# HTML <meta> Tag

<head>  
  <meta charset="UTF-8">  
  <meta name="description" content="Free Web tutorials">  
  <meta name="keywords" content="HTML, CSS, JavaScript">  
  <meta name="author" content="John Doe">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
</head>

The <meta> tag defines metadata about an HTML document. Metadata is data (information) about data.

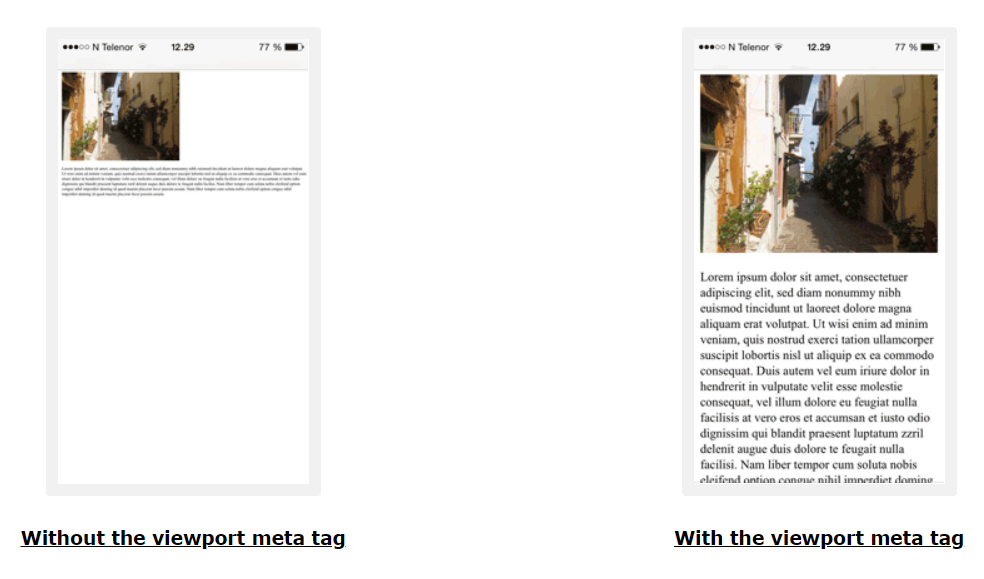
<meta> tags always go inside the <head> element, and are typically used to specify character set, page description, keywords, author of the document, and viewport settings.

Metadata will not be displayed on the page, but is machine parsable.

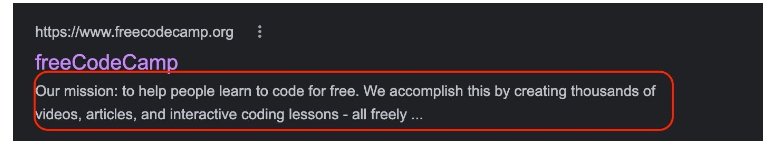
Metadata is used by browsers (how to display content or reload page), search engines (keywords), and other web services.

There is a method to let web designers take control over the viewport (the user's visible area of a web page), through the <meta> tag (See "Setting The Viewport" example below).





<meta name="description" content="Our mission: to help people learn to code for free. We accomplish this by creating thousands of videos, articles, and interactive coding lessons - all freely available to the public.">



# HTML Attributes

Attributes are another important part of HTML markup. An attribute is used to define the characteristics of an 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 <font> element in the example carries an attribute whose name is *face*, which you can use to indicate which typeface you want the text to appear in.
* The *value* is what you want the value of the property to be. The first example was supposed to use the Arial typeface, so the value of the *face* attribute is Arial.

The value of the attribute should be put in double quotation marks, and is separated from the name by the equals sign. You can see that a color for the text has been specified as well as the typeface in this <font> element:

|  |
| --- |
| <font face="arial" color="#CC0000"> |

Many HTML tags have a unique set of their own attributes. These will be discussed as each tag is introduced throughout the tutorial. Right now we want to focus on a set of generic attributes that can be used with just about every HTML Tag in existence.

# 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 can be used to uniquely identify any element within a page ( or style sheet ). 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.

We will discuss style sheet in separate tutorial. For now, the id attribute could be used to distinguish between two paragraph elements, like so:

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

Note that there are some special rules for the value of the id attribute, it must:

* Begin with a letter (A.Z or a.z) and can then be followed by any number of letters, digits (0.9), hyphens, underscores, colons, and periods.
* Remain unique within that document; no two attributes may have the same value within that HTML document.

## 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 or while the element is loading.

For example:

|  |
| --- |
| <h4 title="Hello HTML!">Titled Heading Tag Example</h4> |

Above code will generate following result:



Now try to bring your cursor over "Titled Heading Tag Example" and see the result.

## The class Attribute:

The *class* attribute is used to associate an element with a style sheet, and specifies the class of element. You learn more about the use of the class attribute when you will learn Casecading 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 CSS rules within the element. For example:

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

# Internationalization Attributes:

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

* dir
* lang
* xml:lang

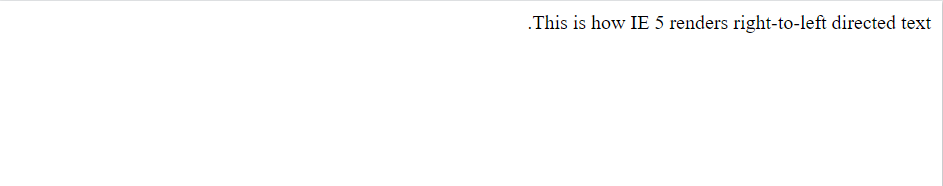
## The dir Attribute:

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 (for languages such as Hebrew or Arabic that are read right to left) |

Example:

|  |
| --- |
| <html dir=rtl>  <head>  <title>Display Directions</title>  </head>  <body>  This is how IE 5 renders right-to-left directed text.  </body>  </html> |



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. This attribute has been replaced by the xml:lang attribute in new XHTML documents.

When **included within the <html> tag**, the ***lang* attribute specifies the language you've generally used within the document**. When used within other tags, the lang attribute specifies the language you used within that tag's content. Ideally, the browser will use *lang* to better render the text for the user.

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

Example:

|  |
| --- |
| <html lang=en>  <head>  <title>English Language Page</title>  </head>  <body>  This page is using English Language  </body>  </html> |

## The xml:lang Attribute:

The *xml:lang* attribute is the XHTML replacement for the *lang* attribute. The value of the *xml:lang* attribute should be an ISO-639 country code as mentioned in previous section.

# Generic Attributes:

Here's a table of some other attributes that are readily usable with many of HTML's 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 an 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 for your elements. |

We will see related examples as we will proceed to study other HTML tags.

# HTML Formatting Tags

If you want people to read what you have written, then structuring your text well is even more important on the Web than when writing for print. People have trouble reading wide, long, paragraphs of text on Web sites unless they are broken up well.

This section will teach you basic text formatting elements like heading elements and paragraph elements.

# Whitespace and Flow:

Before you start to mark up your text, it is best to understand what HTML does when it comes across spaces and how browsers treat long sentences and paragraphs of text.

