

# Hypertext Mark-up Language (HTML)

**HTML** is a language capable of generate a web document that can be viewed using a web browser (e.g. Internet Explorer, Netscape, Mozilla, Opera, Safari, etc)

It is composed by elements (mark-up tags) that instruct the browser how to display contents. Since HTML is not the only language that a web browser can understand, we must specify that what we are writing is HTML.

This can be achieved by embedding the HTML command within oblique (angle) brackets `<>`. These are called **Tags** and they tell the browser a **Specific** task to execute. Most tags come in pairs with a beginning and an end, open `<>` and close `</>`.

**Example:** `<html> </html>`

HTML tags are not case sensitive: `<HTML>` means the same as `<html>`. So why should we use lowercase tags when creating our documents? Simply to get ready for next generation of HTML, this is the XHTML. The World Wide Web Consortium (W3C) recommends lowercase tags in their HTML 4 recommendation, and XHTML (the next generation HTML) demands lowercase tags.

## A HTML Document is structured with two sections:

**Head:** Section capable of holding or importing instructions to be executed in the body.

**Body:** Main section which displays the content of the document to the user.

We can also supply to a HTML command more information by adding properties to the tag. These are known as **Attributes**.

**Attributes** always come in name and value pair, such as: `attribute="value"`. Values should always be surrounded by quotes. Double style quotes are the most common used, however single quotes are also allowed.

**Note:** In some instances we might have a value of an attributes that contains a quote itself. In those cases is necessary to use single quotes. `<title='Gary "Shotgun" Smith'>`

**Example:** `<p align="centre">Welcome to HTML language</p>`

`<p></p>` is a tag that creates a paragraph. Note that inside the opening tag there is extra information (attribute) about the element. This tells the browser to display the text or information (in this case "Welcome to HTML language") in the centre of the page.

**Note:** Each tag has specific attributes. **They can only be specified in the opening tag.** If we need to add more that one, they should be separated by one space.

To work with HTML you need:

- An editor such as Notepad, Dreamweaver, etc.
- To save the file with the extension **.htm or .html** (e.g. Home.html)
- A web browser such as Internet Explorer, Netscape, etc.

## HTM vs. HTML Extension

When you save an HTML file, you can use either the .htm or the .html extension. In the past most of the commonly used software only allowed three letter extensions, however this might be a bad practice to get use to. Newer software, which we use today, is now saving HTML files with .html extension by default.

## Comments in Html

Using comment tags you can insert a comment line into HTML source code. Comment will be ignored by the browser. Comments can be used as an explanation of your code, creating references to sections of a HTML document such as marking the beginning and ending you your header, footer, main content and navigation.

**NOTE:** This is a very good habit to get into since it could save you considerable amount of time if you have to edit your HTML in the future.

**Example:** `<!-- this is a comment -->`

# Internet Concepts

**Network:** Collection of computers and other devices (printers, scanners, etc) connected with the purpose of sharing information and resources.

**Internet:** Collection of thousands of networks around the world constitute the International Network

**URL:** Uniform Resource Locator is an alphanumeric string that determines the exact location of a resource and how it should be accessed. URLs are composed by a Protocol and a Domain name.



**Protocol:** Set of rules and regulations that govern transmission between two network components.

- Hypertext Transfer Protocol (HTTP) transfers hypertext and multimedia.
- File Transfer Protocol (FTP) transfers files between the client and server.

**Domain Name:** Unique name that identifies an Internet site.

**Website:** Collection of document design to present information on the Internet. These can have the form of web pages, images, sound, video, etc.

# Document Structure and Top-Level Elements

<pre>&lt;!DOCTYPE&gt; &lt;html&gt;   &lt;head&gt;     &lt;title&gt;Page Title&lt;/title&gt;   &lt;/head&gt;   &lt;body&gt;     Page Contents here...   &lt;/body&gt; &lt;/html&gt;</pre>	<p><b>!DOCTYPE:</b> Document definition.  <b>HTML:</b> Contains the document.  <b>HEAD:</b> Contain document information.  <b>TITLE:</b> Displays document title  <b>BODY:</b> Contains the document contents.</p>
--	--

## <!DOCTYPE>

This mark-up tag should be the very first thing that you include in your HTML document. This is a declaration, that tells the browser, which HTML or XHTML specification document uses.

