 5 supports <video> tag also. The HTML video tag is used for streaming video files such as a movie clip, song clip on the web page. HTML5 Video tag supports some file formats such as mp4, webM and ogg

Example:

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>
```

Migration from HTML4 to HTML5

This section demonstrates how to convert an HTML4 page into an HTML5 page, without destroying anything of the original content or structure.

Typical HTML4	Typical HTML5
	<header>
	<nav>
	<section>
	<article>
	<footer>

A Typical HTML4 Page

Example

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html;charset=utf-8">
<title>HTML4</title>
<style>
body {font-family: Verdana,sans-serif; font-size: 0.9em; }
div#header,div#footer{ padding:10px;color:white;
                       background-color: black;}
div#content {  margin:5px; padding:10px;background-color:lightgrey;}
div.article {  margin: 5px; padding: 10px; background-color: white;}
div#menu ul {  padding: 0;}
div#menu ul li {  display: inline;  margin: 5px;}
</style>
</head>
<body>
<div id="header"> <h1>Monday Times</h1></div>
<div id="menu"><ul><li>News</li><li>Sports</li><li>Weather</li></ul>
</div>
<div id="content">  <h2>News Section</h2>
  <div class="article"><h2>News Article</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Pellentesque in porta lorem. Morbi condimentum est nibh, et
```

```
consectetur tortor feugiat at.</p>
  </div>
  <div class="article"> <h2>News Article</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Pellentesque in porta lorem. Morbi condimentum est nibh, et
consectetur tortor feugiat at.</p>
  </div>
</div>
<div id="footer"> <p>&amp;copy; 2016 Monday Times. All rights
reserved.</p>
</div>
</body>
</html>
```

Change to HTML5 Doctype

Change the **doctype**:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

to the HTML5 doctype:

Example

```
<!DOCTYPE html>
```

Change to HTML5 Encoding

Change the **encoding** information:

```
<meta http-equiv="Content-Type" content="text/html;charset=utf-8">
```

to HTML5 encoding:

Example

```
<meta charset="utf-8">
```

Change to HTML5 Semantic Elements

The existing CSS contains id's and classes for styling the elements:

```
body {font-family: Verdana,sans-serif;  font-size: 0.9em; }
div#header,div#footer{ padding:10px; color:white;
                    background-color:black; }
 div#content {margin:5px; padding:10px; background-color:lightgrey; }
 div.article { margin:5px; padding:10px; background-color:white; }
 div#menu ul { padding: 0; }
 div#menu ul li { display: inline;  margin: 5px; }
```

Replace with equal CSS styles for HTML5 semantic elements:

```
body {    font-family: Verdana,sans-serif;  font-size: 0.9em;    }
header, footer {padding:10px; color:white; background-color:black; }
section{ margin: 5px; padding: 10px; background-color: lightgrey;}
article {  margin: 5px;   padding: 10px;  background-color: white;}
nav ul {  padding: 0; }
nav ul li { display: inline;  margin: 5px; }
```

Finally, change the elements to HTML5 semantic elements:

Example

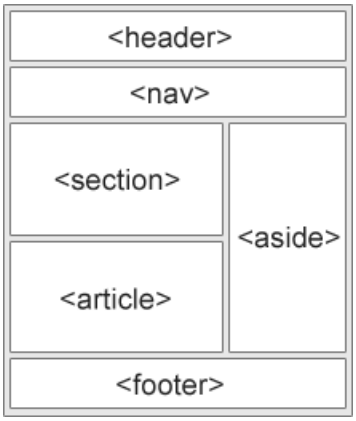
```
<body>
<header><h1>Monday Times</h1></header>
<nav><ul><li>News</li><li>Sports</li><li>Weather</li></ul></nav>
<section><h2>News Section</h2><article>
<h2>News Article</h2> <p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Pellentesque in porta lorem. Morbi condimentum est
nibh, et consectetur tortor feugiat at.</p></article>
<article><h2>News Article</h2><p>Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Pellentesque in porta lorem. Morbi
condimentum est nibh, et consectetur tortor feugiat at.</p>
</article> </section>
<footer> <p>&copy; 2014 Monday Times. All rights reserved.</p>
</footer></body>
```

HTML Layouts

HTML Layout Elements

Websites often display content in multiple columns (like a magazine or newspaper).