You might think that if you put several consecutive spaces between two words, the spaces would appear between those words onscreen, but this is not the case; by default, only one space will be displayed. This is known as white *space collapsing*. So you need to use special HTML tags to create multiple spaces.

Similarly, if you start a new line in your source document, or you have consecutive empty lines, these will be ignored and simply treated as one space. So you need to use special HTML tags to create more number of empty lines.

# Create Headings - The <hn> Elements:

Any documents 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> |

This will display following result:

# Create Paragraph - The <p> Element:

The <p> element offers a way to structure your text. Each paragraph of text should go in between an opening <p> and closing </p> tag as shown below in the example:

|  |
| --- |
| <p>Here is a paragraph of text.</p>  <p>Here is a second paragraph of text.</p>  <p>Here is a third paragraph of text.</p> |

This will produce following result:

|  |
| --- |
|  |

You can use *align* attribute to align your paragraphs.

|  |
| --- |
| <p align="left">This is left aligned.</p> <p align="center">This is center aligned.</p> <p align="right">This is right aligned.</p> <p align="justify">This is jutified. This works when you have multiple lines in your paragraph and you want to justfy all the lines so that they can look more nice.</p> |

This will produce following result:

|  |
| --- |
|  |

# Create Line Breaks - The <br /> Element:

Whenever you use the <br /> element, anything following it starts on the next line. This tag is an example of an **empty** element, where you do not need opening and closing tags, as there is nothing to go in between them.

**Note:** The <br /> element has a space between the characters br and the forward slash. If you omit this space, older browsers will have trouble rendering the line break, while if you miss the forward slash character and just use <br> it is not valid XHTML

Example:

|  |
| --- |
| Hello<br />  You come most carefully upon your hour.<br />  Thanks<br />  Mahnaz |

This will produce following result:

|  |
| --- |
|  |

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_br_tag)

# Centring Content - The <center> Element:

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

Example:

|  |
| --- |
| <p>This is not in the center.</p>  <center>  <p>This is in the center.</p>  </center> |

This will produce following result:

|  |
| --- |
|  |

# Nonbreaking Spaces:

Since the browser will display only one blank space even if you put millions in your code, HTML has the &nbsp; character entity. It makes it possible to display multiple blank spaces.

|  |
| --- |
| A good example of this technique appears in the movie "12 Angry Men." |

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

# HTML provides the &ensp; character entity for 2 non-breaking spaces, and &emsp; for 4 non-breaking spaces.

# Preserve Formatting - The <pre> Element:

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.

|  |
| --- |
| <pre>  function testFunction( strText ){  alert (strText)  }  </pre> |

This will produce following result:

|  |
| --- |
| function testFunction( strText ){  alert (strText)  } |

# Horizontal Rules - The <hr /> Element

Horizontal rules 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.

For example you may want to give a line between two paragraphs as follows:

|  |
| --- |
| <p>This is paragraph one and should be on top</p>  <hr />  <p>This is paragraph two and should be at bottom</p> |

This will produce following result:

|  |
| --- |
|  |

Again <hr /> tag is an example of an empty element, where you do not need opening and closing tags, as there is nothing to go in between them.

**Note:** The <hr /> element has a space between the characters br and the forward slash. If you omit this space, older browsers will have trouble rendering the line break, while if you miss the forward slash character and just use <hr> it is not valid XHTML

## Presentational Tags:

If you use a word processor, you are 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 - The <b> Element:

Anything that appears in a <b>...</b> element is displayed in bold, like the word bold here:

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

This will produce following result:

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

# Italic Text - The <i> Element:

Anything that appears in a <i>...</i> element is displayed in italicized, like the word italicized here:

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

This will produce following result:

|  |
| --- |
| The following word uses a *italicized* typeface. |

# Underlined Text - The <u> Element:

Anything that appears in a <u>...</u> element is displayed with underline, like the word underlined here:

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

This will produce following result:

|  |
| --- |
| The following word uses a underlined typeface. |

# Strike Text - The <strike> Element:

Anything that appears in a <strike>...</strike> element is displayed with strikethrough, which is a thin line through the text:

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

This will produce following result:

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

# Monospaced font - The <tt> Element:

The content of a <tt> element is written in monospaced font. Most fonts are known as variable-width fonts because different letters are of different widths (for example, the letter m is wider than the letter i). In a monospaced font, however, each letter is the same width.

|  |
| --- |
| <p>The following word uses a <tt>monospaced</tt> typeface.</p> |

This will produce following result:

|  |
| --- |
| The following word uses a monospaced typeface. |

# Superscript Text - The <sup> Element:

The content of a <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.

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

This will produce following result:

|  |
| --- |
| The following word uses a superscript typeface. |

# Subscript Text - The <sub> Element:

The content of a <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.

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

This will produce following result:

|  |
| --- |
| The following word uses a subscript typeface. |

# Larger Text - The <big> Element:

The content of the <big> element is displayed one font size larger than the rest of the text surrounding it.

|  |
| --- |
|  |
| . |

# Smaller Text - The <small> Element:

The content of the <small> element is displayed one font size smaller than the rest of the text surrounding it.

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

This will produce following result:

|  |
| --- |
|  |

# Grouping - The <div> and <span> Elements :

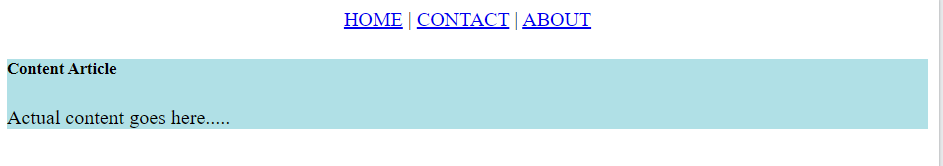
The <div> and <span> elements allow you to group together several elements to create sections or subsections of a page.

For example, you might want to put all of the footnotes on a page within a <div> element to indicate that all of the elements within that <div> element relate to the footnotes. You might then attach a style to this <div> element so that they appear using a special set of style rules.

The <div> element is used to group block-level elements together:

|  |
| --- |
| <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" style="background-color:powderblue;”>  <h5>Content Articles</h5>  <p>Actual content goes here.....</p>  </div> |

This will produce following result:



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

|  |
| --- |
| <div><p>This is the example of <span style="color:green">span tag</span> and the <span style="color:purple">div tag</span> alongwith CSS</p></div> |

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 to allow you to attach a style to a section of a page.

# HTML Phrase Tags

While some of these phrase elements are displayed in a similar manner to the <b>, <i>, <pre>, and <tt> elements you have already seen, they are designed for specific purposes. For example, the <em> and <strong> elements give text emphasis and strong emphasis respectively and there are several elements for marking up quotes.

We will see all phrase tags in this section with examples.