HTML 4.01 specifies three different types:

- **Traditional DTD\*** – use this definition, when you are allowing inline styling within your HTML code (Some **OLD** browsers do not support CSS style sheets).

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

- **Strict DTD** – use this definition, when you have clean mark-up tags free of inline styling in which case you should have your styles defined using CSS.

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

- **Frameset DTD** – this definition type should be used for documents with frames. The Frameset DTD is equal to the Transitional DTD except for the frameset element replaces the body element.

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

**Note:** If you include other mark-up tags before you document definition, browser displaying your site could enter into Safe mode and might not display your site layout and content correctly!

\* DTD = Data Type Definition

# Head Elements

(Elements inside head should not be displayed by the browser!)

## 1) Title <title></title>

Defines the document title.

### **Example:**

```
<title>Welcome to my first website</title>
```

## 2) Meta <meta>

Provides metadata about the document. (e.g. keywords, description, etc)

Attribute	Value	Description
<b>Required</b>		
content	some_text	Defines meta information to be associated with http-equiv or name
<b>Optional</b>		
http-equiv	content-type, expires, refresh set-cookie	The content attribute to an HTTP header
name	Author, description, keywords generator, revised, others	Defines the type of information contained
scheme	some_text	Defines a format to be used to interpret the value of the content attribute

### **Examples:**

#### **Specify keywords that best describe the page**

```
<meta name="keywords" content="keyword1, keywords2..." />
```

#### **Specify text that should be displayed in search engines**

```
<meta name="description" content="The best website..." />
```

#### **Specify that the page should be index and all pages that are linked from it**

```
<meta name="robots" content="all" />
```

#### **Specify that the page should not be index neither and all pages that are linked from it**

```
<meta name="robots" content="none" />
```

## 3) Base <base>

Defines the document based URL.

Attribute	Value	Description
<b>Required</b>		
href	url	Base URL reference
<b>Optional</b>		
target	_blank, _parent, _self, _top	Frame name to render links in

**Examples:**

```
<base href="http://www.webinfinity.eu /students/webdesign/" />
```

\* Now all links can be define only to the file name to link to such as:

```
<a href="feedback.html">Students Feedback</a>
```

```
<a href="contacts.html">Students Contacts</a>
```

\* Instead of specifying the full URL as follow:

```
<a href="http://www.webinfinity.eu/students/webdesign/feedback.html">
Students Feedback
</a>
```

```
<a href="http://www.webinfinity.eu/students/webdesign/contacts.html">
Students Contacts
</a>
```

#### 4) Link <link>

Defines a relationship with another document where instructions can be accessed and used.

Attribute	Value	Description
<b>All optional</b>		
charset	charset	Defines the character encoding of the target URL. Default value is "ISO-8859-1"
href	URL	Defines the files to link to
hreflang	language_code	Defines the base language of the target URL
media	All, Braille, print, projection, screen speech	Specifies on what device the document will be displayed
rel	Alternate, appendix, bookmark, chapter contents, copyright, glossary, help home, index, next, prev, section, start stylesheet, subsection	Defines the relationship between the current document and the targeted document
rev	As above	Defines the relationship between the targeted document and the current document
target	_blank, _self, _top, _parent	Where to open the target URL.
type	text/css, text/javascript, image/gif	Defines the internet media type of the style language

## Body <body>...</body>

(Displays contents of the page.)

### Optional Attributes

Attribute	Value	Description	
Alink	rgb(x,x,x) #xxxxxx colorname	Specifies the color of the active links in the document.	<b>Deprecated! Use style tag instead or css style sheet definition!</b>
Background	file_name	Displays an image background	
Bgcolor	rgb(x,x,x) #xxxxxx colorname	Sets page background to a colour	
Link	As above	Set links colour	
Text	As above	Sets text colour	
Vlink	As above	Set visited links colour	

#### Note:

All "presentation attributes" of the body element were **deprecated** in HTML 4.01.

All "presentation attributes" of the body element are not supported in XHTML 1.0 Strict DTD.

