

# HTML-Hyper Text Markup Language

## What is HTML?

HTML is a language for describing web pages.

- HTML stands for **H**yper **T**ext **M**arkup **L**anguage
- HTML is a **markup** language
- A markup language is a set of markup **tags**
- The tags **describe** document content
- HTML documents contain HTML **tags** and plain **text**
- HTML documents are also called **web pages**

**Hypertext means:** Hypertext is text which contains links to other texts.

**Markup means:** Markup languages are designed for the processing, definition and presentation of text. The language specifies code for formatting, both the layout and style, within a text file. The code used to specify the formatting are called tags. HTML is an example of a widely known and used markup language.

## Elements and tags

Elements give structure to a HTML document and tell the browser how you want your website to be presented. Generally elements consist of a start tag, some content, and an end tag.

## "Tags"?

Tags are labels you use to mark up the beginning and end of an element.

All tags have the same format: they begin with a less-than sign "<" and end with a greater-than sign ">".

Generally speaking, there are two kinds of tags - opening tags: <html> and closing tags: </html>. The only difference between an opening tag and a closing tag is the forward slash "/". You label content by putting it between an opening tag and a closing tag.

Example

```
<h1>this is a heading</h1>  
    <h2>this is a subheading</h2>
```

## What is an attribute?

As you probably remember, elements give structure to a HTML document and tells the browser how you want your website to be presented (for example, `<br />` informs the browser to make a line break). In some elements you can add more information. Such additional information is called an attribute.

Attributes are always written within a start tag and are followed by an equals sign

### **Should tags be typed in uppercase or lowercase?**

Most browsers might not care if you type your tags in upper, lower or mixed cases. `<HTML>`, `<html>` or `<HtMl>` will normally give the same result. However, the **correct** way is to type tags in lowercase. So get into the **habit of writing your tags in lowercase**.

### **The `<html>` Element:**

The `<html>` element is the containing element for the whole HTML document. Each HTML document should have one `<html>` and each document should end with a closing `</html>` tag.

Following two elements appear as direct children of an `<html>` element:

- `<head>`
- `<body>`

As such, start and end HTML tags enclose all the other HTML tags you use to describe the Web page.

### **The `<head>` Element:**

The `<head>` element is just a container for all other header elements. It should be the first thing to appear after the opening `<html>` tag.

Each `<head>` element should contain a `<title>` element indicating the title of the document, although it may also contain any combination of the following elements, in any order:

- The `<base>` tag is used to create a "base" url for all links on the page. Check HTML Base tag.
- The `<object>` tag is designed to include images, JavaScript objects, Flash animations, MP3 files, QuickTime movies and other components of a page. Check HTML Object tag.
- The `<link>` tag is used to link to an external file, such as a style sheet or JavaScript file. Check HTML Link tag.
- The `<style>` tag is used to include CSS rules inside the document. Check HTML Style tag.

- The <script> tag is used to include JAVAScript or VBScript inside the document. Check HTML Script tag.
- The <meta> tag includes information about the document such as keywords and a description, which are particularly helpful for search applications.

## **Marquee Tag**

The HTML <marquee> tag is used for scrolling piece of text or image displayed either horizontally across or vertically down your web site page depending on the settings.

### **Example:**

```
<marquee>This is basic example of marquee</marquee>
```

### **Attributes:**

Attribute	Value	Description
behavior	scroll slid alternate	Defines the type of scrolling.
bgcolor	colorname	Deprecated - Specifies the background color.
direction	up down left right	Defines the direction of scrolling the content.
height	pixels or %	Defines the height of marquee.
hspace	Pixels	Specifies horizontal space around the marquee.
loop	number	Specifies how many times to loop. The default value is INFINITE, which means that the marquee loops endlessly.
scrolldelay	seconds	Defines how long to delay between each jump.

scrollamount	number	Defines how far to jump.
width	pixels or %	Defines the width of marquee.
vspace	Pixels	Specifies vertical space around the marquee.

Eg:

```
<marquee behavior="slide" direction="up" height="50" hspace="5" width="50"
vspace="5">Hello</marquee>
```

## The <title> Element:

You should specify a title for every page that you write inside the <title> element. This element is a child of the <head> element). It is used in several ways:

- It displays at the very top of a browser window.
- It is used as the default name for a bookmark in browsers such as IE and Netscape.
- Its is used by search engines that use its content to help index pages.

Example:-

```
<html>
<head><title>First web page</title></head>
</html>
```

## The <script> Tag

The <script> tag is used to define a client-side script, such as a JavaScript.

The <script> element either contains scripting statements, or it points to an external script file through the src attribute.

### Attributes:

- src - This attribute specifies the location of an external script.
- type - This attribute specifies the scripting language of the element's contents and overrides the default scripting language

Example:

```
<head>  
<script type="text/javascript">  
</script></head>
```

## **The <body> Element:**

The <body> element appears after the <head> element and contains the part of the Web page that you actually see in the main browser window, which is sometimes referred to as body content.

A <body> element may contain anything from a couple of paragraphs under a heading to more complicated layouts containing forms and tables.

### **Attributes of Body Tag:**

- **Bgcolor:** To change the background color of the webpage
- **Text:** to change the text color
- **Background:** Specifies a background image for a document
- **Alink:** Specifies the color of an active link in a document
- **Link:** Specifies the color of unvisited links in a document
- **Vlink:** Specifies the color of visited links in a document

### **Example Code:**

```
<body text="green" bgcolor="blue"  
background="file:///C:/Documents%20and%20Settings/All%20Users/Documents/My%20  
Pictures/Sample%20Pictures/Blue%20hills.jpg">Hello</body>
```

## **Create Headings - The <h1> Elements:**

Any document starts with a heading. You use different sizes for your headings. HTML also have six levels of headings, which use the elements <h1>, <h2>, <h3>, <h4>, <h5>, and <h6>. While displaying any heading, browser adds one line before and after that heading.

Example:

```
<h1>This is heading 1</h1>  
<h2>This is heading 2</h2>
```

```
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>
```

## **PARAGRAPH TAG**

The paragraph tags are used to define a block of text as a paragraph. This HTML element is one of the basic HTML tags you should learn to use properly.

When a block of text is surrounded by the paragraph tags, the browser automatically adds white space before and after the paragraph.

### **Using the Paragraph Tag**

To define a paragraph, you use the opening and closing paragraph set of tags

#### **Example:**

```
<p>The contents of the paragraph.</p>
```

Paragraphs can be formatted in HTML much the same as you would expect to find in a word processing program.

```
<p align="justify">For instance, let's say you had a HUGE school or work...</p>
<p align="center">For instance, let's say you had a HUGE school or work...</p>
```

## **HTML Comments**

Comments can be inserted into the HTML code to make it more readable and understandable. Comments are ignored by the browser and are not displayed.

