HTML stands for **H**yper **T**ext **M**arkup **L**anguage, which is the most used language for developing the web pages.

HTML was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Now we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012.

Hello World Program:

<!DOCTYPE html>

<html>

<body>

<h1>Hello World!</h1>

</body>

</html>

**HTML Tags**

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

Above example of HTML document uses following tags:

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

World Wide Web Consortium (W3C) recommends to use lowercase tags starting from HTML 4.

**HTML Document Structure**

A typical HTML document will have following structure:

Document declaration tag

<html>

<head>

header related tags

</head>

<body>

body related tags

</body>

</html>

**The <!DOCTYPE> Declaration**

The <!DOCTYPE> declaration tag is used by the web browser to understand the version of the HTML used in the document. Current version of HTML is 5 and it makes use of the following declaration:

<!DOCTYPE html>

## Heading Tags

Any Description starts with a heading. You can use different sizes for your headings. HTML also has six levels of headings, which use the elements **<h1>, <h2>, <h3>, <h4>, <h5>, and <h6>**. <h1> tag is large size and <h6> is small size.

<!DOCTYPE html>

<html>

<head>

<title>Heading Example</title>

</head>

<body>

<h1>This is heading 1</h1>

<h2>This is heading 2</h2>

<h3>This is heading 3</h3>

<h4>This is heading 4</h4>

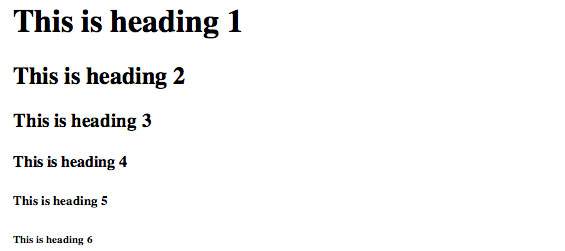
<h5>This is heading 5</h5>

<h6>This is heading 6</h6>

</body>

</html>

This will produce following result:



## Paragraph Tag

The **<p>** tag offers a way to structure your text into different paragraphs. Each paragraph of text should go in between an opening <p> and a closing </p> tags.

<!DOCTYPE html>

<html>

<head>

<title>Paragraph tag Example</title>

</head>

<body>

<p>This is a first paragraph of text.</p>

<p>This is a second paragraph of text.</p>

</body>

</html>

This will produce following result:

Here is a first paragraph of text.

Here is a second paragraph of text.

## Line Break Tag

Whenever you use the **<br />** element, anything following it starts from the next line.

<!DOCTYPE html>

<html>

<head>

<title>Line Break tag Example</title>

</head>

<body>

<p>Hello<br />

You delivered your assignment ontime Thanks<br />

Mahnaz</p>

</body>

</html>

This will produce following result:

Hello  
You delivered your assignment ontime Thank  
Mahnaz

## Centering Content

You can use **<center>** tag to put any content in the center of the page or any table cell.

<!DOCTYPE html>

<html>

<head>

<title>Centring Content Example</title>

</head>

<body>

<p>This text is not in the center.</p>

<center>

<p>This text is in the center.</p>

</center>

</body>

</html>

This will produce following result:

This text is not in the center.

This text is in the center.

## Horizontal Lines

Horizontal lines are used to visually break up sections of a document. The **<hr>** tag creates a line from the current position in the document to the right margin and breaks the line accordingly.

<!DOCTYPE html>

<html>

<head>

<title>Horizontal Line Example</title>

</head>

<body>

<p>This is paragraph one and should be on top</p>

<hr />

<p>This is paragraph two and should be at bottom</p>

</body>

</html>

This will produce following result:

This is paragraph one and should be on top

## Preserve Formatting

Sometimes you want your text to follow the exact format of how it is written in the HTML document. In those cases, you can use the preformatted tag <pre>.

Any text between the opening <pre> tag and the closing </pre> tag will preserve the formatting of the source document.

<!DOCTYPE html>

<html>

<head>

<title>Preserve Formatting Example</title>

</head>

<body>

<pre>

function testFunction( strText ){

alert (strText)

}

</pre>

</body>

</html>

This will produce following result:

function testFunction( strText ){

alert (strText)

}

## No breaking Spaces

Suppose you want to use the phrase "10 Jan 2000." Here you would not want a browser to split the "10,Jan" and "2000" across two lines:

An example of this technique appears in the movie "10 Jan 2000."

In cases where you do not want the client browser to break text, you should use a nonbreaking space entity **&nbsp;** instead of a normal space.

<!DOCTYPE html>

<html>

<head>

<title>Nonbreaking Spaces Example</title>

</head>

<body>

<p>An example of this technique appears in the movie "12&nbsp;Angry&nbsp;Men."</p>

</body>

</html>

# HTML Elements

An **HTML element** is defined by a starting tag. If the element contains other content, it ends with a closing tag, where the element name is preceded by a forward slash as shown below with few tags:

|  |  |  |
| --- | --- | --- |
| **Start Tag** | **Content** | **End Tag** |
| <p> | This is paragraph content. | </p> |
| <h1> | This is heading content. | </h1> |
| <div> | This is division content. | </div> |
| <br /> |  |  |

So here <p>....</p> is an HTML element, <h1>...</h1> is another HTML element. There are some HTML elements which don't need to be closed, such as <img.../>, <hr /> and <br /> elements. These are known as **void elements**.

## Nested HTML Elements

It is very much allowed to keep one HTML element inside another HTML element:

<!DOCTYPE html>

<html>

<head>

<title>Nested Elements Example</title>

</head>

<body>

<h1>This is <i>italic</i> heading</h1>

<p>This is <u>underlined</u> paragraph</p>

</body>

</html>

This will display following result:

# This is *italic* heading

This is underlined paragraph

# HTML Attributes

An attribute is used to define the characteristics of an HTML element and is placed inside the element's opening tag. All attributes are made up of two parts: a **name** and a **value**:

* The **name** is the property you want to set. For example, the paragraph <p> element in the example carries an attribute whose name is **align**, which you can use to indicate the alignment of paragraph on the page.
* The **value** is what you want the value of the property to be set and always put within quotations. The below example shows three possible values of align attribute: **left, center** and **right**.

Attribute names and attribute values are case-insensitive.

<!DOCTYPE html>

<html>

<head>

<title>Align Attribute Example</title>

</head>

<body>

<p align="left">This is left aligned</p>

<p align="center">This is center aligned</p>

<p align="right">This is right aligned</p>

</body>

</html>

This will display following result:

This is left aligned

This is center aligned

This is right aligned

## Core Attributes

The four core attributes that can be used on the majority of HTML elements (although not all) are:

* id
* title
* class
* style

### The id Attribute

The **id** attribute of an HTML tag can be used to uniquely identify any element within an HTML page. There are two primary reasons that you might want to use an id attribute on an element:

* If an element carries an id attribute as a unique identifier it is possible to identify just that element and its content.
* If you have two elements of the same name within a Web page (or style sheet), you can use the id attribute to distinguish between elements that have the same name.

<p id="html">This para explains what is HTML</p>

<p id="css">This para explains what is Cascading Style Sheet</p>

### The title Attribute

The **title** attribute gives a suggested title for the element. They syntax for the **title** attribute is similar as explained for **id** attribute:

The behavior of this attribute will depend upon the element that carries it, although it is often displayed as a tooltip when cursor comes over the element.

<!DOCTYPE html>

<html>

<head>

<title>The title Attribute Example</title>

</head>

<body>

<h3 title="Hello HTML!">Titled Heading Tag Example</h3>

</body>

</html>

This will produce following result:

### Titled Heading Tag Example

Now try to bring your cursor over "Titled Heading Tag Example" and you will see that whatever title you used in your code is coming out as a tooltip of the cursor.

### The class Attribute

The **class** attribute is used to associate an element with a style sheet, and specifies the class of element. You will learn more about the use of the class attribute when you will learn Cascading Style Sheet (CSS). So for now you can avoid it.

The value of the attribute may also be a space-separated list of class names. For example:

class="className1 className2 className3"

### The style Attribute

The style attribute allows you to specify Cascading Style Sheet (CSS) rules within the element.

<!DOCTYPE html>

<html>

<head>

<title>The style Attribute</title>

</head>

<body>

<p style="font-family:arial; color:#FF0000;">Some text...</p>

</body>

</html>

This will produce following result:

Some text...

At this point of time, we are not learning CSS, so just let's proceed without bothering much about CSS. Here you need to understand what are HTML attributes and how they can be used while formatting content.