#### Example:

```
<body>
```

```
Text contents...
```

```
</body>
```

## Generic Block-level Elements

(Block-level elements are those that produce a new line after they are closed)

**Standard or core attributes** - The attributes listed here are the core and language attributes that are standard for all tags (with a few exceptions).

Attribute	Value	Description
class	class_rule or style_rule	Specify the element being member of a style class
id	id_name	Unique identifier of element within the document
style	style_definition	An inline style definition
title	tooltip_text	Mouse over message

### 1) Address <address></address>

Defines contact information such as phone number, email address, postal address, etc.

**Attributes** – see standard/core attribute table

#### Example:

```
<address>
```

97 Uxbridge Road, London<br>  
 Tel: 077 729782 <br>  
 Web: <http://www.webinfinity.eu> email: [info@webinfinity.eu](mailto:info@webinfinity.eu)

</address>

## 2) Blockquote <blockquote></blockquote>

Defines quotation and displays text within indent space from both sides, less space within lines.

**Attributes** - see standard/core attribute table

### Optional Attributes

Attribute	Value	Description
cite	URL	URL of the quote, if it is taken from another source

### Example:

<blockquote cite="http://www.bcoc.co.uk">This creates an indent space</blockquote>

## 3) Deleted <del></del>

Indicates the text inside has been removed.

**Attributes** - see standard/core attribute table

### Optional Attributes

Attribute	Value	Description
cite	URL	Defines a URL to another document which explains why the text was deleted
datetime	YYYYMMDD	Indicates date and time of removal

### Example:

This product new price is <del cite="March special offer">100</del> 75!

## 4) Division <div></div>

Defines a block of HTML elements or text. It is used as a container.

**Attributes** - see standard/core attribute table

### Note:

The "align" attribute of the div element was **deprecated** in HTML 4.01.

The "align" attribute of the div element is not supported in XHTML 1.0 Strict DTD.



**Example:**

```
<div>  
  <p>this is a paragraph</p>  
</div>
```

**5) Heading <h1></h1> <h2></h2> <h3></h3> <h4></h4> <h5></h5> <h6></h6>**

Displays headings of different sizes using bold text, <h1> is the biggest header and <h6> is the smallest header.

**Attributes** - see standard/core attribute table

**Note:**

The "*align*" attribute of the h element was **deprecated** in HTML 4.01.

The "*align*" attribute of the h element is not supported in XHTML 1.0 Strict DTD.

**Example:**

```
<h1>Level-one heading</h1>  
<h2>Level-two heading</h2>  
<h3>Level-three heading</h3>  
<h4>Level-four heading</h4>  
<h5>Level-five heading</h5>  
<h6>Level-six heading</h6>
```

**6) Paragraph <p></p>**

Defines a paragraph.

**Attributes** – see standard/core attribute table

**Note:**

All "*presentation attributes*" and the "*align*" attribute of the p element were **deprecated** in HTML 4.01.

All "*presentation attributes*" and the "*align*" attribute of the p element are not supported in XHTML 1.0 Strict DTD.

**Example:**

```
<p>This is a HTML paragraph</p>
```

**7) Preformatted text <pre></pre>**

Displays text as it is typed in the editor. Makes use of all spaces left in the editor. Uses fixed-pitch font.

**Attributes** - see standard/core attribute table

Note:

The "width" attribute of the pre element was **deprecated** in HTML 4.01.

The "width" attribute of the pre element is not supported in XHTML 1.0 Strict DTD.

### **Example:**

```
<pre>
  The preformatted text      does not ignore spaces.
</pre>
```

## **Lists**

### 1) Ordered List <ol>...</ol> and List item <li>...</li>

Displays a numbered list

**Attributes** - see standard/core attribute table

#### **Optional Attributes**

Attribute	Value	Description	Deprecated! Use style tag instead or css style sheet definition!
compact	compact_rendering	Deprecated.	
start	start_on_number	Specifies the number to start on.	
type	A, a, I, I, 1	Specifies the type of the list.	