Comments are written like this:

#### **Example**

```
<!-- This is a comment -->
```

## **HTML Line Breaks**

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

### Example

```
<p>This is<br />a para<br />graph with line breaks</p>
```

## HTML Text Formatting Tags

Tag	Description
<b>	Defines bold text
<big>	Defines big text
<em>	Defines emphasized text
<i>	Defines italic text
<small>	Defines small text
<strong>	Defines strong text
<sub>	Defines subscripted text
<sup>	Defines superscripted text
<pre>	Defines the preformatted Text

### Example:

```
<html>
<head><title>Text formatting Tags</title></head>
<body>
<b>This text is bold</b><br><i>This text is italic</i><br><u>This text is
underlined</u><br>
<big>This text is bigger in size</big><br><small>This text is smaller in
size</small><br>
<em>This text is emphasized</em><br> ax<sup>2</sup>+bx+c=0<br>
H<sub>2</sub>SO<sub>4</sub><br><pre>
    The text is preformatted</pre><br></body>
```

**This text is bold**

*This text is italic*

This text is underlined

**This text is bigger in size**

This text is smaller in size

*This text is emphasized*

$ax^2+bx+c=0$

H<sub>2</sub>SO<sub>4</sub>

The text is preformatted

## Special Characters used in HTML

Characters	Symbol Displayed	Description
&ndash;	-	Used to show hyphen
&quot;	“	Used to display Quote
&nbsp;		to provide Space
&amp;	&	to display ampersand
&divide;	divide sign	To display the division sign
&lt;	<	To display less than sign
&laquo;	<<	
&raquo;	>>	
&gt;	>	greater than sign
&euro;	Euro sign	To display the euro
&pound;	Pound sign	To display the pound
&plusmn;	+ -	To display the Plusminus
&cent;	cent	To display the cent

## INTRODUCTION TO LISTS

HTML offers several mechanisms for specifying lists of information. All lists must contain one or more list elements. Lists may contain:

- Unordered information.
- Ordered information.
- Definitions.
- Menu
- Directory



## **HTML Unordered Lists**

An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.

The list items are marked with bullets (typically small black circles).

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

How the HTML code above looks in a browser:

- Coffee
- Milk

## **HTML Ordered Lists**

An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.

The list items are marked with numbers.

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

How the HTML code above looks in a browser:

1. Coffee
2. Milk

## **HTML Definition Lists**

A definition list is a list of items, with a description of each item.

The <dl> tag defines a definition list.

The <dl> tag is used in conjunction with <dt> (defines the item in the list) and <dd> (describes the item in the list):

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

How the HTML code above looks in a browser:

Coffee

- black hot drink

Milk

- white cold drink

## **HTML Menu Lists**

This tag defines menu/list of commands. It is used for context menus, toolbars and for listing form controls and commands.

Syntax:

```
<menu>
  <li>File</li>
  <li>Edit</li>
</menu>
```

## **HTML Directory Lists**

It is used to list directory titles

Syntax:

```
<dir>
  <li>HTML</li>
  <li>XML</li>
  <li>CSS</li>
</dir>
```

Lists may also be nested and different list types may be used together, as in the following example, which is a definition list that contains an unordered list (the ingredients) and an ordered list (the procedure):

## **HTML List Tags**

Tag	Description
<ol>	Defines an ordered list
<ul>	Defines an unordered list
<li>	Defines a list item
<dl>	Defines a definition list
<dt>	Defines an item in a definition list
<dd>	Defines a description of an item in a definition list

## **HTML Images - The <img> Tag and the Src Attribute**

In HTML, images are defined with the <img> tag.

The <img> tag is empty, which means that it contains attributes only, and has no closing tag.

To display an image on a page, you need to use the src attribute. Src stands for "source". The value of the src attribute is the URL of the image you want to display.

### **Syntax for defining an image:**

``

The URL points to the location where the image is stored. An image named "boat.gif", located in the "images" directory on "www.w3schools.com" has the URL:

<http://www.w3schools.com/images/boat.gif>.

The browser displays the image where the <img> tag occurs in the document. If you put an image tag between two paragraphs, the browser shows the first paragraph, then the image, and then the second paragraph.

### **HTML Images - The Alt Attribute**

The required alt attribute specifies an alternate text for an image, if the image cannot be displayed.

The value of the alt attribute is an author-defined text:

``

The alt attribute provides alternative information for an image if a 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).

### **HTML Images - Set Height and Width of an Image**

The height and width attributes are used to specify the height and width of an image.

The attribute values are specified in pixels by default:

```

```

**Note:** It is a good practice to specify both the height and width attributes for an image. If these attributes are set, the space required for the image is reserved when the page is loaded. However, without these attributes, the browser does not know the size of the image. The effect will be that the page layout will change during loading (while the images load).

### **Image Attributes:**

Following are most frequently used attributes for <img> tag.

- **width:** sets width of the image. This will have a value like 10 or 20%etc.
- **height:** sets height of the image. This will have a value like 10 or 20% etc.
- **border:** sets a border around the image. This will have a value like 1 or 2 etc.
- **src:** specifies URL of the image file.
- **alt:** this is an alternate text which will be displayed if image is missing.
- **align:** this sets horizontal alignment of the image and takes value either *left*, *right* or *center*.
- **valign:** this sets vertical alignment of the image and takes value either *top*, *bottom* or *center*.
- **hspace:** horizontal space around the image. This will have a value like 10 or 20%etc.
- **vspace:** vertical space around the image. This will have a value like 10 or 20%etc.
- **name:** name of the image with in the document.

### **Example Image code:**

```

```

**Output:**



## **HTML-image links**

Image links are constructed as you might expect, **by embedding an <img> tag inside of an anchor element <a>**. Like HTML text links, image links require opening and closing anchor tags, but instead of placing text between these opening and closing tags, the developer needs to place an image tag -- with a valid source attribute value of course.

### **HTML Image Link Code:**

```
<html>
<body>
<a href="http://www.google.com" target="_blank">

</a>
</body>
</html>
```

By default, many browsers add a small border around image links. This default behavior is intended to give web viewers the ability to quickly decipher the difference between ordinary images and image links.

### **HTML Image Border Code:**

```
<html>
<body>
<a href="http://www.google.com" target="_blank">

</a>
</body>
</html>
```

## **HYPERLINKS**

Anchor tag <a> defines a hyperlink. A hyperlink or a link is word, group of words, or image that we can click to jump to other document. In most browsers, an unvisited link is underlined and blue, a visited link is underlined and purple, and an active link is underlined and red.