## Internationalization Attributes

There are three internationalization attributes, which are available for most (although not all) XHTML elements.

* dir
* lang

### The dir Attribute

The **dir** attribute allows you to indicate to the browser the direction in which the text should flow. The dir attribute can take one of two values, as you can see in the table that follows:

|  |  |
| --- | --- |
| **Value** | **Meaning** |
| ltr | Left to right (the default value) |
| rtl | Right to left |

<!DOCTYPE html>

<html dir="rtl">

<head>

<title>Display Directions</title>

</head>

<body>

This is how renders right-to-left directed text.

</body>

</html>

This will produce following result:

This is how renders right-to-left directed text.

When *dir* attribute is used within the <html> tag, it determines how text will be presented within the entire document. When used within another tag, it controls the text's direction for just the content of that tag.

### The lang Attribute

The **lang** attribute allows you to indicate the main language used in a document, but this attribute was kept in HTML only for backwards compatibility with earlier versions of HTML.

The values of the *lang* attribute are ISO-639 standard two-character language codes. Check [**HTML Language Codes: ISO 639**](https://www.tutorialspoint.com/html/language_iso_codes.htm) for a complete list of language codes.

<!DOCTYPE html>

<html lang="en">

<head>

<title>English Language Page</title>

</head>

<body>

This page is using English Language

</body>

</html>

## Generic Attributes

Here's a table of some other attributes that are readily usable with many of the HTML tags.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Options** | **Function** |
| align | right, left, center | Horizontally aligns tags |
| valign | top, middle, bottom | Vertically aligns tags within an HTML element. |
| bgcolor | numeric, hexidecimal, RGB values | Places a background color behind an element |
| background | URL | Places a background image behind an element |
| id | User Defined | Names an element for use with Cascading Style Sheets. |
| class | User Defined | Classifies an element for use with Cascading Style Sheets. |
| width | Numeric Value | Specifies the width of tables, images, or table cells. |
| height | Numeric Value | Specifies the height of tables, images, or table cells. |
| title | User Defined | "Pop-up" title of the elements. |

# HTML Formatting

If you use a word processor, you must be familiar with the ability to make text bold, italicized, or underlined; these are just three of the ten options available to indicate how text can appear in HTML and XHTML.

## Bold Text

Anything that appears within **<b>...</b>** element, is displayed in bold as shown below:

<!DOCTYPE html>

<html>

<head>

<title>Bold Text Example</title>

</head>

<body>

<p>The following word uses a <b>bold</b> typeface.</p>

</body>

</html>

This will produce following result:

The following word uses a **bold** typeface.

## Italic Text

Anything that appears within **<i>...</i>** element is displayed in italicized as shown below:

<!DOCTYPE html>

<html>

<head>

<title>Italic Text Example</title>

</head>

<body>

<p>The following word uses a <i>italicized</i> typeface.</p>

</body>

</html>

This will produce following result:

The following word uses a *italicized* typeface.

## Underlined Text

Anything that appears within **<u>...</u>** element, is displayed with underline as shown below:

<!DOCTYPE html>

<html>

<head>

<title>Underlined Text Example</title>

</head>

<body>

<p>The following word uses a <u>underlined</u> typeface.</p>

</body>

</html>

This will produce following result:

The following word uses a underlined typeface.

## Strike Text

Anything that appears within **<strike>...</strike>** element is displayed with strikethrough, which is a thin line through the text as shown below:

<!DOCTYPE html>

<html>

<head>

<title>Strike Text Example</title>

</head>

<body>

<p>The following word uses a <strike>strikethrough</strike> typeface.</p>

</body>

</html>

This will produce following result:

The following word uses a ~~strikethrough~~ typeface.

## Superscript Text

The content of a **<sup>...</sup>** element is written in superscript; the font size used is the same size as the characters surrounding it but is displayed half a character's height above the other characters.

<!DOCTYPE html>

<html>

<head>

<title>Superscript Text Example</title>

</head>

<body>

<p>The following word uses a <sup>superscript</sup> typeface.</p>

</body>

</html>

This will produce following result:

The following word uses a superscript typeface.

## Subscript Text

The content of a **<sub>...</sub>** element is written in subscript; the font size used is the same as the characters surrounding it, but is displayed half a character's height beneath the other characters.

<!DOCTYPE html>

<html>

<head>

<title>Subscript Text Example</title>

</head>

<body>

<p>The following word uses a <sub>subscript</sub> typeface.</p>

</body>

</html>

This will produce following result:

The following word uses a subscript typeface.

## *Inserted Text*

*Anything that appears within****<ins>...</ins>****element is displayed as inserted text.*

*<!DOCTYPE html>*

*<html>*

*<head>*

*<title>Inserted Text Example</title>*

*</head>*

*<body>*

*<p>I want to drink <del>cola</del> <ins>wine</ins></p>*

*</body>*

*</html>*

*This will produce following result:*

*I want to drink  wine*

## Deleted Text

Anything that appears within **<del>...</del>** element, is displayed as deleted text.

<!DOCTYPE html>

<html>

<head>

<title>Deleted Text Example</title>

</head>

<body>

<p>I want to drink <del>cola</del> <ins>wine</ins></p>

</body>

</html>

This will produce following result:

I want to drink  wine

## Larger Text

The content of the **<big>...</big>** element is displayed one font size larger than the rest of the text surrounding it as shown below:

<!DOCTYPE html>

<html>

<head>

<title>Larger Text Example</title>

</head>

<body>

<p>The following word uses a <big>big</big> typeface.</p>

</body>

</html>

This will produce following result:

The following word uses a big typeface.

## Smaller Text

The content of the **<small>...</small>** element is displayed one font size smaller than the rest of the text surrounding it as shown below:

<!DOCTYPE html>

<html>

<head>

<title>Smaller Text Example</title>

</head>

<body>

<p>The following word uses a <small>small</small> typeface.</p>

</body>

</html>

This will produce following result:

The following word uses a small typeface.

## Grouping Content

The **<div>** and **<span>** elements allow you to group together several elements to create sections or subsections of a page.

<!DOCTYPE html>

<html>

<head>

<title>Div Tag Example</title>

</head>

<body>

<div id="menu" align="middle" >

<a href="/index.htm">HOME</a> |

<a href="/about/contact\_us.htm">CONTACT</a> |

<a href="/about/index.htm">ABOUT</a>

</div>

<div id="content" align="left" bgcolor="white">

<h5>Content Articles</h5>

<p>Actual content goes here.....</p>

</div>

</body>

</html>

This will produce following result:

[HOME](https://www.tutorialspoint.com/index.htm) | [CONTACT](https://www.tutorialspoint.com/about/contact_us.htm) | [ABOUT](https://www.tutorialspoint.com/about/index.htm)

##### CONTENT ARTICLES

Actual content goes here.....

The <span> element, on the other hand, can be used to group inline elements only. So, if you have a part of a sentence or paragraph which you want to group together, you could use the <span> element as follows

<!DOCTYPE html>

<html>

<head>

<title>Span Tag Example</title>

</head>

<body>

<p>This is the example of <span style="color:green">span tag</span> and the <span style="color:red">div tag</span> along with CSS</p>

</body>

</html>

This will produce following result:

This is the example of span tag and the div tag alongwith CSS

These tags are commonly used with CSS.

# HTML Phrase Tags

The phrase tags have been designed for specific purposes, though they are displayed in a similar way as other basic tags like <b>, <i>, <pre>, and <tt>..

## Emphasized Text

Anything that appears within **<em>...</em>** element is displayed as emphasized text.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Emphasized Text Example</title>

</head>

<body>

<p>The following word uses a <em>emphasized</em> typeface.</p>

</body>

</html>

This will produce following result:

The following word uses a emphasized typeface.

## Marked Text

Anything that appears with-in **<mark>...</mark>** element, is displayed as marked with yellow ink.

<!DOCTYPE html>

<html>

<head>

<title>Marked Text Example</title>

</head>

<body>

<p>The following word has been <mark>marked</mark> with yellow</p>

</body>

</html>

This will produce following result:

The following word has been marked with yellow.

## Strong Text

Anything that appears within **<strong>...</strong>** element is displayed as important text.

<!DOCTYPE html>

<html>

<head>

<title>Strong Text Example</title>

</head>

<body>

<p>The following word uses a <strong>strong</strong> typeface.</p>

</body>

</html>

This will produce following result:

The following word uses a **strong** typeface.

## Text Abbreviation

You can abbreviate a text by putting it inside opening **<abbr>** and closing **</abbr>** tags. If present, the title attribute must contain this full description and nothing else.