### **Example:**

```
<ol>
  <li>First</li>
  <li>Second</li>
</ol>
```

### 2) Unordered List <ul>...</ul> and List item <li>...</li>

Displays a list with bullets type as default

**Attributes** - see standard/core attribute table

#### **Optional Attributes**

Attribute	Value	Description	Deprecated! Use style tag instead or css style sheet definition!
compact	compact_rendering	Deprecated.	
type	Disc, square, circle	Specifies the type of the list.	

### **Example:**

```
<ul>
  <li>First</li>
  <li>Second</li>
</ul>
```

### 3) Definition List <dl>...</dl>

**Definition Term** <dt>...</dt>  
**Definition Descriptor** <dd>...</dd>

Displays a term and its definition

**Attributes** - see standard/core attribute table

**Example:**

```
<dl>
  <dt>HTML</dt>
  <dd>Hypertext Markup Language</dd>

  <dt>HTTP</dt>
  <dd>Hypertext Transfer protocol</dd>
</dl>
```

## Inline-Level Elements

### Horizontal Rule <hr>

Displays a line across the browser

**Attributes** - see standard/core attribute table

**Note:**

All "*presentation attributes*" of the body element were **deprecated** in HTML 4.01.

All "*presentation attributes*" of the body element are not supported in XHTML 1.0 Strict DTD.

**Example:** <hr />

### Marquee <marquee>...</marquee>

Displays text moving across the browser

**Attributes**

Attribute	Value	Description
behaviour	scroll, slide, alternate	Sets the stop point after looping
direction	left, right	Sets in which direction it goes towards to
loop	number	sets the number of display cycles
scrollamount	number	controls the amount of movement (in pixels) between the successive displays that give the impression of animation
scrolldelay	number	controls the delay (in milliseconds) between the

		successive displays that give the impression of animation
align	top, middle, bottom	controls the positioning of the marquee display box relative to the current text
width	%, pixel	Sets the width of display box

**Example:** `<marquee direction="right" loop="2">Welcome to my site</marquee>`

## Phrase elements

Source	Output
<code>&lt;em&gt;Emphasized text&lt;/em&gt;</code>	<i>Emphasized text</i>
<code>&lt;strong&gt;Strong text&lt;/strong&gt;</code>	<b>Strong text</b>
<code>&lt;dfn&gt;Definition term&lt;/dfn&gt;</code>	<i>Definition term</i>
<code>&lt;cite&gt;Citation&lt;/cite&gt;</code>	<i>Citation</i>

**Emphasis** `<em>...</em>` Displays italic text

**Attributes** - see standard/core attribute table

**Example:** `<em>Hello</em>`

**Strong** `<strong>...</strong>` Displays bold text

**Attributes** - see standard/core attribute table

**Example:** `<strong>Hello</strong>`

## Font style elements

Source	Output
<code>&lt;i&gt;Italic text&lt;/i&gt;</code>	<i>Italic text</i>
<code>&lt;b&gt;Bold text&lt;/b&gt;</code>	<b>Bold text</b>
<code>&lt;tt&gt;Teletype text&lt;/tt&gt;</code>	Teletype text
<code>&lt;small&gt;Small text&lt;/small&gt;</code>	Small text
<code>&lt;big&gt;Big text&lt;/big&gt;</code>	Big text