### **Attributes of anchor tag:**

- **href:** Specifies the URL of the page the link goes to.
- **Name:** Specifies the name of an anchor

- **Target:** Specifies where to open the linked document.

### **HTML - hypertext reference (href)**

A Hypertext Reference (*href*) is an HTML attribute of an anchor (link) tag that requires a valid URL in order to properly direct a user to a different location. In other words, this Hypertext Reference is where users will navigate to if they do click on this link.

### **HTML Text Link Code: (EXTERNAL LINK)**

```
<html>
<head><title>Link Window</title></head>
<body>
<a href="C:\Users\Dell\Desktop\html\page1.html" target="frame3">What is Computer?</a>
<a href="C:\Users\Dell\Desktop\html\page2.html" target="C:\Users\Dell\Desktop\html\mainarea.html">Input
Devices</a>
<a href="C:\Users\Dell\Desktop\html\page3.html" target="C:\Users\Dell\Desktop\html\test.html">Output
Devices</a>
</body>
</html>
```

URLs, and each has a slightly different look. The examples above link to what are known as Global URLs

### **HTML - link targets**

The target attribute defines how each link will open when clicked. Will each one open in a new window, or will each one open in the current browser window? As the web designer, you call the shots as to how a user navigates from page to page, so long as you know how to handle the target attribute.

### **Link Targets:**

<b>Target=</b>	<b>Description</b>
_blank	Opens new page in a new browser window
_self	Loads the new page in the current window
_parent	Loads new page into a parent frame
_top	Loads new page into the current browser window, cancelling all frames

## **HTML - email links**

HTML `<a>` tag provides you option to specify an email address to send an email. While using `<a>` tag as an email tag, you will use `mailto:email address` along with *href* attribute. Following is the syntax of using `mailto` instead of using `http`.

### **HTML Email Link Code:**

```
<a href= "mailto:abc@example.com">Send Email</a>
```

---

[Send Email](#)

Now if a user clicks this link, it launches one Email Client (like Lotus Notes, Outlook Express etc.) installed on your user's computer. There is another risk to use this option to send email because if user do not have email client installed on their computer then it would not be possible to send email.

### **Default Settings**

You can specify a default *email subject* and *email body* along with your email address. Following is the example to use default subject and body.

```
<a href="mailto:kanikajethwani@gmail.com?cc=sav@yahoo.com&bcc=abc@gmail.com&subject=test subject&body=I learned to create email link code on this website.">email me</a>">Email Me</a>
```

You can specify the subject, cc, bcc by using the above code.

## **INTERNAL LINKS**

You can also create internal links within a page - for example a table of contents at the top with links to each chapter below. All you need to use is a very useful attribute called `id` (identification) and the symbol `"#"`.

Use the `id` attribute to mark the element to which you want to link. For example:

```
<html>
<body>
<p><a href="#heading1">Link to heading 1</a></p>
<p><a href="#heading2">Link to heading 2</a></p>
<h1 id="heading1">heading 1</h1>
<p>Text text text text</p>
<h1 id="heading2">heading 2</h1>
<p>Text text text text</p>
</body>
</html>
```

### **Output:-**

---

[Link to heading 1](#)

[Link to heading 2](#)

## **heading 1**

Text text text text

## **heading 2**

Text text text text

### **HTML – Tables**

An HTML table is an element comprised of table rows and columns, much like you'd see when working with an application such as Excel. Tables are container elements, and their sole purpose is to house other HTML elements and arrange them in a tabular fashion -- row by row, column by column.

Tables may seem difficult at first, but after working through this lesson, you'll see that they aren't so horrible. A table element consists of three different HTML tags including the <table> tag, <tr> (table rows), and the <td> (table columns) tags.

#### **Table Tag**

The HTML table model allows authors to arrange data -- text, preformatted text, images, links, forms, form fields, other tables, etc. -- into rows and columns of cells.

Each table may have an associated caption (the CAPTION element) that provides a short description of the table's purpose.

Table cells may either contain "header" information (TH element) or "data" (TD element).

#### **Syntax:**



```

<table>
<tr>
  <th>S.No</th>
  <th>Name</th>
  <th>Course</th>
</tr>
<tr>
  <td>1</td>
  <td>ABC</td>
  <td>BCA</td>
</tr>
  <td>2</td>
  <td>PQR</td>
  <td>BBA</td>
</tr>
</table>

```

The direction of text in individual cells can be changed by setting the dir attribute in an element that defines the cell.

- **cellpadding** attribute determines how much space will exist between a table cell border and the elements contained within it,
- **cellspacing** determines how much space will exist between each table cell. Color has been added to the table below to emphasize these attributes.
- **Border:** to specify the border width of a table.
- **Bordercolor:** to specify the border color of table
- **Bgcolor:** to change the background color of a table
- **Align:** to set the alignment of the table as center/left/right
- **Width:** to set the width of a table
- **Height:** to set the height of a table
- **Style:** to set color of the text.

#### Other Attributes of Table tag:

- **Border:** one can to specify the border by using border attribute.

Eg: <table border="2">

- **Bordercolor:** to set the bordercolor of the table

Eg: <table bordercolor="red">

- **Bgcolor:** to set the background color of the table
- **Style attribute:** to set the height, width and color of the text
- **Align:** to align the table

**Caption Tag** is used to provide the heading for a table

### Example:

```
<table dir="rtl" border="2" bordercolor="red"
bgcolor="green" align="right">
<caption>Tables</caption>
<tr><td>A</td><td>B</td></tr>
<tr><td>C</td><td>D</td></tr></table>
```

Output will be like this:

---

A	B
C	D

### The CAPTION element

#### *Attribute definitions*

- **top:** The caption is at the top of the table. This is the default value.
- **bottom:** The caption is at the bottom of the table.
- **left:** The caption is at the left of the table.
- **right:** The caption is at the right of the table.
- **Style:** to set color of the text.
- **align =** This attribute specifies the position of the caption with respect to the table.  
Possible values: top|bottom|left|right

**The TR element:** To enter the data row-wise.

#### **Attributes:**

- **Bgcolor:** to change the background color of a table
- **Style:** to set color of the text.