# Emphasized Text - The <em> Element:

The content of an <em> element is intended to be a point of emphasis in your document, and it is usually displayed in italicized text. The kind of emphasis intended is on words such as "must" in the following sentence:

|  |
| --- |
| <p>You <em>must</em> remember to close elements in XHTML.</p> |

This will produce following result:

|  |
| --- |
| You must remember to close elements in XHTML. |

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_em_tag)

# Strong Text - The <strong> Element:

The <strong> element is intended to show strong emphasis for its content; stronger emphasis than the <em> element. As with the <em> element, the <strong> element should be used only when you want to add strong emphasis to part of a document.

|  |
| --- |
| <p>You <strong>must</strong> remember to close elements in XHTML.</p> |

This will produce following result:

|  |
| --- |
| You **must** remember to close elements in XHTML. |

# Text Abbreviation - The <abbr> Element :

You can indicate when you are using an abbreviated form by placing the abbreviation between opening <abbr> and closing </abbr> tags.

|  |
| --- |
| <p>I have a friend called <abbr title="Abhishek">Abhi</abbr>.</p> |

This will produce following result:

|  |
| --- |
|  |

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_abbr_tag)

# Using Acronym - The <acronym> Element :

The <acronym> element allows you to indicate that the text between an opening <acronym> and closing </acronym> element is an acronym.

When possible use a title attribute whose value is the full version of the acronyms on the <acronym> element, and if the acronym is in a different language, include an xml:lang attribute in XHTML documents.

|  |
| --- |
| <p>This chapter covers marking up text in <acronym title="Extensible Hypertext Markup Language">XHTML</acronym>.</p> |

This will produce following result:

|  |
| --- |
| This chapter covers marking up text in XHTML. |

At present, the major browsers do not change the appearance of the content of the <acronym> element.

# Special Terms - The <dfn> Element :

The <dfn> element allows you to specify that you are introducing a special term. Its use is similar to the words that are in italics in the midst of paragraphs in this book when new key concepts are introduced.

Typically, you would use the <dfn> element the first time you introduce a key term and only in that instance. Most recent browsers render the content of a <dfn> element in an italic font.

|  |
| --- |
| <p>This tutorial teaches you how mark up your documents for the web using <dfn>XHTML</dfn>.</p> |

This will produce following result:

|  |
| --- |
| This tutorial teaches you how mark up your documents for the web using XHTML. |

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_dfn_tag)

# Quoting Text - The <blockquote> Element :

When you want to quote a passage from another source, you should use the <blockquote> element.

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

|  |
| --- |
| <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> |

This will produce following result:

|  |
| --- |
|  |

You can use the *cite* attribute on the <blockquote> element to indicate the source of the quote.

|  |
| --- |
| <p>The following description of XHTML is taken from the W3C Web site:</p>  <blockquote cite="http://www.w3.org/markup/"> 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> |

# Short Quotations - The <q> Element :

The <q> element is intended to be used when you want to add a quote within a sentence rather than as an indented block on its own.

|  |
| --- |
| <p>Amit is in Spain, <q>It is a beautiful place </q>.</p> |

This will produce following result:

|  |
| --- |
|  |

The <q> element can also carry the cite attribute. The value should be a URL pointing to the source of the quote.

# Citations - The <cite> Element :

If you are quoting a text, you can indicate the source placing it between an opening <cite> tag and closing </cite> tag

As you would expect in a print publication, the content of the <cite> element is rendered in italicized text by default.

|  |
| --- |
| <p>This HTML Tutorial is derived from <cite>World Wide Web Standard for HTML</cite>.</p> |

This will produce following result:

|  |
| --- |
| This HTML Tutorial is derived from World Wide Web Standard for HTML. |

# Computer Code - The <code> Element :

Any code to appear on a Web page should be placed inside a <code> element. Usually the content of the <code> element is presented in a monospaced font, just like the code in most programming books.

|  |
| --- |
| <h1> <code>This is inside code element</code></h1> |

This will produce following result:

|  |
| --- |
|  |

# Keyboard Text - The <kbd> Element :

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

The content of a <kbd> element is usually represented in a monospaced font rather like the content of the <code> element.

|  |
| --- |
| <h1> <kbd>This is inside kbd element</kbd></h1> |

This will produce following result:

|  |
| --- |
| This is inside kbd element |

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_kbd_tag)

# Programming Variables - The <var> Element :

This element is usually used in conjunction with the <pre> and <code> elements to indicate that the content of that element is a variable that can be supplied by a user.

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

This will produce following result:

|  |
| --- |
| document.write("user-name") |

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_var_tag)

# Program Output - The <samp> Element :

The <samp> element indicates sample output from a program, script, or the like. Again, it is mainly used when documenting programming concepts. For example:

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

This will produce following result:

|  |
| --- |
| Result produced by the program is Hello World |

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_samp_tag)

# Addresses - The <address> Element :

The <address> element is used to contain any address. For example:

|  |
| --- |
| <address>304, Menna Colony, Hyderabad - INDIA, 500032</address> |

This will produce following result:

|  |
| --- |
| 304, Menna Colony, Hyderabad - INDIA, 500032 |

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_address_tag)

# Block and Inline Elements:

We can categories all the elements into two sections:

* **Block-level elements** - Block-level elements appear on the screen as if they have a carriage return or 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.

The elements which we have not discussed till now, will be discussed in subsequent chapters.

# HTML Comments

Comments are piece of code which is ignored by any web browser. It is good practice to comment your 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.

HTML Comment lines are indicated by the special beginning tag <!-- and ending tag --> placed at the beginning and end of EVERY line to be treated as a comment.

Comments do not nest, and 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.

For example: Given line is a valid comment in HTML

|  |
| --- |
| <!-- This is commented out --> |

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

|  |
| --- |
| < !-- This is commented out --> |

Be careful if you use comments to "comment out" HTML that would otherwise be shown to the user, since some older browsers will still pay attention to angle brackets inside the comment and close the comment prematurely -- so that some of the text that was supposed to be inside the comment mistakenly appears as part of the document.

# Multiline Comments:

You have seen how to comment a single line in HTML. You can comment multiple lines by the special beginning tag <!-- and ending tag --> placed before the first line and end of the lastline to be treated as a comment.

For example:

|  |
| --- |
| <!--  This is a multiline comment <br />  and can span through as many as lines you like.  --> |

# Conditional Comments :

Conditional comments only work in Explorer on Windows, and are thus excellently suited to give special instructions meant only for Explorer on Windows. They are supported from Explorer 5 onwards, and it is even possible to distinguish between 5.0, 5.5 and 6.0.

Conditional comments work as follows:

|  |
| --- |
| <!--[if IE 6]>  Special instructions for IE 6 here  <![endif]--> |

* Their basic structure is the same as an HTML comment (<!-- -->). Therefore all other browsers will see them as normal comments and will ignore them entirely.
* Explorer Windows, though, has been programmed to recognize the special <!--[if IE]> syntax, resolves the if and parses the content of the conditional comment as if it were normal page content.
* Since conditional comments use the HTML comment structure, they can only be included in HTML files, and not in CSS files.

# Using Comment tag

There are few browsers who supports <comment> tag to comment a part of code.

|  |
| --- |
| <p>This is <comment>not</comment> Internet Explorer.</p> |

# Commenting Scripts and Style Sheets:

If you are using Java Script or VB Script in your HTML code then it is recommended to put that script code inside proper HTML Comments to make old browser works properly.