HTML5 offers new semantic elements that define the different parts of a web page:



- <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 (like a sidebar)
- <footer> - Defines a footer for a document or a section
- <details> - Defines additional details
- <summary> - Defines a heading for the <details> element

HTML Layout Techniques

There are five different ways to create multicolumn layouts. Each way has its pros and cons:

- HTML tables (not recommended)
- CSS float property
- CSS flexbox
- CSS framework
- CSS grid

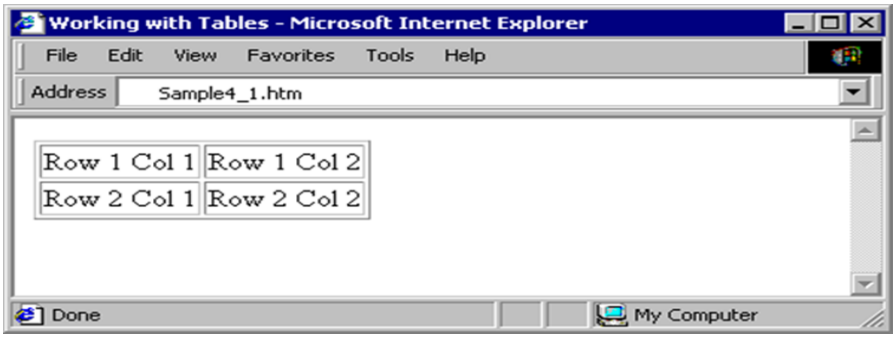
Table

Tables can serve many roles within a given Web page.Tables are defined with the <table> tag.

Starting Tag	Ending Tag	Description
<TABLE>	</TABLE>	Defines the table.
<TR>	</TR>	Defines a new row.
<TD>	</TD>	Defines a new column of data.
<TH>	</TH>	Defines a table header cell.
<CAPTION>	</CAPTION>	Defines a caption for the table that can be placed to the left, top, bottom, or right of the table.

Example

```
<TABLE  BORDER="1">
<TR>
<TD>Row 1 Col 1</TD>
<TD>Row 1 Col 2</TD>
</TR>
<TR>
<TD>Row 2 Col 1</TD>
<TD>Row 2 Col 2</TD>
</TR>
</TABLE>
```



Attributes of <Table> Tag

Attribute	Description
ALIGN	Sets the alignment of the content within the cell. Accepts the values: left, right, and center.
WIDTH	Sets the width of the column. Units are in pixels or percentages.
COLS	Sets the number of columns within the table. (This is new in the HTML 4.0 specifications.)
BORDER	Sets the width of the border. Units are in pixels.
CELLSPACING	Sets the distance between cells. Units are in pixels.
CELLPADDING	Sets the distance between the border of the cell and the cell's contents. Units are in pixels.
RULES	Sets where lines are positioned to separate cells. Possible values include: none, groups, rows, cols, and all.

2.7. HTML Frame

HTML frame is used for create layouts of the page.In HTML frame have two important concepts.

1.frame set

2.frame

<Frameset>Tag: The <FRAMESET> tag requires that you use at least one of two attributes to define the frame layout: ROWS or COLS.

COLS:-Sets the number of frames to create in column format as well as their size.

Its size, separated by commas, defines each column.

Size can be set using either an absolute value or a percentage.