**The TD element:** To provide the data in a row

#### **Attributes:**

- **Bordercolor:** to specify the border color of cell
- **Bgcolor:** to change the background color of a cell
- **Align:** to set the alignment of the text as center/left/right

- **Width:** to set the width of a cell
- **Height:** to set the height of a cell
- **Style:** to set color of the text.
- **Colspan:** to span the cell multiple columns
- **Rowspan:** to span the cell multiple rows

### HTML Table Code:

```
<table border="1">
<tr>
  <td>Row 1 Cell 1</td>
  <td>Row 1 Cell 2</td>
</tr>
<tr>
  <td>Row 2 Cell 1</td>
  <td>Row 2 Cell 2</td>
</tr>
</table>
```

### Basic HTML Table Layout:


Row 1 Cell 1	Row 1 Cell 2
Row 2 Cell 1	Row 2 Cell 2

Content elements like HTML lists, images, and even other table elements can be placed inside each table cell. Doing so aligns the elements in a tabular fashion and provides structure.

### HTML Table Code:

#### HTML Table 2:

```
<table border="1">
<tr><td width="50%">
  <ul> <li>List Item 1</li><li>List Item 2</li> <li>List Item 3</li> </ul>
</td>
<td>
  <ul><li>List Item 4</li> <li>List Item 5</li> <li>List Item 6</li></ul>
</td></tr>
<tr>
<td><p>Avoid losing floppy disks with important school...</p></td>
<td> <a href="http://www.espn.com" target="_blank" rel="nofollow">
  
</a></td></tr></table>
```

<ul style="list-style-type: none"> <li>• List Item 1</li> <li>• List Item 2</li> <li>• List Item 3</li> </ul>	<ul style="list-style-type: none"> <li>• List Item 4</li> <li>• List Item 5</li> <li>• List Item 6</li> </ul>
Avoid losing floppy disks with important school...	

HTML tables allow the web designer to align page content in a tabular fashion while spanning elements horizontally across the web page, rather than stacking them up one on top of another.

### **HTML - Table rows & Table columns**

A table can contain an infinite number of table rows. Each table row is essentially a table element itself, with an opening and closing tag (<tr> </tr>). Table columns are also considered child elements of HTML tables, and like table rows, an HTML table may contain an infinite number of table data cells (<td> </td>).

Table rows and columns are container elements that house other HTML elements such as text links, images, and lists, as we've seen in previous examples. Below, we've applied a background color to the table example in order to help distinguish the different table elements.

### **HTML Table Code:**

```
<table border="1">
  <tr title="You are looking at Row 1" bgcolor="silver">
<td>Row 1 Cell 1</td>
    <td>Row 1 Cell 2</td>
  </tr>
  <tr title="You are looking at Row 2" bgcolor="aqua">
    <td>Row 2 Cell 1</td>
    <td>Row 2 Cell 2</td>
  </tr>
</table>
```

### **HTML Table Code Example:**

Row 1 Cell 1	Row 1 Cell 2
Row 2 Cell 1	Row 2 Cell 2

## **HTML Tables: spanning multiple rows and cells**

Use rowspan to span multiple rows merging together table rows and colspan to span across multiple columns.

### **HTML Table Rowspan Attribute:**

```
<table border="1">
  <tr>
    <td><b>Column 1</b></td>
    <td><b>Column 2</b></td>
    <td><b>Column 3</b></td>
  </tr>
  <tr>
    <td rowspan="2">Row 1 Cell 1</td>
    <td>Row 1 Cell 2</td>
    <td>Row 1 Cell 3</td>
  </tr>
  <tr>
    <td>Row 2 Cell 2</td>
    <td>Row 2 Cell 3</td>
  </tr>
  <tr>
    <td colspan="3">Row 3 Cell 1</td></tr></table>
```

### **HTML Colspan and Rowspan Attributes:**

Column 1	Column 2	Column 3
Row 1 Cell 1	Row 1 Cell 2	Row 1 Cell 3
	Row 2 Cell 2	Row 2 Cell 3
Row 3 Cell 1		

### **Cell padding and Cell spacing**

With the cellpadding and cellspacing attributes, you will be able to adjust the spacing between table cells. Setting the cellpadding attribute determines how much space will exist between a table cell border and the elements contained within it, whereas cellspacing determines how much space will exist between each table cell. Color has been added to the table below to emphasize these attributes.

### HTML Cellpadding/Cellspacing Code:

```
<table border="1" cellspacing="10"
bgcolor="blue">
  <tr>
    <td><b>Column 1</b></td>
    <td><b>Column 2</b></td>
  </tr>
  <tr>
    <td>Row 1 Cell 1</td>
    <td>Row 1 Cell 2</td>
  </tr>
  <tr>
    <td>Row 2 Cell 1</td>
    <td>Row 2 Cell 2</td></tr></table>
```

### HTML Cellspacing and Padding:

Column 1	Column 2
Row 1 Cell 1	Row 1 Cell 2
Row 2 Cell 1	Row 2 Cell 2

And now we will change the cellpadding of the table and remove the cellspacing from the previous example. This should clearly demonstrate the difference between cellpadding and cellspacing.

### HTML Code:

```
<table border="1" cellpadding="10"
bgcolor="blue">
  <tr>
    <td><b>Column 1</b></td>
    <td><b>Column 2</b></td>
  </tr>
  <tr>
    <td>Row 1 Cell 1</td>
    <td>Row 1 Cell 2</td>
  </tr>
  <tr>
    <td>Row 2 Cell 1</td>
    <td>Row 2 Cell 2</td></tr></table>
```

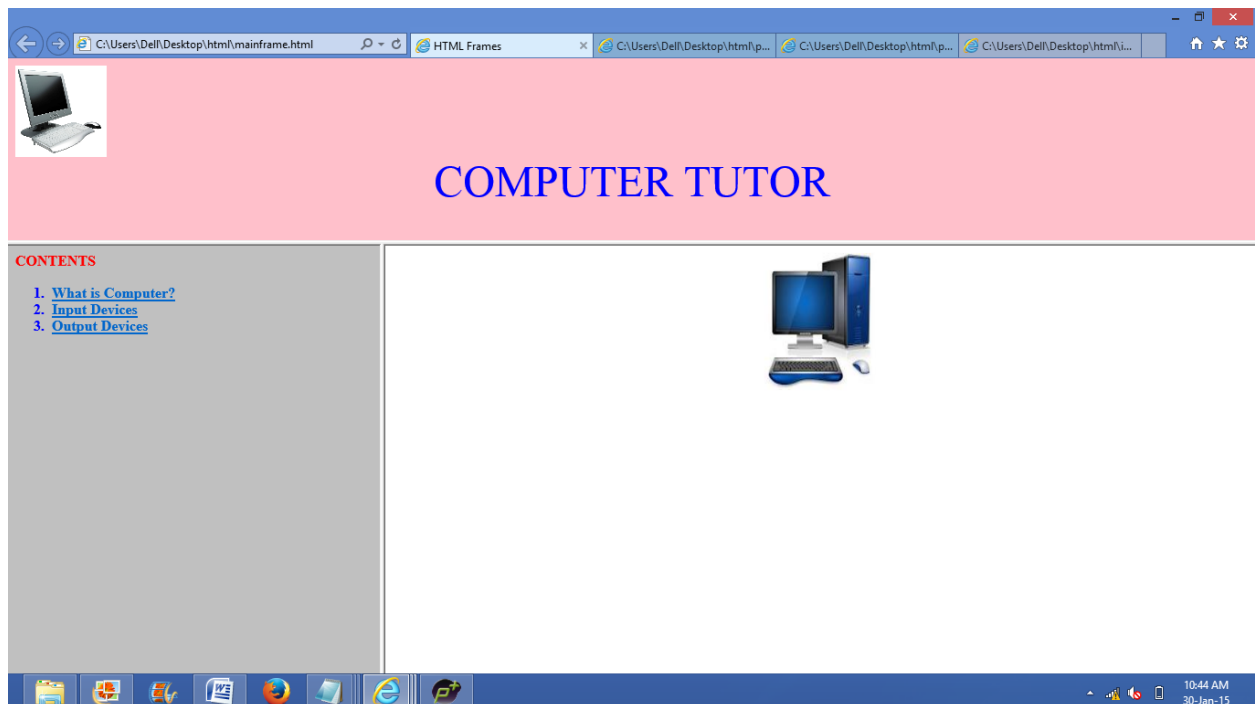
## HTML Cell Pads:

Column 1	Column 2
Row 1 Cell 1	Row 1 Cell 2
Row 2 Cell 1	Row 2 Cell 2

The value you specify for padding and spacing is interpreted by the browser as a pixel value. So a value of 10 is simply 10 pixels wide. Most HTML attributes that use numeric values for their measurements represent a pixel value.

## FRAMES

Frames help you to arrange and present more than one pages in the same window. Look at the figure which may look like an ordinary page, but it is actually three separate web pages are displayed in the same browser window.



Basically a frame is a rectangular region within the browser window that displays a Web page, alongside other pages in other frame. Frames divide the screen into multiple scrollable regions enabling you to present information in a more flexible and useful way.

## Disadvantages of Frames

There are few drawbacks with using frames, so it's never recommended to use frames in your webpage:

- Some smaller devices cannot cope with frames often because their screen is not big enough to be divided up.
- Sometimes your page will be displayed differently on different computers due to different screen resolution.
- The browser's *back button* might not work as the user hopes.
- There are still few browsers that do not support frame technology.

## Creating Frames

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

## Example

```
<html>
<head>
<title>HTML Frames</title>
</head>
<frameset rows="30%,70%">
  <frame name="frame1"
src="C:\Users\Dell\Desktop\html\main.html">
<frameset cols="30%,*">
  <frame name="frame2"
src="C:\Users\Dell\Desktop\html\choice.html">
  <frame name="frame3"
src="C:\Users\Dell\Desktop\html\mainarea.html">
</frameset>
</frameset>
</html>
```



## COMPUTER TUTOR

### CONTENTS

1. [What is Computer?](#)
2. [Input Devices](#)
3. [Output Devices](#)





## The <frameset> Tag Attributes

Following are important attributes of the <frameset> tag:

Attribute	Description
Cols	<p>specifies how many columns are contained in the frameset and the size of each column. You can specify the width of each column in one of four ways:</p> <ul style="list-style-type: none"><li>• Absolute values in pixels. For example to create three vertical frames, use <i>cols="100, 500, 100"</i>.</li><li>• A percentage of the browser window. For example to create three vertical frames, use <i>cols="10%, 80%, 10%"</i>.</li><li>• Using a wildcard symbol. For example to create three vertical frames, use <i>cols="10%, *, 10%"</i>. In this case wildcard takes remainder of the window.</li><li>• As relative widths of the browser window. For example to create three vertical frames, use <i>cols="3*, 2*, 1*"</i>. This is an alternative to percentages. You can use relative widths of the browser window. Here the window is divided into sixths: the first column takes up half of the window, the second takes one third, and the third takes one sixth.</li></ul>
rows	<p>This attribute works just like the cols attribute and takes the same values, but it is used to specify the rows in the frameset. For example to create two horizontal frames, use <i>rows="10%, 90%"</i>. You can specify the height of each row in the same way as explained above for columns.</p>
border	<p>This attribute specifies the width of the border of each frame in pixels. For example <i>border="5"</i>. A value of zero means no border.</p>
frameborder	<p>This attribute specifies whether a three-dimensional border should be displayed between frames. This attribute takes value either 1 (yes) or 0 (no). For example <i>frameborder="0"</i> specifies no border.</p>
framespacing	<p>This attribute specifies the amount of space between frames in a frameset. This can take any integer value. For example <i>framespacing="10"</i> means there should be 10 pixels spacing between each frames.</p>

## The <frame> Tag Attributes

Following are important attributes of <frame> tag:

Attribute	Description
-----------	-------------

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

### Browser Support for Frames

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

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

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

### Frame's name and target attributes

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

Let's see following example where a mainframe.htm file has following code:

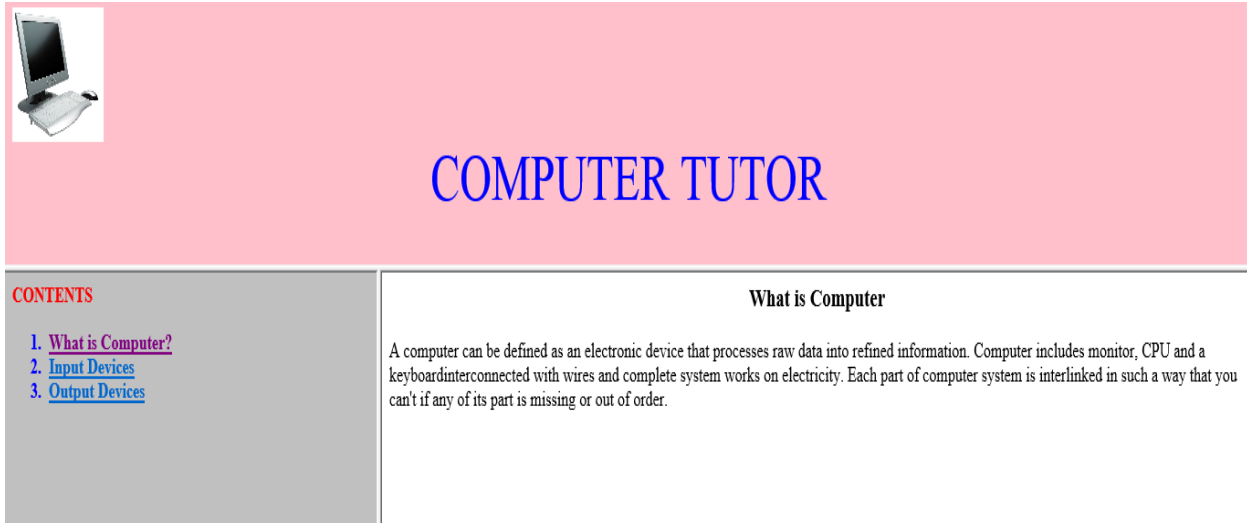
```
<html>
<head>
<title>HTML Frames</title>
</head>
<frameset rows="30%,70%">
  <frame name="frame1"
src="C:\Users\Dell\Desktop\html\main.html">
<frameset cols="30%,*">
  <frame name="frame2"
src="C:\Users\Dell\Desktop\html\choice.html">
  <frame name="frame3"
src="C:\Users\Dell\Desktop\html\mainarea.html">
</frameset>
</frameset>
</html>
```

```
<html>
<head><title>Computer Tutorial</title></head>
<body bgcolor="pink">

<center><font color="blue" size="8">COMPUTER TUTOR</font></center>
</body>
</html>
```

```
<html>
<head><title>Link Window</title></head>
<body bgcolor="silver">
<p><font color="red"><b>CONTENTS</b></font></p>
<ol><font color="blue">
<li><a href="C:\Users\Dell\Desktop\html\page1.html" target="frame3">What is
Computer?</a></li>
<li><a href="C:\Users\Dell\Desktop\html\page2.html" target="frame3">Input
Devices</a></li>
<li><a href="C:\Users\Dell\Desktop\html\page3.html" target="frame3">Output
Devices</a></li>
</font></ol>
</body>
</html>
```

Hence the output is:



When we click on any of the hyperlinks the explanation is shown in **frame3**. The target attribute of the <a> tag consists of “**frame3**”.

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

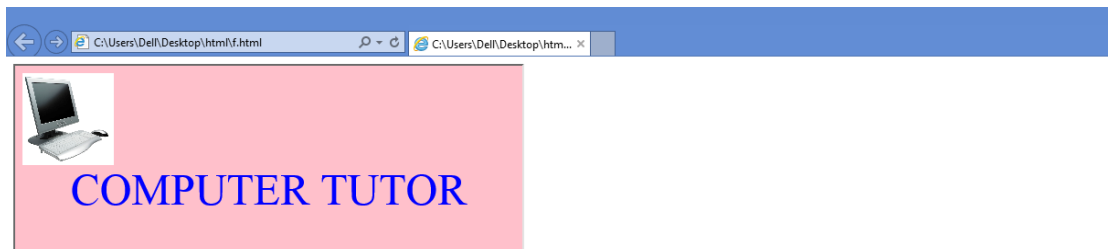
Option	Description
_self	Loads the page into the current frame.
_blank	Loads a page into a new browser window, opening a new window.
_parent	Loads the page into the parent window, which in the case of a single frameset is the main browser window.
_top	Loads the page into the browser window, replacing any current frames.
targetframe	Loads the page into a named targetframe.

### Inline frames (iframes or floating frames)

**HTML IFRAME Element:** You can define an inline frame with HTML tag <iframe>. The <iframe> tag is not somehow related to <frameset> tag, instead, it can appear anywhere in your document. The <iframe> tag defines a rectangular region within the document in which the browser can display a separate document, including scrollbars and borders.

The **src** attribute is used to specify the URL of the document that occupies the inline frame.

```
<html>
<body>
<iframe name="example" src="C:\Users\Dell\Desktop\html\main.html" width="555" height="200">
</body>
</html>
```



### The <iframe> Tag Attributes

Most of the attributes of the <iframe> tag, including *name*, *class*, *frameborder*, *id*, *longdesc*, *marginheight*, *marginwidth*, *name*, *scrolling*, *style*, and *title* behave exactly like the corresponding attributes for the <frame> tag.

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

## HTML <map> Tag

### Example

An image-map, with clickable areas:

```

<map name="planetmap">
  <area shape="rect" coords="0,0,82,126" href="sun.htm" alt="Sun">
  <area shape="circle" coords="90,58,3" href="mercur.htm" alt="Mercury">
  <area shape="circle" coords="124,58,8" href="venus.htm" alt="Venus">
</map>
```

The <map> tag is used to define a client-side image-map. An image-map is an image with clickable areas.

The required name attribute of the <map> element 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> elements that defines the clickable areas in the image map.

## HTML - Web Forms

An HTML form is part of a Web page that includes area where readers can enter information to e send back to the publisher of the web page.

Generally, a web form allows the web page readers to send back some information to the publisher of the web page.

HTML web forms are a composition of buttons, checkboxes, and text input fields embedded inside of HTML documents with one goal in mind: to capture user input. By doing things such as providing fields for user data such as names, phone number, and email addresses, web forms give users the opportunity to interact directly with a webpage.

HTML forms are placed on a web page using the **<form> tag**. This tag should encapsulate a series of other form elements, identifying them as a single cohesive web form.

Syntax is:

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

The most important form element is the **input element**.

The input element is used to select user information.

An input element can vary in many ways, depending on the type attribute. An input element can be of type text field, checkbox, password, radio button, submit button, and more.

The most used input types are described below.

## **Text Fields**

`<input type="text">` defines a one-line input field that a user can enter text into:

```
<form>
First name: <input type="text" name="firstname"><br />
Last name: <input type="text" name="lastname">
</form>
```

How the HTML code above looks in a browser:

First name:

Last name:

**Note:** The form itself is not visible. Also note that the default width of a text field is 20 characters.

## **Password Field**

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

```
<form>
Password: <input type="password" name="pwd" >
</form>
```

How the HTML code above looks in a browser:

Password:

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

## **Radio Buttons**

`<input type="radio">` defines a radio button. Radio buttons let a user select ONLY ONE of a limited number of choices:

```
<form>
<input type="radio" name="gender" value="male"> Male<br />
<input type="radio" name="gender" value="female"> Female
</form>
```

How the HTML code above looks in a browser:

- ☒ Male  
☐ Female

## **Checkboxes**

`<input type="checkbox" />` defines a checkbox. Checkboxes let a user select ZERO or MORE options of a limited number of choices.

```
<form>
<input type="checkbox" name="vehicle" value="Bike" /> I have a bike<br />
<input type="checkbox" name="vehicle" value="Car" /> I have a car
</form>
```

How the HTML code above looks in a browser:

- ☐ I have a bike  
☐ I have a car

## **Submit Button**

`<input type="submit">` defines a submit button.