For example:

|  |
| --- |
| <script>  <!--  document.write("Hello World!")  //-->  </script> |

Similarly if you are using Casecading Style Sheet in your HTML code then it is recommended to put that style sheet code inside proper HTML Comments to make old browser works properly.

For example:

|  |
| --- |
| <style>  <!--  img{  border:0px;  }  //-->  </style> |

**NOTE:** To become familiar with JAVA Script and Cascading Style Sheet you need to refer different tutorial.

# HTML Fonts

Font face and color depends entirely on the computer and browser that is being used to view your page. But the <font> tag is used to add style, size, and color to the text on your site. 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 page, simply use the <font> tag. The text that follows will remain changed until you close with the </font> tag. You can change any or all of the font attributes at the one time, by including all the required changes within the one <font> tag.

**NOTE: The font and basefont tags are deprecated and it is supposed to be removed in a future version of HTML. So it should not be used. Its is suggested to use css styles to manipulate your font.**

# Font Size:

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

Example:

|  |
| --- |
| <font size="1">Font size="1"</font>  <font size="2">Font size="2"</font>  <font size="3">Font size="3"</font>  <font size="4">Font size="4"</font>  <font size="5">Font size="5"</font>  <font size="6">Font size="6"</font>  <font size="7">Font size="7"</font> |

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" |

**SPECIFY THE RELATIVE FONT SIZE. <font size="+n"> or <font size="-n">:**  
You can specify how many sizes larger or how many sizes smaller than the preset font size should be.

Example:

|  |
| --- |
| <font size="-1">Font size="-1"</font>  <font size="+1">Font size="+1"</font>  <font size="+2">Font size="+2"</font>  <font size="+3">Font size="+3"</font>  <font size="+4">Font size="+4"</font> |

This will produce following result:

|  |
| --- |
| Font size="-1" Font size="+1" Font size="+2" Font size="+3" Font size="+4" |

# Font Face:

You can set any font you like 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 they will default to Times New Roman of your font with size attribute. See below few examples on using different font face

Example:

|  |
| --- |
| <font face="Times New Roman" size="5">Times New Roman</font>  <font face="Verdana" size="5">Verdana</font>  <font face="Comic sans MS" size="5">Comic Sans MS</font>  <font face="WildWest" size="5">WildWest</font>  <font face="Bedrock" size="5">Bedrock</font> |

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.

Example:

|  |
| --- |
| <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 that it has available. If none of your selections are installed....then it will display the default font face *Times New Roman*.

Check a complete list of [HTML Standard Fonts](http://www.tutorialspoint.com/html/html_fonts_ref.htm).

# 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. Check a complete list of [HTML Color Name with Codes](http://www.tutorialspoint.com/html/html_color_names.htm).

Example:

|  |
| --- |
| <font color="#FF00FF">This text is hexcolor #FF00FF</font>  <font color="red">This text is red</font> |

This will produce following result:

|  |
| --- |
| This text is hexcolor #FF00FF This text is red |

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_font_tag)

# 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> element. You can then use the <font> elements to override the <basefont> settings.

The attributes that the <basefont> element takes are exactly the same as for the <font> element. You can also set the size of fonts relative to the size of the <basefont> by giving them a value of +1 for a size larger or -2 for two sizes smaller

**NOTE:** This element is deprecated in HTML 4 and will be removed from HTML, the preferred option is to use CSS styles. Your browser may not have support for this tag.

# Example:

|  |
| --- |
| <basefont face="arial, verdana, sans-serif" size="2" color="#ff0000">  <p>This is the page's default font.</p>  <h2>Example of the &lt;basefont&gt; Element</h2>  <p><font size="+2" color="darkgray">Here is some darkgray text  two sizes larger</font></p>  <p><font face="courier" size="-1" color="#000000">Here is a courier  font, a size smaller, in black</font></p> |

This will produce following result:

|  |
| --- |
| This is the page's default font. Example of the <basefont> Element Here is some darkgray text two sizes larger  Here is a courier font, a size smaller, in black |

As you can see, the default font now takes on the properties specified in the <basefont> element. It is red, size 2, and uses the Arial typeface.

The paragraph after the <h2> element uses a font size two sizes larger than the default size and is gray text, whereas the following paragraph uses a font one size smaller than the default font. You can also see that the color of this font is black (overriding the default).

# HTML Marquees

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

**NOTE:** The HTML <marquee> is an MSIE extension, but is now supported by NS 7 also. So please check if your browser supports this tag or not.

# Syntax:

A simple syntax to use marquee is as follows:

|  |
| --- |
| <marquee attribute\_name="attribute\_value"....more attributes>  One or more lines or text message or image  </marquee> |

# Attributes:

A HTML marquee can have following attributes:

* **width:** how wide the marquee is. This will have a value like 10 or 20%etc.
* **height:** how tall the marquee is. This will have a value like 10 or 20% etc.
* **direction:** which direction the marquee should scroll. This will have value either *up*, *down*, *left* or *right*.
* **behavior:** what type of scrolling. This will have value *scroll*, *slid* and *alternate*.
* **scrolldelay:** how long to delay between each jump. This will have a value like 10 etc.
* **scrollamount:** how far to jump. This will have a value like 10 etc.
* **loop:** how many times to loop. The default value is INFINITE, which means that the marquee loops endlessly.
* **bgcolor:** background color. This will have any color name or color hex value.
* **hspace:** horizontal space around the marquee. This will have a value like 10 or 20%etc.
* **vspace:** vertical space around the marquee. This will have a value like 10 or 20%etc.

# Examples:

Here are few examples to demonstrate the usage of marquee tag.

|  |
| --- |
| <marquee>This is basic example of marquee</marquee> |

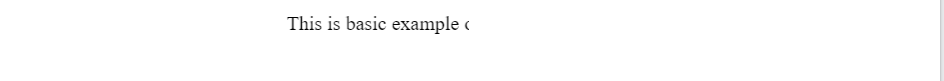
|  |
| --- |
| <marquee width="50%">This example will take only 50% width</marquee> |

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

|  |
| --- |
| <marquee direction="up">This text will scroll from bottom to up</marquee> |

This will produce following result:

# 



# HTML Images

Images are very important to beautify as well as to depicts many concepts on your web page. Its is true that one single image is worth than thuasands of words. So as a Web Developer you should have clear understanding on how to use images in your web pages.

# Insert Image - The <img> Element:

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

|  |
| --- |
| <img src="image URL" attr\_name="attr\_value"...more attributes /> |

# 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.
* **id:** id of the image with in the document.
* **style:** this will be used if you are using CSS.
* **title:** specifies a text title. The browser, perhaps flashing the title when the mouse passes over the link.
* **ismap and usemap:** These attributes for the <img> tag tell the browser that the image is a special mouse-selectable visual map of one or more hyperlinks, commonly known as an **image map**. We will see how to use these attributes in [Image Links](http://www.tutorialspoint.com/html/html_image_links.htm) chapter.