**Italic** `<i>...</i>` Displays italic text

**Attributes** - see standard/core attribute table

**Example:** `<i>Hello</i>`

**Bold** `<b>...</b>` Displays bold text

**Attributes** - see standard/core attribute table

**Example:** `<b>Hello</b>`

**Typewriter text** `<tt>...</tt>` Displays text as typed in typewriting

**Attributes** - see standard/core attribute table

**Example:** `<tt>Hello this is like text typed in books</tt>`

**Small** `<small>...</small>` Displays text of size 1

**Attributes** - see standard/core attribute table

**Example:** `<small>small text of size 1</small>`

**Big** `<big>...</big>` Displays text of size 4

**Attributes** - see standard/core attribute table

**Example:** `<big>big text of size 4</big>`

**Superscript** `<sup>...</sup>`

Displays superscript text slightly above of previous word

**Attributes** - see standard/core attribute table

**Example:** `Bickenhall College<sup>TM</sup>`

**Subscript** `<sub>...</sub>`

Displays subscript text slightly below of previous word

**Attributes** - see standard/core attribute table

**Example:** `H<sub>2</sub>O`

## Image <img>

Displays an image

**Attributes** - see standard/core attribute table

Attribute	Value	Description
src	URL	The URL of the image to display
alt	text	Short description of the image
<b>Optional</b>		
width	pixels, %	Sets the width of an image
height	pixels, %	Sets the height of an image
border	pixels	Sets a border around an image.
vspace	pixels	Sets white space on the top and bottom of the image.
hspace	pixels	Sets white space on the left and right side of the image.
align	top, bottom, middle, left right	Specifies how to align the image according to surrounding text.

Deprecated! Use style tag instead or css style sheet definition!

**Example:** 

## Multimedia <embed>

Plays sound or displays a video

**Attributes** - see standard/core attribute table

Attribute	Value	Description
Src		Defines the source of multimedia file
Width	pixels	Sets the width for Windows media player
Height	pixels	Sets the height for Windows media player
loop	true, false, number	Sets the number of times the file should play
Volume	1/10	Sets the volume
Controller	true/false	Shows or hides the play, backward, forward buttons

**Example:** <embed src="x.mpg" width="300" height="300" />

## Link (Anchor) <a>...</a>

Create a link to a specific location. This can be in the current site in which only the file name to navigate to is required. We can create a link to an external site by using href attribute. An internal link can also be created by having a the # key follow by a name (target) on the page .

**Attributes** - see standard/core attribute table

Attribute	Value	Description
href	text	Sets the file target
target	_blank, _self, _top, _parent	Where to open the target URL.
name*	field_name	Sets a unique name for the input element.

**\*Note:**

In the future versions of XHTML name attribute will be replaced by Id.

**Examples:**

```

<a href=" PageName.html" target="_blank">Click Here</a> *(opens in a new window)
<a href="http://www.webinfinity.eu"> College Site</a> *(opens in the current widow)
<a href="#menu">View menu</a>
.
.
.
<a name="menu">This link will take you to a particular part of a long page, where ever
you place this link reference</a>

```

## Tables

```

<table>
  <tr>
    <th> Creates a table header cell. The text within will be centred and bold </th>
  </tr>
  <tr>
    <td> Creates a table cell </td>
  </tr>
</table>

```

Tables are the most important element in HTML. They allow us to design any type of layout. If you have used MS Word you should know what a table looks like. Its structure is composed by Columns and Rows. "TH" is Table Heading, "TR" is Table Row and "TD" is Table Data.

**Examples:**

Paragraph 1	Paragraph 2
-------------	-------------

```

<table>
  <tr>
    <td>Paragraph 1</td>
    <td>Paragraph 2</td>
  </tr>
</table>

```

Heading
Paragraph 1
Paragraph 2

```

<table>
  <tr>
    <th>Heading</th>
  </tr>
  <tr>
    <td>Paragraph 1</td>
  </tr>
  <tr>
    <td>Paragraph 2</td>
  </tr>
</table>

```

Paragraph 1	Paragraph 2
Paragraph 3	Paragraph 4

```
<table>
  <tr>
    <td>Paragraph 1</td>
    <td>Paragraph 2</td>
  </tr>
  <tr>
    <td>Paragraph 3</td>
    <td>Paragraph 4</td>
  </tr>
</table>
```

## Merging Cells

Top	
Left	Right

```
<table>
  <tr>
    <td colspan="2">Top</td>
  </tr>
  <tr>
    <td>Left</td>
    <td>Right</td>
  </tr>
</table>
```

Top Left	Right
Bottom Left	

```
<table>
  <tr>
    <td>Top Left</td>
    <td rowspan="2">Right</td>
  </tr>
  <tr>
    <td>Bottom Left</td>
  </tr>
</table>
```

## Nested tables (tables within tables)

Paragraph 1
-------------

```
<table>
  <tr>
    <td>Paragraph 1</td>
  </tr>
  <tr>
    <td>
      <!-- Start Nested table -->
      <table>
        <tr>
          <td>Paragraph 1</td>
```



Paragraph 1	Paragraph 2
Paragraph 3	Paragraph 4

```

        <td>Paragraph 2</td>
    </tr>
    <tr>
        <td>Paragraph 3</td>
        <td>Paragraph 4</td>
    </tr>
</table>

<!-- End Nested table -->

</td>
</tr>
</table>

```

**Note:** Please remember that to produce a layout you must use tables and nested tables. This can only be created inside a TD (table data) and not elsewhere. You should be able to design any page structure using this HTML element.