A submit button is used to send form data to a server. The data is sent to the page specified in the form's action attribute. The file defined in the action attribute usually does something with the received input:

```
<form name="input" action="html_form_action.asp" method="get">
Username: <input type="text" name="user">
<input type="submit" value="Submit">
</form>
```

How the HTML code above looks in a browser:



---

Username:

HTML form elements rely on *action* and *method* attributes to identify where to send the form data for processing (action) and how to process the data (method). In the code above, we've inserted some make-believe values to represent what a typical HTML form might look like behind the scenes.

Unfortunately, HTML alone is unable to process form data. A scripting language such as PHP, PERL, and/or JavaScript must be used with HTML forms to process data captured by HTML form elements.

### **HTML Email Form Element:**

```
<form name="myWebForm" action=mailto:youremail@email.com method="post">  
  <input type="checkbox" /> Checkbox 1<br />  
  <input type="text" /> Text Field 1<br />  
  <input type="submit" value="SUBMIT" />  
</form>
```

How the HTML code above looks in a browser:

---

☐ Checkbox 1  
 Text Field 1

### **HTML - Select Lists**

HTML *select* fields provide essentially the same functionality as HTML Checkbox Fields. They allow the user to select one or more values from a pre-determined series of options.

#### **<select> Tag**

This tag can be used to create both scrolling lists and pull-down pick lists.

Incorporating a select field into a web page is done using the <select> tag. List values are then added to the field using the <option> tag, similar to how list items <li> are added to ordered list elements (<ol>).

## HTML Drop Down List:

By default, select fields, popularly called drop down lists, only allow the user to choose a single value. This behavior and appearance may be changed by adjusting the *multiple* and *size* attributes as demonstrated below.

### HTML Selection Field Code:

```
<select size="3" name="selectionField" multiple="yes" >
  <option value="CA" selected>California -- CA </option>
  <option value="CO" >Colorado -- CO</option>
  <option value="CN" >Connecticut -- CN</option>
</select>
```



If we do not use the size and multiple attribute, then the list will look like as:

```
<select name="selectionField" >
  <option selected>California </option>
  <option>Colorado</option>
  <option>Connecticut</option>
</select>
```

How the HTML code above looks in a browser:



### HTML Selection Element:

With the above settings, the user is now able to select multiple values by pressing and holding the **Control** (ctrl) key and clicking each value.

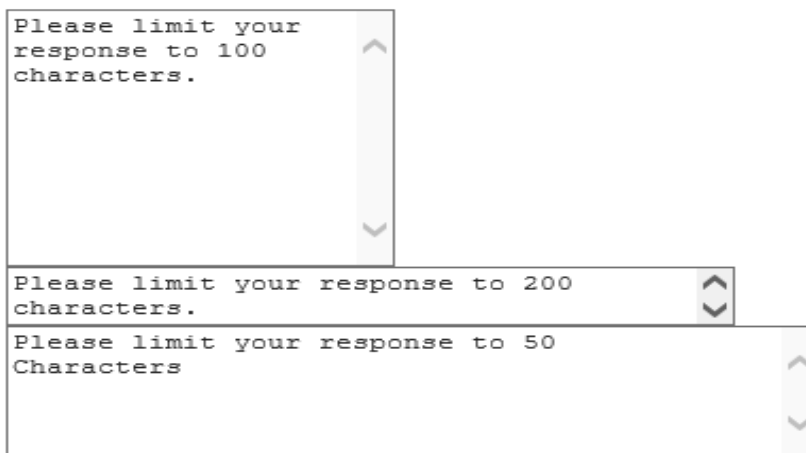
## HTML TEXTAREA

An HTML textarea is an oversized Text Field capable of capturing "blurb" type information from a user. If you've ever posted on a forum or left feedback on your favorite blog, you probably do so using an HTML textarea.

### HTML TextArea Code:

```
<textarea name="myTextArea" cols="20" rows="10">Please limit your response to 100
characters.</textarea><br />
<textarea name="myTextArea" cols="40" rows="2">Please limit your response to 200
characters.</textarea><br />
<textarea name="myTextArea" cols="45" rows="5">Please limit your response to 50
Characters</textarea><br>
```

How the HTML code above looks in a browser:



## ADDING SOUNDS IN HTML

There are two methods of adding sound in the HTML document.

### Using <embed> tag:

Sometimes you need to add music or video into your web page. The easiest way to add video or sound to your web site is to include the special HTML tag called **<embed>**. This tag causes the browser itself to include controls for the multimedia automatically provided browser supports <embed> tag and given media type.

```
<html>
<body>
<embed src="Happy Birthday.mp3" autostart="true"
loop="true" width="2" height="0">
</embed>
</body>
</html>
```

The width and height attribute given above causes the player to be **invisible**. If you do not want it to be invisible, you can specify your own dimensions.

### **The <embed> Tag Attributes**

Following is the list of important attributes which can be used with <embed> tag.

Attribute	Description
align	Determines how to align the object. It can be set to either <i>center</i> , <i>left</i> or <i>right</i> .
autostart	This Boolean attribute indicates if the media should start automatically. You can set it either true or false.
loop	Specifies if the sound should be played continuously (set loop to true), a certain number of times (a positive value) or not at all (false)
playcount	Specifies the number of times to play the sound. This is alternate option for <i>loop</i> if you are using IE.
hidden	Specifies if the multimedia object should be shown on the page. A false value means no and true values means yes.
width	Width of the object in pixels
height	Height of the object in pixels
name	A name used to reference the object.
src	URL of the object to be embedded.
volume	Controls volume of the sound. Can be from 0 (off) to 100 (full volume).

### **Using HTML Object tag**

The **<object>** element allows HTML authors to specify everything required by an object for its presentation by a user agent

You can embed an HTML document in an HTML document itself as follows:

```

<html>
<body>
<object data="Happy Birthday.mp3" width="300" height="200">
  alt : <a href="C:\Users\Dell\Desktop\html\page1.html">test.htm</a>
</object>
</body>
</html>

```

Here *alt* attribute will come into picture if browser does not support *object* tag.

We can create a hyperlink to a sound clip by:

```

<html>
<body>
<a href="Happy Birthday.mp3">Play</a>
</body>
</html>

```

## **HTML <blockquote> Tag**

The HTML <blockquote> tag is used for indicating long quotations (i.e. quotations that span multiple lines) or a section that is quoted from another resource. It should contain only block-level elements within it, and not just plain text.