## A Simple Example:

|  |
| --- |
| <img src="http://www.tutorialspoint.com/images/html.gif" alt="HTML Tutorial" /> |

**Image Attributes - width, height, title, border and align:**

Now let us try to set some more attributes:

|  |
| --- |
| <img src="http://www.tutorialspoint.com/images/html.gif" alt="HTML Tutorial" width="100" height="100"  border="2" align="right" title="HTML Tutorial" /> |

# HTML Text Links

[previous](http://www.tutorialspoint.com/html/html_images.htm)[next](http://www.tutorialspoint.com/html/html_image_links.htm)[AddThis Social Bookmark Button](http://www.addthis.com/bookmark.php)

Web pages can contain 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, phrases, and images. Thus you can create hyperlinks using text or images available on your any web page.

In this tutorial you will learn how to create text links between the different pages of your site, links within pages of your sites, and how to link to other sites ( or external sites). If you want to know more about URL then check [Understanding URL Tutorial](http://www.tutorialspoint.com/html/understanding_url_tutorial.htm).

# Linking Documents - The <a> Element:

A link is specified using the <a> element. This element is called **anchor tag** as well. 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 this tag.

|  |
| --- |
| <a href="Document URL" attr\_name="attr\_value"...more attributes /> |

# Anchor Attributes:

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

* **href:** specifies the URL of the target of a hyperlink. Its value is any valid document URL, absolute or relative, including a fragment identifier or a JavaScript code fragment.
* **target:** specify where to display the contents of a selected hyperlink. If set to "\_blank" then a new window will be opened to display the loaded page, if set to "\_top" or "\_parent" then same window will be used to display the loaded document, if set to "\_self" then loads the new page in current window. By default its "\_self".
* **name & id:** attributes places a label within a document. When that label is used in a link to that document, it is the equivalent of telling the browser to goto that label.
* **event:** attributes like *onClick*, *onMouseOver* etc. are used to trigger any Javascript ot VBscript code.
* **title:** attribute lets you specify a title for the document to which you are linking. The value of the attribute is any string, enclosed in quotation marks. The browser might use it when displaying the link, perhaps flashing the title when the mouse passes over the link.
* **accesskey:** attribute attribute provides a keyboard shortcut that can be used to activate a link. For example, you could make the T key an access key so that when the user presses either the Alt or Ctrl key on his keyboard (depending on his operating system) along with the T key, the link gets activated.

## A Simple Example:

|  |
| --- |
| <a href="http://www.tutorialspoint.com/" target="\_blank" >TP Home</a> |  <a href="http://www.amrood.com/" target="\_self" >AMROOD Home</a> | <a href="http://www.change-images.com/" target="\_top" >Change Images Home</a> |

This will produce following result, Click and come back to proceed with rest of the tutorial:

|  |
| --- |
| [Tutorials Point](http://www.tutorialspoint.com/) | [AMROOD](http://www.amrood.com/) | [Change Images](http://www.change-images.com/) |

# Base Path for Links:

It is not required to give a complete URL for every link. You can get rid of it if you will use <base> tag in your 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.

For example we have used following base tag in all the pages at tutorialspoint.com:

|  |
| --- |
| <head>  <base href="http://www.tutorialspoint.com/">  </head> |

So now if you will use **<a href="/html/index.htm"** then it will be considered as **<a href="http://www.tutorialspoint.com/html/index.htm"**.

# Linking to a Page Section:

You can create a link to a particular section of a page by using *name* attribute. Here we will create three links with-in this page itself.

First create a link to reach to the top of this page. Here is the code we have used for the title heading *HTML Text Links*

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

Now you have a place where you can reach. To reach to this place use the following code with-in this document anywhere:

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

This will produce following link and you try using this link to reach to the top of this page:

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

**NOTE:** Here we are using relative path. You can give complete URL and then # and then link name eg. http://www.tutorialspoint.com/html/html\_text\_links.htm#top

You can use this type of URL in any other page to reach directly to a particular section.

# Setting Link Colors:

You can set colors of your links, active links and visited links using *link*, *alink* and *vlink* attributes of <body> tag. But it is recommended to use CSS to set colors of links, visited links and active links.

Following is the example we have used for our web side tutorialspoint.com

|  |
| --- |
| a:link {color:#900B09; background-color:transparent}  a:visited {color:#900B09; background-color:transparent}  a:active {color:#FF0000; background-color:transparent}  a:hover {color:#FF0000; background-color:transparent} |

You can refer to Style Sheet Tutorial for a complete understanding on CSS.

Otherwise you can use <body> tag to set link colors. Here is the syntax.

|  |
| --- |
| <body alink="#FF0000" link="#900B09" vlink="#900B09">  .......  </body> |

# Create 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:

|  |
| --- |
| <a href="http://www.example.com/file.pdf">Download File</a> |

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

|  |
| --- |
| [Download File](http://www.example.com/file.pdf) |

You can not make an image download able until you follow the following procedure.

## How To Raise a "File Download" Dialog Box ?

Sometime it is desired that you want to give option where a use will click a link and it will pop up a "File Download" box to the user in stead of displaying actual content. This is very easy and will be achived through HTTP header.

This HTTP header will be different from the header mentioned in previous section.

For example,if you want make a **FileName** file downloadable from a given link then its syntax will be as follows.

|  |
| --- |
| #!/usr/bin/perl  # HTTP Header  print "**Content-Type:**application/octet-stream; name=\"FileName\"\r\n";  print "**Content-Disposition:** attachment; filename=\"FileName\"\r\n\n";  # Actual File Content will go hear.  open( FILE, "<FileName" );  while(read(FILE, $buffer, 100) )  {  print("$buffer");  } |

# HTML Image Links

[previous](http://www.tutorialspoint.com/html/html_text_links.htm)[next](http://www.tutorialspoint.com/html/html_email_links.htm)[AddThis Social Bookmark Button](http://www.addthis.com/bookmark.php)

Previous chapters has tought you how to create hyper text link using text and how to use images in your web page. Now we will learn how to use images to create hyper links. See example below:

|  |
| --- |
| <a href="http://www.tutorialspoint.com/index.htm" target="\_self" >  <img src="/images/home.gif" alt="Tutorials Point Home" border="0"/>  </a> |

This will create following hyperlink at tutorialspoint.com home.

|  |
| --- |
| [Tutorials Point Home](http://www.tutorialspoint.com/index.htm) |

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

# Mouse-Sensitive Images:

The HTML and XHTML standards provide a feature that lets you embed many different links inside the same image. Clicking different areas of the image causes the browser to link to different target documents. Such mouse-sensitive images known as *image maps*.

There are two ways to create image maps:

* **A server-side image maps:** is enabled by the *ismap* attribute for the <img> tag and requires access to a server and related image-map processing applications.
* **A client-side image maps:** is created with the *usemap* attribute for the <img> tag, along with corresponding <map> and <area> tags.

## Server-Side Image Maps:

You add an image to an anchor simply by placing an <img> tag within the body of the <a> tag. Make that embedded image into a mouse-sensitive one by adding the *ismap* attribute to the <img> tag. This special <img> attribute tells the browser that the image is a special map containing more than one link.