<!DOCTYPE html>

<html>

<head>

<title>Text Abbreviation</title>

</head>

<body>

<p>My best friend's name is <abbr title="Abhishek">Abhy</abbr>.</p>

</body>

</html>

This will produce following result:

My best friend's name is Abhy.

## Quoting Text

When you want to quote a passage from another source, you should put it in between **<blockquote>...</blockquote>** tags.

Text inside a <blockquote> element is usually indented from the left and right edges of the surrounding text, and sometimes uses an italicized font.

<!DOCTYPE html>

<html>

<head>

<title>Blockquote Example</title>

</head>

<body>

<p>The following description of XHTML is taken from the W3C Web site:</p>

<blockquote>XHTML 1.0 is the W3C's first Recommendation for XHTML, following on from earlier work on HTML 4.01, HTML 4.0, HTML 3.2 and HTML 2.0.</blockquote>

</body>

</html>

This will produce following result:

The following description of XHTML is taken from the W3C Web site:

XHTML 1.0 is the W3C's first Recommendation for XHTML, following on from earlier work on HTML 4.01, HTML 4.0, HTML 3.2 and HTML 2.0.

## Short Quotations

The **<q>...</q>** element is used when you want to add a double quote within a sentence.

<!DOCTYPE html>

<html>

<head>

<title>Double Quote Example</title>

</head>

<body>

<p>Amit is in Spain, <q>I think I am wrong</q>.</p>

</body>

</html>

This will produce following result:

Amit is in Spain, I think I am wrong.

## Keyboard Text

When you are talking about computers, if you want to tell a reader to enter some text, you can use the **<kbd>...</kbd>** element to indicate what should be typed in, as in this example.

<!DOCTYPE html>

<html>

<head>

<title>Keyboard Text Example</title>

</head>

<body>

<p>Regular text. <kbd>This is inside kbd element</kbd> Regular text.</p>

</body>

</html>

This will produce following result:

Regular text. This is inside kbd element Regular text.

## Programming Variables

## <code> element displays its contents styled in a fashion intended to indicate that the text is a short fragment of computer code. By default, the content text is displayed using the user agent's default monospace font.

This element is usually used in conjunction with the **<pre>** and **<code>**elements to indicate that the content of that element is a variable.

<!DOCTYPE html>

<html>

<head>

<title>Variable Text Example</title>

</head>

<body>

<p><code>document.write("<var>user-name</var>")</code></p>

</body>

</html>

This will produce following result:

document.write("user-name")

## Program Output

The **<samp>...</samp>** element indicates sample output from a program, and script etc. Again, it is mainly used when documenting programming or coding concepts.

<!DOCTYPE html>

<html>

<head>

<title>Program Output Example</title>

</head>

<body>

<p>Result produced by the program is <samp>Hello World!</samp></p>

</body>

</html>

This will produce following result:

Result produced by the program is Hello World!

## Address Text

The **<address>...</address>** element is used to contain any address.

<!DOCTYPE html>

<html>

<head>

<title>Address Example</title>

</head>

<body>

<address>388A, Road No 22, Jubilee Hills - Hyderabad</address>

</body>

</html>

This will produce following result:

388A, Road No 22, Jubilee Hills - Hyderabad

**HTML Comments**

Comment is a piece of code which is ignored by any web browser. It is a good practice to add comments into your HTML code, especially in complex documents, to indicate sections of a document, and any other notes to anyone looking at the code. Comments help you and others understand your code and increases code readability.

HTML comments are placed in between **<!-- ... -->** tags. So any content placed with-in <!-- ... --> tags will be treated as comment and will be completely ignored by the browser.

<!DOCTYPE html>

<html>

<head> <!-- Document Header Starts -->

<title>This is document title</title>

</head> <!-- Document Header Ends -->

<body>

<p>Document content goes here.....</p>

</body>

</html>

This will produce following result without displaying the content given as a part of comments:

Document content goes here.....

**Valid vs Invalid Comments**

Comments do not nest which means a comment cannot be put inside another comment. Second the double-dash sequence "--" may not appear inside a comment except as part of the closing --> tag. You must also make sure that there are no spaces in the start-of-comment string.

Here given comment is a valid comment and will be wiped off by the browser.

<!DOCTYPE html>

<html>

<head>

<title>Valid Comment Example</title>

</head>

<body>

<!-- This is valid comment -->

<p>Document content goes here.....</p>

</body>

</html>

But following line is not a valid comment and will be displayed by the browser. This is because there is a space between the left angle bracket and the exclamation mark.

<!DOCTYPE html>

<html>

<head>

<title>Invalid Comment Example</title>

</head>

<body>

< !-- This is not a valid comment -->

<p>Document content goes here.....</p>

</body>

</html>

This will produce following result:

< !-- This is not a valid comment -->

Document content goes here.....

**Multiline Comments**

So far we have seen single line comments, but HTML supports multi-line comments as well.

You can comment multiple lines by the special beginning tag <!-- and ending tag --> placed before the first line and end of the last line as shown in the given example below.

<!DOCTYPE html><html>

<head>

<title>Multiline Comments</title>

</head>

<body>

<!--

This is a multiline comment and it can

span through as many as lines you like.

-->

<p>Document content goes here.....</p>

</body>

</html>

This will produce following result:

Document content goes here.....

**HTML Images**

Images are very important to beautify as well as to depict many complex concepts in simple way on your web page. This tutorial will take you through simple steps to use images in your web pages.

**Insert Image**

You can insert any image in your web page by using **<img>** tag. Following is the simple syntax to use this tag.

<img src="Image URL" ... attributes-list/>

The <img> tag is an empty tag, which means that it can contain only list of attributes and it has no closing tag.

<!DOCTYPE html>

<html>

<head>

<title>Using Image in Webpage</title>

</head>

<body>

<p>Simple Image Insert</p>

<img src="images/logo.png" alt="Test Image" />

</body>

</html>

This will produce following result:

Simple Image Insert



You can use PNG, JPEG or GIF image file based on your comfort but make sure you specify correct image file name in **src** attribute. Image name is always case sensitive.

The **alt** attribute is a not mandatory attribute which specifies an alternate text for an image, if the image cannot be displayed.

**Set Image Width/Height**

You can set image width and height based on your requirement using **width**and **height** attributes. You can specify width and height of the image in terms of either pixels or percentage of its actual size.

<!DOCTYPE html>

<html>

<head>

<title>Set Image Width and Height</title>

</head>

<body>

<p>Setting image width and height</p>

<img src="/html/images/test.png" alt="Test Image" width="150" height="100"/>

</body>

</html>

This will produce following result:

Setting image width and height



**Set Image Border**

By default image will have a border around it, you can specify border thickness in terms of pixels using **border** attribute. A thickness of 0 means, no border around the picture.

<!DOCTYPE html>

<html>

<head>

<title>Set Image Border</title>

</head>

<body>

<p>Setting image Border</p>

<img src="/html/images/test.png" alt="Test Image" border="3"/>

</body>

</html>

This will produce following result:

Setting image Border



**Set Image Alignment**

By default image will align at the left side of the page, but you can use **align**attribute to set it in the center or right.

Example

<!DOCTYPE html>

<html>

<head>

<title>Set Image Alignment</title>

</head>

<body>

<p>Setting image Alignment</p>

<img src="/html/images/test.png" alt="Test Image" border="3" align="right"/>

</body>

</html>

This will produce following result:

Setting image Alignment



**HTML Tables**

The HTML tables allow to arrange data like text, images, links, etc. into rows and columns of cells.

The HTML tables are created using the **<table>** tag in which the **<tr>** tag is used to create table rows and **<td>** tag is used to create data cells.

<!DOCTYPE html>

<html>

<head>

<title>HTML Tables</title>

</head>

<body>

<table border="1">

<tr>

<td>Row 1, Column 1</td>

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

</tr>

<tr>

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

<td>Row 2, Column 2</td>

</tr>

</table>

</body>

</html>

This will produce following result:

|  |  |
| --- | --- |
| Row 1, Column 1 | Row 1, Column 2 |
| Row 2, Column 1 | Row 2, Column 2 |

Here **border** is an attribute of <table> tag and it is used to put a border across all the cells. If you do not need a border then you can use border="0".