**Attributes** - see standard/core attribute table

## Table

Attribute	Value	Description
align	left, center, right	Aligns the table.
bgcolor	rgb(x,x,x), #xxxxxx, colorname	Sets the background color of the table.
border	pixels	Sets the border width.
cellpadding	pixels, %	Sets the space between the cell walls and contents
cellspacing	pixels, %	Sets the space between cells
frame	void, above, below, hside, lhs, rhs, vsides, box, border	Sets how the outer borders should be displayed. <b>Note:</b> Must be used in conjunction with the "border" attribute!
rules	none, groups, rows, cols, all	Sets the horizontal/vertical divider lines. <b>Note:</b> Must be used in conjunction with the "border" attribute!
summary	text	Sets a summary of the table for speech-synthesizing/non-visual browsers
width	%, pixels	Sets the width of the table

## Table Row

Attribute	Value	Description
align	right, left, center, justify, char	Defines the text alignment in cells
bgcolor	rgb(x,x,x), #xxxxxx, colorname	Specifies the background color of the table cell.
valign	top, middle, bottom, baseline	Specifies the vertical

## Table Heading and Table Data

Attribute	Value	Description
valign	top, middle, bottom, baseline	Sets the vertical alignment of cell content
align	left, right, center, justify, char	Sets the horizontal alignment of cell content

colspan	number	Indicates the number of columns this cell should span	<b>Deprecated! Use style tag instead or css style sheet definition!</b>
rowspan	number	Indicates the number of rows this cell should span	
bgcolor	rgb(x,x,x), #xxxxxx, colorname	Sets the background color of the table cell.	
height	pixels	Sets the height of the table cell.	
width	pixels, %	Sets the width of the table cell.	
nowrap	nowrap	Whether to disable or enable automatic text wrapping in this cell.	

## Form <form>...</form>

An HTML form is an element capable of sending data from the client to the server.

### Attributes:

Attribute	Value	Description
action	URL	Sets the target file where the information is sent
method	get, post	Specify how the data should be sent. Default is get.
name	form_name	Creates a unique identifier

### **Note:**

**method = "get":** This method sends the form contents in the URL: URL?name=value&name=value. If the form values contains non-ASCII characters or exceeds 100 characters you **MUST** use method="post"!

**method = "post":** This method sends the form contents in the body of the request. Most browsers are unable to bookmark post requests!

### Example:

```
<form name="registration" action="confirmation.html" method="post">
```

The form is composed by other elements (form objects) that allow a user to input data. These are the following:

## Text <input type="text">

Allows the user to enter a single line of text

**Attributes** - see standard/core attribute table

Attribute	Value	Description
type	input_type	Indicates the type of the input element. The default value is "text"
name	field_name	Creates a unique name for the input element.
maxlength*	number	Defines the maximum number of characters allowed in a text field. *Only used with type="text"
value	text	Displays text inside the box, default value of the element.
	readonly	Sets only read capability (contents can't be changed)

		*Only used with type="text"
size	number_of_char	Sets the size of the input element.

**Example:** `<input type="text" name="CustName" size="20" />`

## Password `<input type="password">`

Allows the user to enter a single line of text but does only display

**Attributes** - see standard/core attribute table

Same as above

**Example:** `<input type="password" name="CustPwd" size="20" />`

## Text Area `<textarea></textarea>`

Allows the user to enter large amount of text

**Attributes** - see standard/core attribute table

Attribute	Value	Description
cols	number	Sets the number of columns to display
rows	number	Sets the number of rows to display
<b>Optional</b>		
name	name_of_textarea	Creates an unique identifier
	readonly	Indicates that the user cannot modify the content in the text-area

**Example:** `<textarea name="CustPwd" cols="3" rows="20"></textarea>`

## Radio `<input type="radio">`

Allows the user have single selection

**Attributes** - see standard/core attribute table

Attribute	Value	Description
type	input_type	Indicates the type of the input element. The default value is "radio"
name	field_name	Creates a unique name for the input element.
value	text	Defines the result of the input element when clicked. The result is sent to the form's action URL.
	checked	Indicates that the input element should be checked when it first loads.