When the user clicks some place within the image, the browser passes the coordinates of the mouse pointer along with the URL specified in the <a> tag to the document server. The server uses the mouse-pointer coordinates to determine which document to deliver back to the browser.

When *ismap* is used, the href attribute of the containing <a> tag must contain the URL of a server application like amap file or cgi script etc. to process the incoming request based on the passed coordinates.

The coordinates of the mouse position are screen pixels counted from the upper-left corner of the image, beginning with (0,0). The coordinates, preceded by a question mark, are added to the end of the URL.

For example, if a user clicks 50 pixels over and 30 pixels down from the upper-left corner of the image displayed from the following link:

|  |
| --- |
| <a href="/cgi-bin/logo.map" target="\_self" >  <img ismap src="/images/html.gif"  alt="HTML" border="0"/>  </a> |

Then the browser sends the following search parameters to the HTTP server which can be processed by cgi script or map file and you can link whatever you like to these coordinates:

|  |
| --- |
| /cgi-bin/logo.map?50,30 |

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_ismap_attr)

**NOTE:** Converting the coordinates into a specific document is handled by the server side application, either cgi programme or special map files provided by seb server. For more detail Check [Using HTML ismap](http://www.tutorialspoint.com/html/using_html_ismap.htm).

# Client-Side Image Maps:

Client side image maps are enabled by the *usemap* attribute for the <img /> tag and defined by special <map> and <area> extension tags.

The image that is going to form the map is inserted into the page using the <img /> element as normal, except it carries an extra attribute called usemap. The value of the usemap attribute is the value of the name attribute on the <map> element, which you are about to meet, preceded by a pound or hash sign.

The <map> element actually creates the map for the image and usually follows directly after the <img /> element. It acts as a container for the <area /> elements that actually define the clickable hotspots. The <map> element carries only one attribute, the name attribute, which is the name that identifies the map. This is how the <img /> element knows which <map> element to use.

The <area> element specifies the shape and the coordinates that define the boundaries of each clickable hotspot. Here's an example from the image map:

|  |
| --- |
| <img src=/images/html.gif alt="HTML Map" border="0" usemap="#html"/>  <!-- Create Mappings -->  <map name="html">  <area shape="circle"  coords="154,150,59" href="link1.htm" alt="link 1"  target="\_self" />  <area shape="poly"  coords="272,79,351,79,351,15,486,15,486,218,272,218,  292,166,292,136,270,76" alt="link 2"  href="link2.htm" target="\_self" />  <area shape="rect"  coords="325,224,488,286" alt="link 3"  href="link3.htm" target="\_self" />  </map> |

The actual value of coords is totally dependent on the shape in question. Here is a summary, to be followed by detailed examples:

**rect = x1 , y1 , x2 , y2**

x1 and y1 are the coordinates of the upper left corner of the rectangle; x2 and y2 are the coordinates of the lower right corner. Therefore, a rectangle which goes from 10,5 to 20,25 would have the attribute *coords="10,5,20,25"*. A rectangle which defines the upper-left quarter of an image might use *coords="0,0,50%,50%"*.

**circle = xc , yc , radius**

xc and yc are the coordinates of the center of the circle, and radius is the circle's radius. A circle centered at 200,50 with a radius of 25 would have the attribute *coords="200,50,25"*; one centered at the image's center and having a diameter of half the image would be defined by *coords="50%,50%,25%"*.

**poly = x1 , y1 , x2 , y2 , x3 , y3 , ... xn , yn**

The various x-y pairs define vertices (points) of the polygon, with a "line" being drawn from one point to the next point. A diamond-shaped polygon with its top point at 20,20 and 40 pixels across at its widest points would have the attribute *coords="20,20,40,40,20,60,0,40"*. A "line" is always drawn from the coordinates of the last point to the coordinates of the first point in order to close the polygon.

All coordinates are relative to the upper-left corner of the image (0,0). Each shape has a related URL.You can use any image software to know the coordinates of different positions.

**NOTE:** Following image crop utility can help you to identify image cordinates online [Image Crop Utility](http://www.change-images.com/crop-images.htm). Just upload your image and click the area to identify cordinates of that area.

# HTML Email Links

This is very easy to put an HTML email link on your page. But while doing so, you need to put your email address on your web page which can cause a spamming problem for your email account. There are many guys over the internet who can run programs to harvest these types of emails for spamming. So if you are going to put your email link on a public website then you have be prepared for anti-spamming as well.

You can have another option to facilitate people to send you emails. This option is to use HTML forms to take user data and then use CGI script to send an email.

A simple example, check our [Contact Us](http://www.tutorialspoint.com/about/contact_us.htm) Form. We take user feedback using this form and then we are using one CGI program which is collecting this information and sending email to one given email ID.

You will learn about HTML Forms in [HTML Forms](http://www.tutorialspoint.com/html/html_forms.htm) and you can learn about CGI in our another tutorial [PERL & CGI](http://www.tutorialspoint.com/perl/perl_cgi.htm).

# HTML Email Tag:

HTML <a> tag provides you facility to specifiy an email address to send an email. While using <a> tag as an email tag then 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:

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

Now if a user will click this link then it will lanuch one Email Client ( like Lotus Notes, Outlook Express etc. ) installed on your user's computer.

This is another risk to use this procedure because if users do not have email client installed on their computer then it would not be possible to send you email.

## Default Email Subject and Body:

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:

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

# HTML Tables

Tables are very useful to arrange in HTML and they are used very frequently by almost all web developers. Tables are just like spreadsheets and they are made up of rows and columns.

You will create a table in HTML/XHTML by using <table> tag. Inside <table> element the table is written out row by row. A row is contained inside a <tr> tag . which stands for table row. And each cell is then written inside the row element using a <td> tag . which stands for table data.

## Example:

|  |
| --- |
| <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> |

This will produce following result:

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

**NOTE:** In the above example *border* is an attribute of <table> and it will put border across all the cells. If you do not need a border then you cal use *border="0"*. The border attribute and other attributes also mentione din this session are deprecated and they have been replaced by CSS. So it is recommended to use CSS instead of using any attribute directly.

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_table_tag)

# Table Heading - The <th> Element:

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

|  |
| --- |
| <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> |

This will produce following result. You can see its making heading as a bold one:

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

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_th_attr)

**NOTE:** Each cell must, however, have either a <td> or a <th> element in order for the table to display correctly even if that element is empty.