**Table Heading**

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

<!DOCTYPE html>

<html>

<head>

<title>HTML Table Header</title>

</head>

<body>

<table border="1">

<tr>

<th>Name</th>

<th>Salary</th>

</tr>

<tr>

<td>Ramesh Raman</td>

<td>5000</td>

</tr>

<tr>

<td>Shabbir Hussein</td>

<td>7000</td>

</tr>

</table>

</body>

</html>

This will produce following result:

|  |  |
| --- | --- |
| **Name** | **Salary** |
| Ramesh Raman | 5000 |
| Shabbir Hussein | 7000 |

**Cellpadding and Cellspacing Attributes**

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

<!DOCTYPE html>

<html>

<head>

<title>HTML Table Cellpadding</title>

</head>

<body>

<table border="1" cellpadding="5" cellspacing="5">

<tr>

<th>Name</th>

<th>Salary</th>

</tr>

<tr>

<td>Ramesh Raman</td>

<td>5000</td>

</tr>

<tr>

<td>Shabbir Hussein</td>

<td>7000</td>

</tr>

</table>

</body>

</html>

This will produce following result:

|  |  |
| --- | --- |
| **Name** | **Salary** |
| Ramesh Raman | 5000 |
| Shabbir Hussein | 7000 |

**Colspan and Rowspan Attributes**

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

<!DOCTYPE html>

<html>

<head>

<title>HTML Table Colspan/Rowspan</title>

</head>

<body>

<table border="1">

<tr>

<th>Column 1</th>

<th>Column 2</th>

<th>Column 3</th>

</tr>

<tr><td rowspan="2">Row 1 Cell 1</td><td>Row 1 Cell 2</td><td>Row 1 Cell 3</td></tr>

<tr><td>Row 2 Cell 2</td><td>Row 2 Cell 3</td></tr>

<tr><td colspan="3">Row 3 Cell 1</td></tr>

</table>

</body>

</html>

This will produce following result:

|  |  |  |
| --- | --- | --- |
| **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 | | |

**Tables Backgrounds**

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

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

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

<!DOCTYPE html>

<html>

<head>

<title>HTML Table Background</title>

</head>

<body>

<table border="1" bordercolor="green" bgcolor="yellow">

<tr>

<th>Column 1</th>

<th>Column 2</th>

<th>Column 3</th>

</tr>

<tr><td rowspan="2">Row 1 Cell 1</td><td>Row 1 Cell 2</td><td>Row 1 Cell 3</td></tr>

<tr><td>Row 2 Cell 2</td><td>Row 2 Cell 3</td></tr>

<tr><td colspan="3">Row 3 Cell 1</td></tr>

</table>

</body>

</html>

This will produce following result:

|  |  |  |
| --- | --- | --- |
| **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 | | |

<!DOCTYPE html>

<html>

<head>

<title>HTML Table Background</title>

</head>

<body>

<table border="1" bordercolor="green" background="/images/test.png">

<tr>

<th>Column 1</th>

<th>Column 2</th>

<th>Column 3</th>

</tr>

<tr><td rowspan="2">Row 1 Cell 1</td><td>Row 1 Cell 2</td><td>Row 1 Cell 3</td></tr>

<tr><td>Row 2 Cell 2</td><td>Row 2 Cell 3</td></tr>

<tr><td colspan="3">Row 3 Cell 1</td></tr>

</table>

</body>

</html>

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

|  |  |  |
| --- | --- | --- |
| **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 | | |

**Table Height and Width**

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

<!DOCTYPE html>

<html>

<head>

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

</head>

<body>

<table border="1" width="400" height="150">

<tr>

<td>Row 1, Column 1</td>

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

</tr>

<tr>

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

<td>Row 2, Column 2</td>

</tr>

</table>

</body>

</html>

This will produce following result:

|  |  |
| --- | --- |
| Row 1, Column 1 | Row 1, Column 2 |
| Row 2, Column 1 | Row 2, Column 2 |

**Table Caption**

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

<!DOCTYPE html>

<html>

<head>

<title>HTML Table Caption</title>

</head>

<body>

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

<caption>This is the caption</caption>

<tr>

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

</tr>

</table>

</body>

</html>

This will produce following result:

|  |  |
| --- | --- |
| This is the caption | |
| row 1, column 1 | row 1, columnn 2 |

**Table Header, Body, and Footer**

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

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

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

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

<!DOCTYPE html>

<html>

<head>

<title>HTML Table</title>

</head>

<body>

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

<thead>

<tr>

<td colspan="4">This is the head of the table</td>

</tr>

</thead>

<tfoot>

<tr>

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

</tr>

</tfoot>

<tbody>

<tr>

<td>Cell 1</td>

<td>Cell 2</td>

<td>Cell 3</td>

<td>Cell 4</td>

</tr>

</tbody>

</table>

</body>

</html>

This will produce following result:

| This is the head of the table | | | |
| --- | --- | --- | --- |
| This is the foot of the table | | | |
| Cell 1 | Cell 2 | Cell 3 | Cell 4 |

**Nested Tables**

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

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

<!DOCTYPE html>

<html>

<head>

<title>HTML Table</title>

</head>

<body>

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

<tr>

<td>

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

<tr>

<th>Name</th>

<th>Salary</th>

</tr>

<tr>

<td>Ramesh Raman</td>

<td>5000</td>

</tr>

<tr>

<td>Shabbir Hussein</td>

<td>7000</td>

</tr>

</table>

</td>

</tr>

</table>

</body>

</html>

This will produce following result:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Name** | **Salary** | | Ramesh Raman | 5000 | | Shabbir Hussein | 7000 | |

**HTML Lists**

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

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

**HTML Unordered Lists**

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

<!DOCTYPE html>

<html>

<head>

<title>HTML Unordered List</title>

</head>

<body>

<ul>

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

<li>Radish</li>

</ul>

</body>

</html>

This will produce following result:

* Beetroot
* Ginger
* Potato
* Radish

**The type Attribute**

You can use **type** attribute for <ul> tag to specify the type of bullet you like. By default it is a disc. Following are the possible options:

<ul type="square">

<ul type="disc">

<ul type="circle">

Following is an example where we used <ul type="square">

<!DOCTYPE html>

<html>

<head>

<title>HTML Unordered List</title>

</head>

<body>

<ul type="square">

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

<li>Radish</li>

</ul>

</body>

</html>

This will produce following result:

* Beetroot
* Ginger
* Potato
* Radish

Following is an example where we used <ul type="disc"> :

<!DOCTYPE html>

<html>

<head>

<title>HTML Unordered List</title>

</head>

<body>

<ul type="disc">

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

<li>Radish</li>

</ul>

</body>

</html>

This will produce following result:

* Beetroot
* Ginger
* Potato
* Radish

Following is an example where we used <ul type="circle"> :

<!DOCTYPE html>

<html>

<head>

<title>HTML Unordered List</title>

</head>

<body>

<ul type="circle">

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

<li>Radish</li>

</ul>

</body>

</html>

This will produce following result:

* Beetroot
* Ginger
* Potato
* Radish

**HTML Ordered Lists**

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

<!DOCTYPE html>

<html>

<head>

<title>HTML Ordered List</title>

</head>

<body>

<ol>

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

<li>Radish</li>

</ol>

</body>

</html>

This will produce following result:

1. Beetroot
2. Ginger
3. Potato
4. Radish

**The type Attribute**

You can use **type** attribute for <ol> tag to specify the type of numbering you like. By default it is a number. Following are the possible options:

<ol type="1"> - Default-Case Numerals.

<ol type="I"> - Upper-Case Numerals.

<ol type="i"> - Lower-Case Numerals.

<ol type="a"> - Lower-Case Letters.

<ol type="A"> - Upper-Case Letters.

Following is an example where we used <ol type="1">

<!DOCTYPE html>

<html>

<head>

<title>HTML Ordered List</title>

</head>

<body>

<ol type="1">

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

<li>Radish</li>

</ol>

</body>

</html>

This will produce following result:

1. Beetroot
2. Ginger
3. Potato
4. Radish

Following is an example where we used <ol type="I">

<!DOCTYPE html>

<html>

<head>

<title>HTML Ordered List</title>

</head>

<body>

<ol type="I">

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

<li>Radish</li>

</ol>

</body>

</html>

This will produce following result:

1. Beetroot
2. Ginger
3. Potato
4. Radish

Following is an example where we used <ol type="i">

<!DOCTYPE html>

<html>

<head>

<title>HTML Ordered List</title>

</head>

<body>

<ol type="i">

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

<li>Radish</li>

</ol>

</body>

</html>

This will produce following result:

1. Beetroot
2. Ginger
3. Potato
4. Radish

Following is an example where we used <ol type="A">

<!DOCTYPE html>

<html>

<head>

<title>HTML Ordered List</title>

</head>

<body>

<ol type="A">

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

<li>Radish</li>

</ol>

</body>

</html>

This will produce following result:

1. Beetroot
2. Ginger
3. Potato
4. Radish

Following is an example where we used <ol type="a">

<!DOCTYPE html>

<html>

<head>

<title>HTML Ordered List</title>

</head>

<body>

<ol type="a">

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

<li>Radish</li>

</ol>

</body>

</html>

This will produce following result:

1. Beetroot
2. Ginger
3. Potato
4. Radish

**The start Attribute**

You can use **start** attribute for <ol> tag to specify the starting point of numbering you need. Following are the possible options:

<ol type="1" start="4"> - Numerals starts with 4.