Example

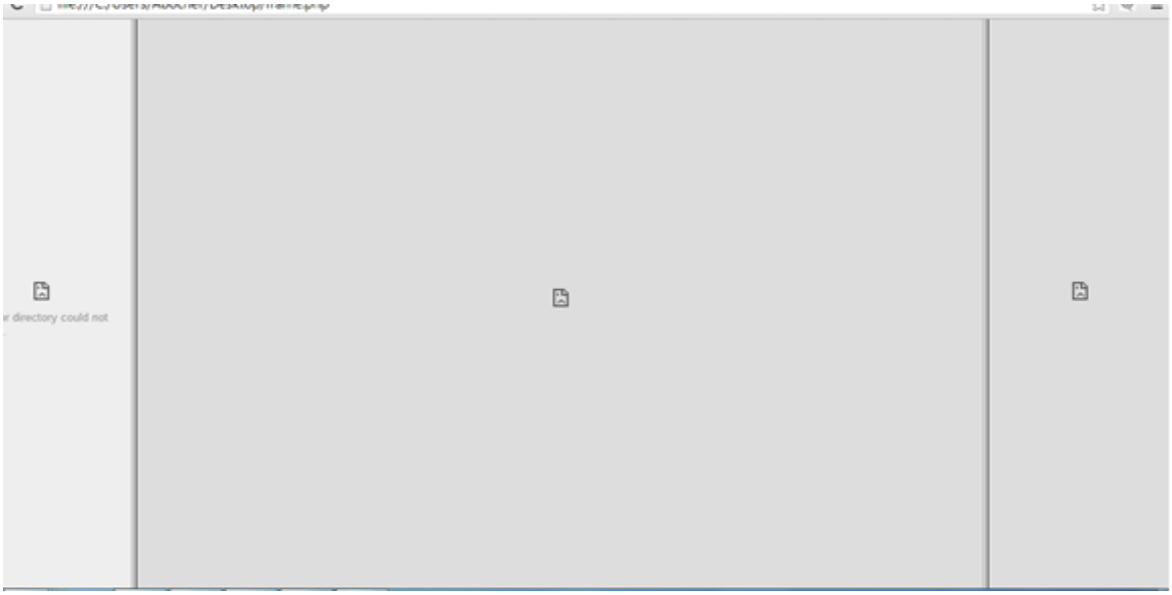
<frameset cols="15%,70%,*" frameborder="1“>

<frame name="list" src="list.htm">

<frame name="main" src="maincontent.htm">

<frame name="advert" src="advert.htm">

</frameset>



ROWS:-Sets the number of frames to create in row format.

Sizecan be set using either an absolute value or a percentage. Example

<frameset rows="15%,70%,*">

<frame name="list" src="list.htm">

<frame name="main" src="maincontent.htm">

<frame name="advert" src="advert.htm">

</frameset>

<Frame> Tag Attribute

SOURCE:-Defines the HTML document to load within that frame.

NAME:-Sets the name of the frame.

ALIGN:-Sets the alignment of the frame and the surrounding text. Valid values are: top, middle, bottom, left, and right.

SCROLLING:-Sets whether scrollbars are used. Valid values are: yes, no, and auto.

With the value auto, the browser will add scrollbars only if the data overruns the space provided.

FRAMEBORDER:-Determines if the border is visible around the frames.

Valid values are 0 and 1 where 0 means not visible.

MARGINHEIGHT:-Sets the amount of space between the top and bottom edges of a frame and its contents.

MARGINWIDTH:-Sets the amount of space between the left and right edges of a frame and its contents.

NORESIZE :- Prevents the user from resizing the frame. This attribute is not assigned a value.

2.8. HTML Charset

HTML Encoding (Character Sets)

To display an HTML page correctly, a web browser must know which character set (character encoding) to use.

What is Character Encoding?

ASCII was the first **character encoding standard** (also called character set). ASCII defined 128 different alphanumeric characters that could be used on the internet: numbers (0-9), English letters (A-Z), and some special characters like ! $ + - ( ) @ < > .

ANSI (Windows-1252) was the original Windows character set, with support for 256 different character codes.

ISO-8859-1 was the default character set for HTML 4. This character set also supported 256 different character codes. Because ANSI and ISO-8859-1 were so limited, HTML 4 also supported UTF-8.

UTF-8 (Unicode) covers almost all of the characters and symbols in the world. The default character encoding for HTML5 is UTF-8.

The HTML charset Attribute

To display an HTML page correctly, a web browser must know the character set used in the page. This is specified in the `<meta>` tag:

For HTML4:

<meta http-equiv="Content-Type" content="text/html;charset=ISO-8859-1">

For HTML5:

<meta charset="UTF-8">

If a browser detects ISO-8859-1 in a web page, it defaults to ANSI, because ANSI is identical to ISO-8859-1 except that ANSI has 32 extra characters.