# Table Cellpadding and Cellspacing:

There are two attribiutes called *cellpadding* and *cellspacing* which you will use to adjust the white space in your table cell. Cellspacing defines the width of the border, while cellpadding represents the distance between cell borders and the content within. Following is the example:

|  |
| --- |
| <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> |

This will produce following result:

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

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_tablepadding_attr)

# 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. Following is the example:

|  |
| --- |
| <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> |

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 | | | |

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_tablespan_attr)

# Tables Backgrounds

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

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

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

Here is an example of using *bgcolor* attribute:

|  |
| --- |
| <table border="5" bordercolor="green" bgcolor="gray">  <tr>  <th>Column 1</th>  <th>Column 2</th>  <th>Column 3</th>  </tr>  <tr><td rowspan="2">Row 1 Cell 1</td>  <td bgcolor="red">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> |

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 | | | |

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_tablebgcolor_attr)

Here is an example of using *background* attribute:

|  |
| --- |
| <table border="1" background="/images/home.gif">  <tr>  <th>Column 1</th>  <th>Column 2</th>  <th>Column 3</th>  </tr>  <tr><td rowspan="2">Row 1 Cell 1</td>  <td bgcolor="red">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" background="/images/pattern1.gif">  Row 3 Cell 1  </td></tr>  </table> |

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 | | | |

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_tablebackground_attr)

# 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 integer value or in terms of percentage of available screen area. Following is the example:

|  |
| --- |
| <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> |

This will produce following result:

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

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_tablehw_attr)

# Using Table Caption:

The *caption* tags will serve as a title or explanation and show up at the top of the table. This tag is depracated in newer version of HTML/XHTML.

|  |
| --- |
| <table border="1">  <caption>This is the caption</caption>  <tr>  <td>row 1, column 1</td><td>row 1, columnn 2</td>  </tr>  </table> |

This will produce following result:

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

# Using a 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 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>

|  |
| --- |
| <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>  <tr>  ...more rows here containing four cells...  </tr>  </tbody>  <tbody>  <tr>  <td>Cell 1</td>  <td>Cell 2</td>  <td>Cell 3</td>  <td>Cell 4</td>  </tr>  <tr>  ...more rows here containing four cells...  </tr>  </tbody>  </table> |

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 | | ...more rows here containing four cells... | | | | | Cell 1 | Cell 2 | Cell 3 | Cell 4 | | ...more rows here containing four cells... | | | | |

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_tablehfb_attr)

# 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.

|  |
| --- |
| <table border="1">  <tr>  <td>  <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>  </td>  <td>  <ul>  <li>This is another cell</li>  <li>Using list inside this cell</li>  </ul>  </td>  </tr>  <tr>  <td>Row 2, Column 1</td>  <td>Row 2, Column 2</td>  </tr>  </table> |

This will produce following result:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | **Name** | **Salary** | | Ramesh Raman | 5000 | | Shabbir Hussein | 7000 | | * This is another cell * Using list inside this cell | | Row 2, Column 1 | Row 2, Column 2 | |

# HTML Frames

Frames divide a browser window into several pieces or panes, each pane containing a separate XHTML/HTML document. One of the key advantages that frames offer is that you can then load and reload single panes without having to reload the entire contents of the browser window. A collection of frames in the browser window is known as a frameset.

The window is divided up into frames in a similar pattern to the way tables are organized: into rows and columns. The simplest of framesets might just divide the screen into two rows, while a complex frameset could use several rows and columns.

There are few drawbacks also you should be aware of with frames are as follows:

* Some browsers do not print well from framesets.
* Some smaller devices cannot cope with frames, often because their screen is not big enough to be divided up.
* Some time 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 who do not support farme technology.

To create a frameset document, first you need the <frameset> element, which is used instead of the <body> element. The frameset defines the rows and columns your page is divided into, which in turn specify where each individual frame will go. Each frame is then represented by a <frame> element.

You also need to learn the <noframes> element, which provides a message for users whose browsers do not support frames.

Now we will discuss these tags in detail one by one.

# Creating Frames - The <frameset> Element:

* The <frameset> tag replaces the <body> element in frameset documents.
* The <frameset> tag defines how to divide the window into frames.
* Each frameset defines a set of rows **or** columns. If you define frames by using rows then horizontal frames are created. If you define frames by using columns then vertical farmes are created.
* The values of the rows/columns indicate the amount of screen area each row/column will occupy.
* Each farme is indicated by <frame> tag and it defines what HTML document to put into the frame.

# Example:

Following is the example to create three horizontal frames:

|  |
| --- |
| <html>  <head>  <title>Frames example</title>  </head>  <frameset rows="10%,80%,10%">  <frame src="/html/top\_frame.htm" />  <frame src="/html/main\_frame.htm" />  <frame src="/html/bottom\_frame.htm" />  <noframes>  <body>  Your browser does not support frames.  </body>  </noframes>  </frameset>  </html> |

Now create three HTML files called *top\_frame.htm*, *main\_frame.htm* and *bottom\_frame.htm* to be loaded into three frames with some content.

To become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_frameset_tag)

# The <frameset> Element Attributes:

Following are important attributes of <frameset> and should be known to you to use frameset.

* **cols:** 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:
  + Absolute values in pixels. For example to create three vertical frames, use *cols="100, 500,100"*.
  + A percentage of the browser window. For example to create three vertical frames, use *cols="10%, 80%,10%"*.
  + Using a wildcard symbol. For example to create three vertical frames, use *cols="10%, \*,10%"*. In this case wildcard takes remainder of the window.
  + As relative widths of the browser window. For example to create three vertical frames, use *cols="3\*,2\*,1\*"*. 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.
* **rows:** attribute works just like the cols attribute and can take the same values, but it is used to specify the rows in the frameset. For example to create two horizontal frames, use *rows="10%, 90%"*. You can specify the height of each row in the same way as explained above for columns.
* **border:** attribute specifies the width of the border of each frame in pixels. For example border="5". A value of zero specifies that no border should be there.
* **frameborder:** specifies whether a three-dimensional border should be displayed between frames. This attrubute takes value either 1 (yes) or 0 (no). For example frameborder="0" specifies no border.
* **framespacing:** specifies the amount of space between frames in a frameset. This can take any integer value. For example framespacing="10" means there should be 10 pixels spacing between each frames.

# Loading Content - The <frame> Element:

The <frame> element indicates what goes in each frame of the frameset. The <frame> element is always an empty element, and therefore should not have any content, although each <frame> element should always carry one attribute, src, to indicate the page that should represent that frame.

From the above example, lets take small snippet:

|  |
| --- |
| <frame src="/html/top\_frame.htm" />  <frame src="/html/main\_frame.htm" />  <frame src="/html/bottom\_frame.htm" /> |

# The <frame> Element Attributes:

Following are important attributes of and should be known to you to use frames.

* **src:** indicates the file that should be used in the frame. Its value can be any URL. For example, src="/html/top\_frame.htm" will load an HTML file avaible in html directory.
* **name:** 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 a second frame, in which case the second frame needs a name to identify itself as the target of the link.
* **frameborder:** attribute specifies whether or not the borders of that frame are shown; it overrides the value given in the frameborder attribute on the <frameset> element if one is given, and the possible values are the same. This can take values either 1 (yes) or 0 (no).
* **marginwidth:** 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:** 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:** 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:** 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 - The <noframes> Element:

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.

In XHTML 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 the <frameset> element it should understand what is inside the <body> element contained in the <noframes> element.