<ol type="I" start="4"> - Numerals starts with IV.

<ol type="i" start="4"> - Numerals starts with iv.

<ol type="a" start="4"> - Letters starts with d.

<ol type="A" start="4"> - Letters starts with D.

Following is an example where we used <ol type="i" start="4" >

<!DOCTYPE html>

<html>

<head>

<title>HTML Ordered List</title>

</head>

<body>

<ol type="i" start="4">

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

</ol>

</body>

</html>

This will produce following result:

1. Beetroot
2. Ginger
3. Potato

**HTML Definition Lists**

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

Definition List makes use of following three tags.

* <dl> - Defines the start of the list
* <dt> - A term
* <dd> - Term definition
* </dl> - Defines the end of the list

<!DOCTYPE html>

<html>

<head>

<title>HTML Definition List</title>

</head>

<body>

<dl>

<dt><b>HTML</b></dt>

<dd>This stands for Hyper Text Markup Language</dd>

<dt><b>HTTP</b></dt>

<dd>This stands for Hyper Text Transfer Protocol</dd>

</dl>

</body>

</html>

This will produce following result:

**HTML**

This stands for Hyper Text Markup Language

**HTTP**

This stands for Hyper Text Transfer Protocol

**HTML Text Links**

A webpage can contain various links that take you directly to other pages and even specific parts of a given page. These links are known as hyperlinks.

Hyperlinks allow visitors to navigate between Web sites by clicking on words and images. Thus you can create hyperlinks using text or images available on a webpage.

**Linking Documents**

A link is specified using HTML tag <a>. This tag is called **anchor tag** and anything between the opening <a> tag and the closing </a> tag becomes part of the link and a user can click that part to reach to the linked document. Following is the simple syntax to use <a> tag.

<a href="Document URL" ... attributes-list>Link Text</a>

Let's try following example which links http://www.elearninfotech.com at your page:

<!DOCTYPE html>

<html>

<head>

<title>Hyperlink Example</title>

</head>

<body>

<p>Click following link</p>

<a href="http://www.elearninfotech.com" target="\_self">ELearn Infotech</a>

</body>

</html>

This will produce following result, where you can click on the link generated **ELearn Infotech** to reach to the home page of ELearn Infotech.

Click following link

[ELearn Infotech](https://www.tutorialspoint.com/)

**The target Attribute**

We have used **target** attribute in our previous example. This attribute is used to specify the location where linked document is opened. Following are possible options:

|  |  |
| --- | --- |
| **Option** | **Description** |
| \_blank | Opens the linked document in a new window or tab. |
| \_self | Opens the linked document in the same frame. |
| \_parent | Opens the linked document in the parent frame. |
| \_top | Opens the linked document in the full body of the window. |

Try following example to understand basic difference in few options given for target attribute.

<!DOCTYPE html>

<html>

<head>

<title>Hyperlink Example</title>

<base href="http://www.elearninfotech.com/">

</head>

<body>

<p>Click any of the following links</p>

<a href="/html/index.htm" target="\_blank">Opens in New</a> |

<a href="/html/index.htm" target="\_self">Opens in Self</a> |

<a href="/html/index.htm" target="\_parent">Opens in Parent</a> |

<a href="/html/index.htm" target="\_top">Opens in Body</a>

</body>

</html>

This will produce following result, where you can click on different links to understand the difference between various options given for target attribute.

Click any of the following links

[Opens in New](https://www.tutorialspoint.com/html/index.htm) | [Opens in Self](https://www.tutorialspoint.com/html/index.htm) | [Opens in Parent](https://www.tutorialspoint.com/html/index.htm) | [Opens in Body](https://www.tutorialspoint.com/html/index.htm)

**Use of Base Path**

When you link HTML documents related to the same website, it is not required to give a complete URL for every link. You can get rid of it if you use **<base>**tag in your HTML document header. This tag is used to give a base path for all the links. So your browser will concatenate given relative path to this base path and will make a complete URL.

Following example makes use of <base> tag to specify base URL and later we can use relative path to all the links instead of giving complete URL for every link.

<!DOCTYPE html>

<html>

<head>

<title>Hyperlink Example</title>

<base href="http://www.elearninfotech.com/">

</head>

<body>

<p>Click following link</p>

<a href="index.htm" target="\_blank">HTML Tutorial</a>

</body>

</html>

This will produce following result, where you can click on the link generated **HTML Tutorial** to reach to the HTML tutorial.

Now given URL **<a href="index.htm"** is being considered as **<a href="http://www.elearninfotech.com/index.htm"**.

Click following link

[HTML Tutorial](https://www.tutorialspoint.com/html/index.htm)

**Linking to a Page Section**

You can create a link to a particular section of a given webpage by using **name** attribute. This is a two step process.

First create a link to the place where you want to reach with-in a webpage and name it using <a...> tag as follows:

<h1>HTML Text Links <a name="top"></a></h1>

Second step is to create a hyperlink to link the document and place where you want to reach:

<a href="/html/html\_text\_links.htm#top">Go to the Top</a>

This will produce following link, where you can click on the link generated **Go to the Top** to reach to the top of the HTML Text Link tutorial.

|  |
| --- |
| [**Go to the Top**](https://www.tutorialspoint.com/html/html_text_links.htm#top) |

**Setting Link Colors**

You can set colors of your links, active links and visited links using **link**, **alink**and **vlink** attributes of <body> tag.

<!DOCTYPE html>

<html>

<head>

<title>Hyperlink Example</title>

<base href="http://www.elearninfotech.com/">

</head>

<body alink="#54A250" link="#040404" vlink="#F40633">

<p>Click following link</p>

<a href="/html/index.htm" target="\_blank" >HTML Tutorial</a>

</body>

</html>

This will produce following result. Just check color of the link before clicking on it, next check its color when you activate it and when the link has been visited.

Click following link

[HTML Tutorial](https://www.tutorialspoint.com/html/index.htm)

**Download Links**

You can create text link to make your PDF, or DOC or ZIP files downloadable. This is very simple, you just need to give complete URL of the downloadable file as follows:

<!DOCTYPE html>

<html>

<head>

<title>Hyperlink Example</title>

</head>

<a href="http://www.elearninfotech.com/page.pdf">Download PDF File</a>

</body>

</html>

This will produce following link and will be used to download a file.

[Download PDF File](https://www.tutorialspoint.com/page.pdf)

**HTML Image Links**

We have seen how to create hypertext link using text and we also learnt how to use images in our webpages. Now we will learn how to use images to create hyperlinks.

It's simple to use an image as hyperlink. We just need to use an image inside hyperlink at the place of text as shown below:

<!DOCTYPE html>

<html>

<head>

<title>Image Hyperlink Example</title>

</head>

<body>

<p>Click following link</p>

<a href="http://www.elearninfotech.com" target="\_self">

<img src="/images/logo.png" alt="ELearn Infotech" border="0"/>

</a>

</body>

</html>

This will produce following result, where you can click on the images to reach to the home page of ELearn Infotech.

Click following link



This was the simplest way of creating hyperlinks using images. Next we will see how we can create Mouse-Sensitive Image Links.

**HTML Email Links**

Its not difficult to put an HTML email link on your webpage but it can cause unnecessary spamming problem for your email account. There are people who can run programs to harvest these types of emails and later use them for spamming in various ways.

You can have other options to facilitate people to send you emails. One option could be to use HTML forms to collect user data and then use PHP or ASP script to send an email.

**HTML Email Tag**

HTML <a> tag provides you option to specifiy 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.

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

This code will generate following link which you can use to send email.

[Send Email](mailto:abc@example.com)

**Default Settings**

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

<a href="mailto:abc@example.com?subject=Feedback&body=Message">

Send Feedback

</a>

This code will generate following link which you can use to send email.

[Send Feedback](mailto:abc@example.com?subject=Feedback&body=Message)

**HTML Iframes**

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.