**Example:** `<input type="radio" name=" CustomerGender" Value="Female" />Female`

## Checkbox `<input type="checkbox">`

Allows the user to have multiple selections

**Attributes** - see standard/core attribute table

Same as above except the type="checkbox"

**Example:** `<input type="checkbox" name=" CustomerDegree" Value=" Degree" />Degree`

## File <input type="file">

Allows the user to attach a file

**Attributes** - see standard/core attribute table

Attribute	Value	Description
type	input_type	Indicates the type of the input element. The default value is "file"
name	field_name	Displays text inside the box, default value of the element.
size	number_of_char	Sets the size of the input element.
value	text	Displays text inside the box, default value of the element.

**Example:** <input type="file" name="CustAtt" size="40" />

## Select <select><optgroup><option></option><optgroup></select>

Allows the user to select from a drop down list

**Attributes** - see standard/core attribute table

Attribute	Value	Description
name	field_name	Displays text inside the box, default value of the element.
size	number	Sets the number of visible items in the drop-down list
multiple	multiple	Specifies that multiple items can be selected at a time

**Example:** <select name="CustomerCountry">  
 <option value="">Select</option>  
 <optgroup label="European Countries">  
 <option value="UK">UK</option>  
 <option value="France">France</option>  
 </optgroup>  
 <optgroup label="Asian Countries">  
 <option value="India">India</option>  
 <option value="China">China</option>  
 </optgroup>  
 </select>

## Submit <input type="submit">

Button that allows submitting the form

**Attributes** - see standard/core attribute table

Attribute	Value	Description
type	input_type	Indicates the type of the input element. The default value is "submit"
name	field_name	Sets a unique name for the input element.
value	text	Sets the text on the button.

disabled

Disables the button

**Example:** `<input type="submit" name="sb" value="Submit Form" />`

## Reset `<input type="reset">`

Button that allows resetting the form

**Attributes** - see standard/core attribute table  
- see submit input type

**Example:** `<input type="reset" name="reset" value="Reset Form" />`

## Frame

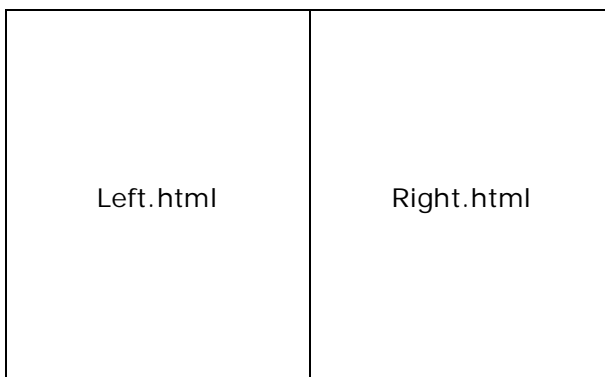
`<frameset>` `<frame>`...`</frame>``</frameset>`

This tag allows creating a container to display multiple files at the same time. Unlike a normal HTML page, which has a body, frames do not need this element as they only serve as a container. Consider the following examples.

### Examples:



```
<html>
  <head>
    <title>Frame1.html</title>
  </head>
  <frameset rows="50%,50%">
    <frame src="Top.html" />
    <frame src="Bottom.html" />
  <noframes>
    <body>
      Your Browser Does not support frames
    </body>
  </noframes>
</frameset>
</html>
```