**The HTML <blockquote> tag also supports following additional attributes:**

Attribute	Value	Description
cite	URL	URL of the quote, if it is taken from the web.

```

<html>
<body>
<h1>About WWF</h1>
<p>Here is a quote from WWF's website :</p>
<blockquote cite="http://www.worldwildlife.org/who/index.html">
For 50 years, WWF has been protecting the future of nature. The world's
leading conservation organization, WWF works in 100 countries and is
supported by 1.2 million members in the United States and close to 5
million globally.
</blockquote>
</body>
</html>

```

It will look like following in the browser window:

## About WWF

Here is a quote from WWF's website :

For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.

## HTML <q> Tag

The HTML <q> tag is used for indicating short quotations (i.e. quotations that span multiple lines).

Example:

```
<html>
<head>
<title>HTML q Tag</title>
</head>
<body>
Amit said <q> I am learning HTML </q>
</body>
</html>
```

### Output:

Amit said “ I am learning HTML ”

### Attributes

The HTML <q> tag also supports following additional attributes:

Attribute	Value	Description
cite	URL	URL of the quote, if it is taken from the web.

## **HTML <basefont> Tag**

The HTML <basefont> tag is used to specify a base font for the document to use. This base font is applied to complete document. This tag is deprecated now. It is an unpaired tag and is included in <head> tag. It is supported only by IE.

```
<html>
<head>
<basefont face="arial" color="blue" size="4"/>
<title>HTML basefont Tag</title>
</head>
<body>
<p>The HTML basefont tag is now deprecated. You should use CSS font to set font properties
instead.</p>
</body>
</html>
```

### **Attributes:**

The HTML <basefont> tag also supports following additional attributes:

Attribute	Value	Description
color	colorname	<i>Deprecated</i> - Specifies the color of the text.
face	font names separated by comma	<i>Deprecated</i> - Specifies the font family of the text.
Size	1 to 7	<i>Deprecated</i> - Specifies the font size of the text.

## **Emphasizing text implicitly and explicitly**

**Implicit tags** are those that allow the browser to choose, within limitations, how the marked-up text will be displayed. Header tags are actually an example of an implicit tag, since the HTML designer has no control over how much bigger or smaller a header tag will be. Although most browsers will render header tags in somewhat similar ways, others have to come up with another system for emphasis, such as underlining or highlighting the text.

Because HTML was originally created with the overriding mission of being displayed on nearly any computer system, implicit tags for emphasis were a necessity. HTML allows the designer to decide what text will be emphasized. But only explicit tags tell the Web browser how to render that text.

### Explicit Styles

Explicit tags are also often called *physical tags*, since they very specifically tell the Web browser how you want the text to physically appear. The browser is given no choice in the matter.

The basic explicit tags are containers that let the user mark text as **bold, italic, or underlined**

### HTML Physical Container Tags

S. No.	Tags	Meaning
1.	<B>Text</B>	Bold text
2.	<I>Text</I>	Italic text
3.	<U>Text</U>	Underlined text

An important feature of explicit (physical) tags is that they can generally be used in combination with other tags. As you'll see in the next section, this isn't always a good idea with implicit tags. For instance, most graphic browsers will render the following example by applying both tags to the text

```
<html>
<body>
<H1><I>HTML</I></H1>
<B><I>This is bold and italic</I></B>
</body>
</html>
```

### Output:

---

# *HTML*

*This is bold and italic*

### Implicit HTML Tags

Implicit styles are often called *logical styles*, since they allow the browser some freedom in how it will display the text. These tags, like the header tags, are generally relative to one another, depending on the browser being used to view them. Following are some of the common implicit (logical) tags



### Some Basic Logical HTML Tags

Tags	Meaning	Generally Rendered as
<EM>Text</EM>	Emphasis	Italic text
<STRONG>Text</STRONG>	Strong emphasis	Bold text
<TT>Text</TT>	Teletype	Monospaced text

The basic difference between logical and physical tags is as:

First, these logical (implicit) tags will always be rendered by any Web browser that views them. Even text browsers (which are unable to show italic text) will display the <EM> or <STRONG> tags by underlining, boldfacing, or highlighting the text.

Second, **these tags are generally not effective when used together**. Where <B><I>text</I></B> will sometimes offer useful results, <EM><STRONG>text</STRONG></EM> rarely will. Combining these tags with other tags (such as header tags or physical tags) is often either ineffective or redundant.

### HTML <tt> Tag

The HTML <tt> tag specifies teletype text. It is used to display monospaced text.

```
<html>
<head>
<title>HTML tt Tag</title>
</head>
<body>
<p>Implicit tags in HTML</p>
<tt>It will display monospaced text</tt>
</body>
</html>
```

**Output:**

---

Implicit tags in HTML

It will display monospaced text

### HTML <cite> Tag

The HTML <cite> tag specifies a citation. It can be defined as title of a work.

Example:

```

<html>
<head>
<title>HTML cite Tag</title>
</head>
<body>
<p>The learning content can be referred from <cite>Data Structures & Algorithms in
Java</cite><p>
</body>
</html>

```

### **Output:**

---

The learning content can be referred from *Data Structures & Algorithms in Java*

## **OTHER HTML TAGS**

Tag	Meaning	Generally Rendered as
<CODE>, </CODE>	Programming lines	Monospaced (like <TT>)
<KBD>, </KBD>	Keyboard text	Monospaced
<SAMP>, </SAMP>	Sample output	Monospaced
<VAR>, </VAR>	Variable	Italic
<DFN>, </DFN>	Term definition	Regular text
<ADDRESS>, </ADDRESS>	Street or e-mail address	Italic text

### **Difference between Relative Links and Absolute Links**

S. No.	Absolute Links	Relative Links
1.	The absolute link is of the form protocol://domain/path	Relative link is of the form <a href="path">text</a>
2.	The absolute path always includes the domain name of the website.	The relative path only includes a file or a file path.
3.	Absolute links are used to link to a location on another website.	Relative links are used to link to pages or files within a website.
4.	Absolute links are specified in the title bar of search engines.	Relative links are included in the web page source code.
5.	Eg. http://mywebsite.com/home.html	Eg. <a href="home.html">click here</a>

## **HTML GOALS**

1. To create basic elements of web pages.
2. To design and develop complete websites.
3. To allow text, images, videos and sound clips to be embedded inside web pages.

4. To create structured documents by denoting the structural format for text such as headings, paragraphs, lists quotes and other items.
5. To embed scripts like javascript which affect the behavior of web pages.
6. To create interactive forms using HTML form elements.
7. To embed cascading style sheets (CSS) to define the appearance and layout of HTML content.

### **Advantages of HTML**

1. Easy to use and learn.
2. Simple and flexible syntax.
3. Supported by all browsers.
4. Widely used.
5. License free.

### **Disadvantages of HTML**

1. Cannot produce dynamic output.
2. Limited security.
3. Complex documents are difficult to maintain.
4. Dependent on scripting language for advance features.