Following is the example to show how to use the <iframe>:

<!DOCTYPE html>

<html>

<head>

<title>HTML Iframes</title>

</head>

<body>

<p>Document content goes here...</p>

<iframe src="/html/menu.htm" width="555" height="200">

Sorry your browser does not support inline frames.

</iframe>

<p>Document content also go here...</p>

</body>

</html>

This will produce following result:

Document content goes here...

Document content can also go here...

**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 avalaible 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". |

**HTML Blocks**

All the HTML elements can be categorized into two categories **(a)** Block Level Elements **(b)** Inline Elements

**Block Elements**

Block elements appear on the screen as if they have a line break before and after them. For example the <p>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <ul>, <ol>, <dl>, <pre>, <hr />, <blockquote>, and <address> elements are all block level elements. They all start on their own new line, and anything that follows them appears on its own new line.

**Inline Elements**

Inline elements, on the other hand, can appear within sentences and do not have to appear on a new line of their own. The <b>, <i>, <u>, <em>, <strong>, <sup>, <sub>, <big>, <small>, <li>, <ins>, <del>, <code>, <cite>, <dfn>, <kbd>, and <var> elements are all inline elements.

**Grouping HTML Elements**

There are two important tags which we use very frequently to group various other HTML tags (i) <div> tag and (ii) <span> tag

The <div> tag

This is the very important block level tag which plays a big role in grouping various other HTML tags and applying CSS on group of elements. Even now <div> tag can be used to create webpage layout where we define different parts ( Left, Right, Top etc) of the page using <div> tag. This tag does not provide any visual change on the block but this has more meaning when it is used with CSS.

Following is a simple example of <div> tag. We will learn Cascading Style Sheet (CSS) in a separate chapter but we used it here to show the usage of <div> tag:

<!DOCTYPE html>

<html>

<head>

<title>HTML div Tag</title>

</head>

<body>

<!-- First group of tags -->

<div style="color:red">

<h4>This is first group</h4>

<p>Following is a list of vegetables</p>

<ul>

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

<li>Radish</li>

</ul>

</div>

<!-- Second group of tags -->

<div style="color:green">

<h4>This is second group</h4>

<p>Following is a list of fruits</p>

<ul>

<li>Apple</li>

<li>Banana</li>

<li>Mango</li>

<li>Strawberry</li>

</ul>

</div>

</body>

</html>

This will produce following result:

**The <span> tag**

The HTML <span> is an inline element and it can be used to group inline-elements in an HTML document. This tag also does not provide any visual change on the block but has more meaning when it is used with CSS.

The difference between the <span> tag and the <div> tag is that the <span> tag is used with inline elements where as the <div> tag is used with block-level elements.

Following is a simple example of <span> tag. We will learn Cascading Style Sheet (CSS) in a separate chapter but we used it here to show the usage of <span> tag:

<!DOCTYPE html>

<html>

<head>

<title>HTML span Tag</title>

</head>

<body>

<p>This is <span style="color:red">red</span> and this is <span style="color:green">green</span></p>

</body>

</html>

This will produce following result:

This is red, and this is green

**HTML Backgrounds**

HTML provides you following two good ways to decorate your webpage background.

* Html Background with Colors
* Html Background with Images

**Html Background with Colors**

The **bgcolor** attribute is used to control the background of an HTML element, specifically page body and table backgrounds. Following is the syntax to use bgcolor attribute with any HTML tag.

<tagname bgcolor="color\_value"...>

This color\_value can be given in any of the following formats:

<!-- Format 1 - Use color name -->

<table bgcolor="lime" >

<!-- Format 2 - Use hex value -->

<table bgcolor="#f1f1f1" >

<!-- Format 3 - Use color value in RGB terms -->

<table bgcolor="rgb(0,0,120)" >

Here are the examples to set background of an HTML tag:

<!DOCTYPE html>

<html>

<head>

<title>HTML Background Colors</title>

</head>

<body>

<!-- Format 1 - Use color name -->

<table bgcolor="yellow" width="100%">

<tr><td>

This background is yellow

</td></tr>

</table>

<!-- Format 2 - Use hex value -->

<table bgcolor="#6666FF" width="100%">

<tr><td>

This background is sky blue

</td></tr>

</table>

<!-- Format 3 - Use color value in RGB terms -->

<table bgcolor="rgb(255,0,255)" width="100%">

<tr><td>

This background is green

</td></tr>

</table>

</body>

</html>

This will produce following result:

|  |
| --- |
| This background is yellow |

|  |
| --- |
| This background is sky blue |

|  |
| --- |
| This background is green |

**Html Background with Images**

The **background** attribute can also be used to control the background of an HTML elmement, specifically page body and table backgrounds. You can specify an image to set background of your HTML page or table. Following is the syntax to use background attribute with any HTML tag.

**Note:**The *background* attribute is deprecated and it is recommended to use Style Sheet for background setting.

<tagname background="Image URL"...>

The most frequently used image formats are JPEG, GIF and PNG images.

Here are the examples to set background images of a table.

<!DOCTYPE html>

<html>

<head>

<title>HTML Background Images</title>

</head>

<body>

<!-- Set table background -->

<table background="/images/html.gif" width="100%" height="100">

<tr><td>

This background is filled up with HTML image.

</td></tr>

</table>

</body>

</html>

This will produce following result:

|  |
| --- |
| This background is filled up with HTML image. |

**Transparent Backgrounds**

You might have seen many pattern or transparent backgrounds on various websites. This simply can be achieved by using patterned image or transparent image in the background.

It is suggested that while creating patterns or transparent GIF or PNG images, use the smallest dimensions possible even as small as 1x1 to avoid slow loading.

<!DOCTYPE html>

<html>

<head>

<title>HTML Background Images</title>

</head>

<body>

<!-- Set a table background using pattrern -->

<table background="/images/pattern1.gif" width="100%" height="100">

<tr><td>

This background is filled up with a pattern image.

</td></tr>

</table>

<!-- Another on table background using pattrern -->

<table background="/images/pattern2.gif" width="100%" height="100">

<tr><td>

This background is filled up with a pattern image.

</td></tr>

</table>

</body>

</html>

This will produce following result:

|  |
| --- |
| This background is filled up with a pattern image. |

|  |
| --- |
| This background is filled up with a pattern image. |

# HTML Colors

Colors are very important to give a good look and feel to your website. You can specify colors on page level using <body> tag or you can set colors for individual tags using **bgcolor** attribute.

The <body> tag has following attributes which can be used to set different colors:

* **bgcolor -** sets a color for the background of the page.
* **text -** sets a color for the body text.
* **alink -** sets a color for active links or selected links.
* **link -** sets a color for linked text.
* **vlink -** sets a color for *visited links* - that is, for linked text that you have already clicked on.

## HTML Color Coding Methods

There are following three different methods to set colors in your web page:

* **Color names -** You can specify color names directly like green, blue or red.
* **Hex codes -** A six-digit code representing the amount of red, green, and blue that makes up the color.
* **Color decimal or percentage values -** This value is specified using the rgb( ) property.

Now we will see these coloring schemes one by one.

## HTML Colors - Color Names

You can sepecify direct a color name to set text or background color. W3C has listed 16 basic color names that will validate with an HTML validator but there are over 200 different color names supported by major browsers.

### W3C Standard 16 Colors

Here is the list of W3C Standard 16 Colors names and it is recommended to use them.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Black |  | Gray |  | Silver |  | White |
|  | Yellow |  | Lime |  | Aqua |  | Fuchsia |
|  | Red |  | Green |  | Blue |  | Purple |
|  | Maroon |  | Olive |  | Navy |  | Teal |

Here are the s to set background of an HTML tag by color name:

<!DOCTYPE html>

<html>

<head>

<title>HTML Colors by Name</title>

</head>

<body text="blue" bgcolor="green">

<p>Use different color names for for body and table and see the result.</p>

<table bgcolor="black">

<tr>

<td>

<font color="white">This text will appear white on black background.</font>

</td>

</tr>

</table>

</body>

</html>

### HTML Colors - Hex Codes

A hexadecimal is a 6 digit representation of a color. The first two digits(RR) represent a red value, the next two are a green value(GG), and the last are the blue value(BB).

A hexadecimal value can be taken from any graphics software like Adobe Photoshop, Paint shop Pro or MS Paint.