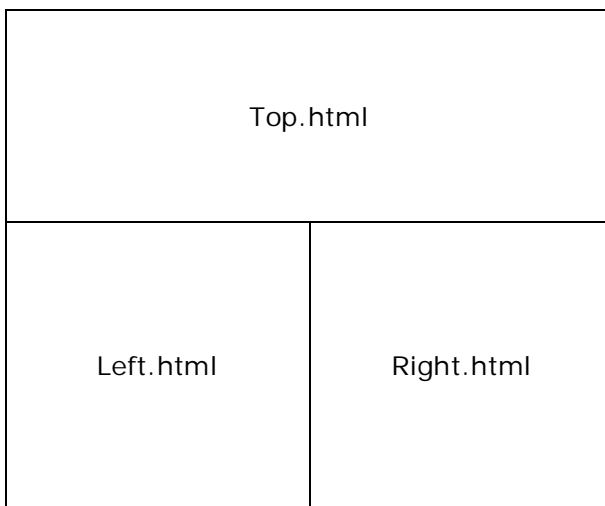
```
<html>
  <head>
    <title>Frame2.html</title>
  </head>
  <frameset cols="50%,50%">
    <frame src="Left.html" />
    <frame src="Right.html" />
  <noframes>
    <body>
      Your Browser Does not support frames
    </body>
  </noframes>
</frameset>
</html>
```

```

    </noframes>
</frameset>
</html>

```

## Nested Frames (Frameset within another)



```

<html>
  <head>
    <title>NestedFrames.html</title>
  </head>
  <frameset rows="50%,*">
    <frame src="Top.html" />
    <frameset cols="50%,*">
      <frame src="Left.html" />
      <frame src="Right.html" />
    </frameset>
  </frameset>
  <noframes>
    <body>
      Your Browser Does not support frames
    </body>
  </noframes>
</frameset>
</html>

```

**Note:** The \* key represents "everything" or "All" and its widely used in HTML Frame.

### Frameset Attributes

Attribute	Value	Description
cols	pixel, %, *	Sets the size of the column
rows	pixel, %, *	Sets the size of the row

### Frame Attributes

Attribute	Value	Description
src	URL	Defines the URL of the file to show in the frame
name	frame_name	Defines a unique name for the frame (to use in scripts)
frameborder	0, 1	Specifies whether or not to display border around the frame
scrolling	yes, no, auto	Determines scrollbar action
	noresize	When set to noresize the user cannot resize the frame

**<NoFrames><body>...</body></NoFrames>**

Some **old** browsers do not support frames. Inside this tag we can display a message to those.

## Iframe <iframe>...</iframe>

This tag allows embedding an external file within the current one. They are also known as floating frames.

### Attributes

Attribute	Value	Description
src	URL	The URL of the document to show in the iframe
name	frame_name	Specifies a unique name of the iframe (to use in scripts)
frameborder	1 ,0	Specifies whether or not to display a frame border
scrolling	Yes, no ,auto	Define scroll bars
height	Pixels, %	Sets the height of the iframe
width	Pixels, %	Sets the width of the iframe
align	Left, right, top, middle,bottom	Specifies how to align the iframe

**Example:** <iframe src="http://www.webinfinity.eu" width="200" height="300"></iframe>

### HTML Tags being deprecated –

- <centre>
- <s> & s<strike>
- <font>
- <u>

## Why HTML 4.01 and what is the purpose of using correct tags to layout your html documents?

"The Semantic Web approach develops languages for expressing information in a machine processable form" (W3C)

The line above is a quote, so it resides in a pair of <blockquote> tags. This tag let's a screen reader know that the text is a quote and not just another paragraph. Screen readers are machines that enable visually impaired people to surf the web. Creating a **semantically correct document** in its simplest form is just one case of using the HTML 4.01 (XHTML 1.0) elements supplied with the specification to layout your document correctly. Each element provided in the HTML 4.01 (XHTML 1.0) specification has been designed to be used

in a specific way and each has a specific meaning. A h (heading) element is a title and any text between title tags is instantly recognized as a title by a machine reader, just as text within <p> tags is recognized as being a paragraph. Not all tags are semantic in their makeup. Take the <span> tag, for instance. A <span> tag has no meaning associated with it; it is simply a container element. If you were using a <span> tag and styled it to emphasize text within a paragraph, you would be better off using the <em> tag. This is what it was designed to do. It says to the machine reader *this text should be emphasized*; in turn, the machine reader emphasizes the text within the <em> tags.