Each hexadecimal code will be preceded by a pound or hash sign #. Following is a list of few colors using hexadecimal notation.

|  |  |
| --- | --- |
| **Color** | **Color HEX** |
|  | #000000 |
|  | #FF0000 |
|  | #00FF00 |
|  | #0000FF |
|  | #FFFF00 |
|  | #00FFFF |
|  | #FF00FF |
|  | #C0C0C0 |
|  | #FFFFFF |

Here are the s to set background of an HTML tag by color code in hexadecimal:

<!DOCTYPE html>

<html>

<head>

<title>HTML Colors by Hex</title>

</head>

<body text="#0000FF" bgcolor="#00FF00">

<p>Use different color hexa for for body and table and see the result.</p>

<table bgcolor="#000000">

<tr>

<td>

<font color="#FFFFFF">This text will appear white on black background.</font>

</td>

</tr>

</table>

</body>

</html>

## HTML Colors - RGB Values

This color value is specified using the **rgb( )** property. This property takes three values, one each for red, green, and blue. The value can be an integer between 0 and 255 or a percentage.

Following is a list to show few colors using RGB values.

|  |  |
| --- | --- |
| **Color** | **Color RGB** |
|  | rgb(0,0,0) |
|  | rgb(255,0,0) |
|  | rgb(0,255,0) |
|  | rgb(0,0,255) |
|  | rgb(255,255,0) |
|  | rgb(0,255,255) |
|  | rgb(255,0,255) |
|  | rgb(192,192,192) |
|  | rgb(255,255,255) |

Here are the s to set background of an HTML tag by color code using rgb() values:

<!DOCTYPE html>

<html>

<head>

<title>HTML Colors by RGB code</title>

</head>

<body text="rgb(0,0,255)" bgcolor="rgb(0,255,0)">

<p>Use different color code for for body and table and see the result.</p>

<table bgcolor="rgb(0,0,0)">

<tr>

<td>

<font color="rgb(255,255,255)">This text will appear white on black background.</font>

</td>

</tr>

</table>

</body>

</html>

**HTML Fonts**

Fonts play very important role in making a website more user friendly and increasing content readability. Font face and color depends entirely on the computer and browser that is being used to view your page but you can use HTML **<font>** tag to add style, size, and color to the text on your website. You can use a **<basefont>** tag to set all of your text to the same size, face, and color.

The font tag is having three attributes called **size, color**, and **face** to customize your fonts. To change any of the font attributes at any time within your webpage, simply use the <font> tag. The text that follows will remain changed until you close with the </font> tag.

**Set Font Size**

You can set content font size using **size** attribute. The range of accepted values is from 1(smallest) to 7(largest). The default size of a font is 3.

<!DOCTYPE html>

<html>

<head>

<title>Setting Font Size</title>

</head>

<body>

<font size="1">Font size="1"</font><br />

<font size="2">Font size="2"</font><br />

<font size="3">Font size="3"</font><br />

<font size="4">Font size="4"</font><br />

<font size="5">Font size="5"</font><br />

<font size="6">Font size="6"</font><br />

<font size="7">Font size="7"</font>

</body>

</html>

This will produce following result:

Font size="1"  
Font size="2"  
Font size="3"  
Font size="4"  
Font size="5"  
Font size="6"  
Font size="7"

**Setting Font Face**

You can set font face using *face* attribute but be aware that if the user viewing the page doesn't have the font installed, they will not be able to see it. Instead user will see the default font face applicable to the user's computer.

<!DOCTYPE html>

<html>

<head>

<title>Font Face</title>

</head>

<body>

<font face="Times New Roman" size="5">Times New Roman</font><br />

<font face="Verdana" size="5">Verdana</font><br />

<font face="Comic sans MS" size="5">Comic Sans MS</font><br />

<font face="WildWest" size="5">WildWest</font><br />

<font face="Bedrock" size="5">Bedrock</font><br />

</body>

</html>

This will produce following result:

Times New Roman  
Verdana  
Comic Sans MS  
WildWest  
Bedrock

**Specify alternate font faces**

A visitor will only be able to see your font if they have that font installed on their computer. So, it is possible to specify two or more font face alternatives by listing the font face names, separated by a comma.

<font face="arial,helvetica">

<font face="Lucida Calligraphy,Comic Sans MS,Lucida Console">

When your page is loaded, their browser will display the first font face available. If none of the given fonts are installed, then it will display the default font face *Times New Roman*.

**Setting Font Color**

You can set any font color you like using *color* attribute. You can specify the color that you want by either the color name or hexadecimal code for that color.

<!DOCTYPE html>

<html>

<head>

<title>Setting Font Color</title>

</head>

<body>

<font color="#FF00FF">This text is in pink</font><br />

<font color="red">This text is red</font>

</body>

</html>

This will produce following result:

This text is in pink  
This text is red

**The <basefont> Element:**

The <basefont> element is supposed to set a default font size, color, and typeface for any parts of the document that are not otherwise contained within a <font> tag. You can use the <font> elements to override the <basefont> settings.

The <basefont> tag also takes color, size and face attributes and it will support relative font setting by giving size a value of +1 for a size larger or -2 for two sizes smaller.

<!DOCTYPE html>

<html>

<head>

<title>Setting Basefont Color</title>

</head>

<body>

<basefont face="arial, verdana, sans-serif" size="2" color="#ff0000">

<p>This is the page's default font.</p>

<h2> of the &lt;basefont&gt; Element</h2>

<p><font size="+2" color="darkgray">

This is darkgray text with two sizes larger

</font></p>

<p><font face="courier" size="-1" color="#000000">

It is a courier font, a size smaller and black in color.

</font></p>

</body>

</html>

This will produce following result:

This is the page's default font.

of the <basefont> Element

This is darkgray text with two sizes larger

It is a courier font, a size smaller and black in color.

**HTML Forms**

HTML Forms are required when you want to collect some data from the site visitor. For during user registration you would like to collect information such as name, email address, credit card, etc.

A form will take input from the site visitor and then will post it to a back-end application such as ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

There are various form elements available like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.

The HTML **<form>** tag is used to create an HTML form and it has following syntax:

<form action="Script URL" method="GET|POST">

form elements like input, textarea etc.

</form>

**Form Attributes**

Apart from common attributes, following is a list of the most frequently used form attributes:

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| action | Backend script ready to process your passed data. |
| method | Method to be used to upload data. The most frequently used are GET and POST methods. |

**HTML Form Controls**

There are different types of form controls that you can use to collect data using HTML form:

* Text Input Controls
* Checkboxes Controls
* Radio Box Controls
* Select Box Controls
* File Select boxes
* Hidden Controls
* Clickable Buttons
* Submit and Reset Button

**Text Input Controls**

There are three types of text input used on forms:

* **Single-line text input controls -** This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML **<input>** tag.
* **Password input controls -** This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTMl <input> tag.
* **Multi-line text input controls -** This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML **<textarea>** tag.

**Single-line text input controls**

This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML <input> tag.

<!DOCTYPE html>

<html>

<head>

<title>Text Input Control</title>

</head>

<body>

<form >

First name: <input type="text" name="first\_name" />

<br>

Last name: <input type="text" name="last\_name" />

</form>

</body>

</html>

This will produce following result:

Top of Form

First name:    
Last name: 

Bottom of Form

Attributes

Following is the list of attributes for <input> tag for creating text field.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| type | Indicates the type of input control and for text input control it will be set to **text**. |
| name | Used to give a name to the control which is sent to the server to be recognized and get the value. |
| value | This can be used to provide an initial value inside the control. |
| size | Allows to specify the width of the text-input control in terms of characters. |
| maxlength | Allows to specify the maximum number of characters a user can enter into the text box. |

**Password input controls**

This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML <input> tag but type attribute is set to **password**.

<!DOCTYPE html>

<html>

<head>

<title>Password Input Control</title>

</head>

<body>

<form >

User ID : <input type="text" name="user\_id" />

<br>

Password: <input type="password" name="password" />

</form>

</body>

</html>

This will produce following result:

Top of Form

User ID :      
Password: 

Bottom of Form

Attributes

Following is the list of attributes for <input> tag for creating password field.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| type | Indicates the type of input control and for password input control it will be set to **password**. |
| name | Used to give a name to the control which is sent to the server to be recognized and get the value. |
| value | This can be used to provide an initial value inside the control. |
| size | Allows to specify the width of the text-input control in terms of characters. |
| maxlength | Allows to specify the maximum number of characters a user can enter into the text box. |

**Multiple-Line Text Input Controls**

This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.

<!DOCTYPE html>

<html>

<head>

<title>Multiple-Line Input Control</title>

</head>

<body>

<form>

Description : <br />

<textarea rows="5" cols="50" name="description">

Enter description here...

</textarea>

</form>

</body>

</html>

This will produce following result:

Top of Form

Description :   


Bottom of Form

Attributes

Following is the list of attributes for <textarea> tag.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| name | Used to give a name to the control which is sent to the server to be recognized and get the value. |
| rows | Indicates the number of rows of text area box. |
| cols | Indicates the number of columns of text area box |

**Checkbox Control**

Checkboxes are used when more than one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to **checkbox**.

Here is an HTML code for a form with two checkboxes:

<!DOCTYPE html>

<html>

<head>

<title>Checkbox Control</title>

</head>

<body>

<form>

<input type="checkbox" name="maths" value="on"> Maths

<input type="checkbox" name="physics" value="on"> Physics

</form>

</body>

</html>

This will produce following result:

Top of Form

 Maths  Physics

Bottom of Form

Attributes

Following is the list of attributes for <checkbox> tag.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| type | Indicates the type of input control and for checkbox input control it will be set to **checkbox**. |
| name | Used to give a name to the control which is sent to the server to be recognized and get the value. |
| value | The value that will be used if the checkbox is selected. |
| checked | Set to *checked* if you want to select it by default. |

**Radio Button Control**

Radio buttons are used when out of many options, just one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to **radio**.

<!DOCTYPE html>

<html>

<head>

<title>Radio Box Control</title>

</head>

<body>

<form>

<input type="radio" name="subject" value="maths"> Maths

<input type="radio" name="subject" value="physics"> Physics

</form>

</body>

</html>

This will produce following result:

Top of Form

 Maths  Physics

Bottom of Form

Attributes

Following is the list of attributes for radio button.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| type | Indicates the type of input control and for checkbox input control it will be set to **radio**. |
| name | Used to give a name to the control which is sent to the server to be recognized and get the value. |
| value | The value that will be used if the radio box is selected. |
| checked | Set to *checked* if you want to select it by default. |

**Select Box Control**

A select box, also called drop down box which provides option to list down various options in the form of drop down list, from where a user can select one or more options.

<!DOCTYPE html>

<html>

<head>

<title>Select Box Control</title>

</head>

<body>

<form>

<select name="dropdown">

<option value="Maths" selected>Maths</option>

<option value="Physics">Physics</option>

</select>

</form>

</body>

</html>

This will produce following result:

Top of Form



Bottom of Form

Attributes

Following is the list of important attributes of <select> tag:

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| name | Used to give a name to the control which is sent to the server to be recognized and get the value. |
| size | This can be used to present a scrolling list box. |
| multiple | If set to "multiple" then allows a user to select multiple items from the menu. |

Following is the list of important attributes of <option> tag:

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| value | The value that will be used if an option in the select box box is selected. |
| selected | Specifies that this option should be the initially selected value when the page loads. |
| label | An alternative way of labeling options |

**File Upload Box**

If you want to allow a user to upload a file to your web site, you will need to use a file upload box, also known as a file select box. This is also created using the <input> element but type attribute is set to **file**.

<!DOCTYPE html>

<html>

<head>

<title>File Upload Box</title>

</head>

<body>

<form>

<input type="file" name="fileupload" accept="image/\*" />

</form>

</body>

</html>

This will produce following result:

Top of Form

Bottom of Form

Attributes

Following is the list of important attributes of file upload box:

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| name | Used to give a name to the control which is sent to the server to be recognized and get the value. |
| accept | Specifies the types of files that the server accepts. |

**Button Controls**

There are various ways in HTML to create clickable buttons. You can also create a clickable button using <input> tag by setting its type attribute to **button**. The type attribute can take the following values:

|  |  |
| --- | --- |
| **Type** | **Description** |
| submit | This creates a button that automatically submits a form. |
| reset | This creates a button that automatically resets form controls to their initial values. |
| button | This creates a button that is used to trigger a client-side script when the user clicks that button. |
| image | This creates a clickable button but we can use an image as background of the button. |

<!DOCTYPE html>

<html>

<head>

<title>File Upload Box</title>

</head>

<body>

<form>

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

<input type="reset" name="reset" value="Reset" />

<input type="button" name="ok" value="OK" />

<input type="image" name="imagebutton" src="/html/images/logo.png" />

</form>

</body>

</html>

This will produce following result:

Top of Form

**Bottom of Form**

**Hidden Form Controls**

Hidden form controls are used to hide data inside the page which later on can be pushed to the server. This control hides inside the code and does not appear on the actual page. For , following hidden form is being used to keep current page number. When a user will click next page then the value of hidden control will be sent to the web server and there it will decide which page has be displayed next based on the passed current page.

<!DOCTYPE html>

<html>

<head>

<title>File Upload Box</title>

</head>

<body>

<form>

<p>This is page 10</p>

<input type="hidden" name="pagename" value="10" />

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

<input type="reset" name="reset" value="Reset" />

</form>

</body>

</html>

This will produce following result:

Top of Form

This is page 10

Bottom of Form

**HTML Marquees**

An HTML marquee is a scrolling piece of text displayed either horizontally across or vertically down your webpage depending on the settings. This is created by using HTML <marquees> tag.

Syntax

A simple syntax to use HTML <marquee> tag is as follows:

<marquee attribute\_name="attribute\_value"....more attributes>

One or more lines or text message or image

</marquee>

**The <marquee> Tag Attributes**

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

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| width | This specifies the width of the marquee. This can be a value like 10 or 20% etc. |
| height | This specifies the height of the marquee. This can be a value like 10 or 20% etc. |
| direction | This specifies the direction in which marquee should scroll. This can be a value like *up*, *down*, *left* or *right*. |
| behavior | This specifies the type of scrolling of the marquee. This can have a value like *scroll*, *slide* and *alternate*. |
| scrolldelay | This specifies how long to delay between each jump. This will have a value like 10 etc. |
| scrollamount | This specifies the speed of marquee text. This can have a value like 10 etc. |
| loop | This specifies how many times to loop. The default value is INFINITE, which means that the marquee loops endlessly. |
| bgcolor | This specifies background color in terms of color name or color hex value. |

Below are few s to demonstrate the usage of marquee tag.

s - 1

<!DOCTYPE html>

<html>

<head>

<title>HTML marquee Tag</title>

</head>

<body>

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

</body>

</html>

This will produce following result:



s - 2

<!DOCTYPE html>

<html>

<head>

<title>HTML marquee Tag</title>

</head>

<body>

<marquee width="50%">This will take only 50% width</marquee>

</body>

</html>

This will produce following result:



s - 3

<!DOCTYPE html>

<html>

<head>

<title>HTML marquee Tag</title>

</head>

<body>

<marquee direction="right">This text will scroll from left to right</marquee>

</body>

</html>

This will produce following result:



**HTML Layouts**

A webpage layout is very important to give better look to your website. It takes considerable time to design a website's layout with great look and feel.

Now a days, all modern websites are using CSS and Javascript based framework to come up with responsive and dynamic websites but you can create a good layout using simple HTML tables or division tags in combination with other formatting tags. This chapter will give you few s on how to create a simple but working layout for your webpage using pure HTML and its attributes.

**HTML Layouts - Using DIV, SPAN**

The <div> element is a block level element used for grouping HTML elements. While the <div> tag is a block-level element, the HTML <span> element is used for grouping elements at an inline level.

Although we can achieve pretty nice layouts with HTML tables, but tables weren't really designed as a layout tool. Tables are more suited to presenting tabular data.

<!DOCTYPE html>

<html>

<head>

<title>HTML Layouts using DIV, SPAN</title>

</head>

<body>

<div style="width:100%">

<div style="background-color:#b5dcb3; width:100%">

<h1>This is Web Page Main title</h1>

</div>

<div style="background-color:#aaa; height:200px;width:100px;float:left;">

<div><b>Main Menu</b></div>

HTML<br />

PHP<br />

PERL...

</div>

<div style="background-color:#eee; height:200px;width:350px;float:left;">

<p>Technical and Managerial Tutorials</p>

</div>

<div style="background-color:#aaa; height:200px;width:100px;float:right;">

<div><b>Right Menu</b></div>

HTML<br />

PHP<br />

PERL...

</div>

<div style="background-color:#b5dcb3;clear:both">

<center>

Copyright © 2007 Elearninfotech.com

</center>

</div>

</div>

</body>

</html>

This will produce following result:

This is Web Page Main title

**Main Menu**

HTML  
PHP  
PERL...

Technical and Managerial Tutorials

**Right Menu**

HTML  
PHP  
PERL...

Copyright © 2007 Elearninfotech.com