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

# 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 the pages with the content into a separate frame.

As you have already seen, each <frame> element can carry the *name* attribute to give each frame a name.This name is used in the links to indicate which frame the new page should load into. Consider this very simple example, create following content in index.htm file:

|  |
| --- |
| <frameset cols="200, \*">  <frame src="/html/menu.htm" name="menu\_page" />  <frame src="/html/main.htm" name="main\_page" />  </frameset> |

There are two columns in this example. The first is 200 pixels wide and will contain the navigation bar. The second column or frame will contain the main part of the page. The links on the left side navigation bar will load pages into the right side main page.

Keep some content in main.htm file and the links in the menu.htm file look like this:

|  |
| --- |
| <a href="http://www.google.com" target="main\_page">Google</a>  <br /><br />  <a href="http://www.microsoft.com" target="main\_page">Microsoft</a>  <br /><br />  <a href="http://news.bbc.co.uk/" target="main\_page">BBC News</a> |

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_frame_tag)

The *target* attribute can also take the attribute values listed in the table that follows.

|  |  |
| --- | --- |
| **Vlaue** | **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.. |

# Inline Frames - The <iframe> Element:

You can define an inline frame with the <iframe> tag. The <iframe> tag is not used within a <frameset> tag. Instead, it appears anywhere in your document. The <iframe> tag defines a rectangular region within the document in which the browser displays a separate document, including scrollbars and borders.

Use the *src* attribute with <iframe> to specify the URL of the document that occupies the inline frame.

All of the other, optional attributes for 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.

Following is the example to show how to use the <iframe>. This tag is used along with <body> tag:

|  |
| --- |
| <body>  ...other document content...  <iframe src="/html/menu.htm" width="75" height="200" align="right">  Your browser does not support inline frames. To view this  <a href="/html/menu.htm">document</a> correctly, you'll need  a copy of Internet Explorer or the latest Netscape Navigator.  </iframe>  ...subsequent document content...  </body> |

The *align* attribute lets you control where the frame gets placed in line with the adjacent text or moved to the edge of the document, allowing text to flow around the frame.

For inline alignment, use *top, middle, or bottom* as the value of this attribute. The frame is aligned with the *top, middle, or bottom* of the adjacent text, respectively. To allow text to flow around the inline frame, use the *left or right* values for this attribute. The frame is moved to the left or right edge of the text flow, respectively, and the remaining content of the document is flowed around the frame. A value of *center* places the inline frame in the middle of the display, with text flowing above and below.

# HTML Lists Formatting

You can list out your items, subjects or menu in the form of a list. HTML gives you three different types of lists.

* **<ul>** - An unordered list. This will list items using bullets
* **<ol>** - A ordered list. This will use different schemes of numbers to list your items
* **<dl>** - A definition list. This is arrange 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. The most common unordered list you will find on the Web is a collection of hyperlinks to other documents.

This list is created by using <ul> tag. Each item in the list is marked with a butllet. The bullet itself comes in three flavors: squares, discs, and circles. The default bullet displayed by most web browsers is the traditional full disc.

One Movie list is given below:

|  |
| --- |
| <center>  <h2>Movie List</h2>  </center>  <ul>  <li>Ram Teri Ganga Meli</li>  <li>Mera Naam Jocker</li>  <li>Titanic</li>  <li>Ghost in the ship</li>  </ul> |

This will produce following result:

|  |
| --- |
| Movie List  * Ram Teri Ganga Meli * Mera Naam Jocker * Titanic * Ghost in the ship |

You can use *type* attribute to specify the type of bullet you like. By default its is a disc. Following are the possible way:

|  |
| --- |
| <ul type="square">  <ul type="disc">  <ul type="circle"> |

|  |  |  |
| --- | --- | --- |
| **<ul type="square">** | **<ul type="disc">** | **<ul type="circle">** |
| * Hindi * English * Maths * Physics | * Hindi * English * Maths * Physics | * Hindi * English * Maths * Physics |

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_ul_tag)

# HTML Ordered Lists:

The typical browser formats the contents of an ordered list just like an unordered list, except that the items are numbered instead of bulleted. The numbering starts at one and is incremented by one for each successive ordered list element tagged with <li>

This list is created by using <ol> tag. Each item in the list is marked with a number.

One Movie list is given below:

|  |
| --- |
| <center>  <h2>Movie List</h2>  </center>  <ol>  <li>Ram Teri Ganga Meli</li>  <li>Mera Naam Jocker</li>  <li>Titanic</li>  <li>Ghost in the ship</li>  </ol> |

This will produce following result:

|  |
| --- |
| Movie List  1. Ram Teri Ganga Meli 2. Mera Naam Jocker 3. Titanic 4. Ghost in the ship |

You can use *type* attribute to specify the type of numbers you like. By default its is a generic numbers. Following are the other possible way:

|  |
| --- |
| <ol type="I"> - Upper-Case Numerals.  <ol type="i"> - Lower-Case Numerals.  <ol type="a"> - Lower-Case Letters.  <ol type="A"> - Upper-Case Letters. |

|  |  |  |  |
| --- | --- | --- | --- |
| **<ol type="I">** | **<ol type="i">** | **<ol type="a">** | **<ol type="A">** |
| 1. Hindi 2. English 3. Maths 4. Physics | 1. Hindi 2. English 3. Maths 4. Physics | 1. Hindi 2. English 3. Maths 4. Physics | 1. Hindi 2. English 3. Maths 4. Physics |

You can use *start* attribute to specify the beginning of any index. By default its is a first number or character. In the following example index starts from 5:

|  |
| --- |
| <center>  <h2>Movie List</h2>  </center>  <ol start="5">  <li>Ram Teri Ganga Meli</li>  <li>Mera Naam Jocker</li>  <li>Titanic</li>  <li>Ghost in the ship</li>  </ol> |

This will produce following result:

|  |
| --- |
| Movie List  1. Ram Teri Ganga Meli 2. Mera Naam Jocker 3. Titanic 4. Ghost in the ship |

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_ol_tag)

# HTML Definition Lists:

HTML and XHTML also support a list style entirely different from the ordered and unordered lists we have discussed so far - definition lists . Like the entries you find in a dictionary or encyclopedia, complete with text, pictures, and other multimedia elements, 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

Example:

|  |
| --- |
| <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> |

This will produce following result:

|  |
| --- |
| **HTML**  This stands for Hyper Text Markup Language  **HTTP**  This stands for Hyper Text Transfer Protocol |

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_dl_tag)

# Appropriate List Usage:

Following are just a suggestion and there is no hard and fast rule to use them:

Use unordered lists for:

* Link collections
* Short, nonsequenced groups of text
* Emphasizing the high points of a presentation

Use ordered lists for:

* Tables of contents
* Sets of sequential sections of text
* Assigning numbers to short phrases that can be referenced elsewhere

Use definition lists for:

* Glossaries
* Custom bullets - make the item after the <dt> tag an icon-sized bullet image)
* Any list of name/value pairs

# HTML Page Layouts

Web page layout is very important to give better look to your website. You should design your webpage layout very carefully.

You may have noticed that there are many websites which have put their content in multiple columns - they are formatted like a magazine or newspaper. This is easily achieved by using tables or division or span tags. Sometime you use CSS as well to position various elements or to create backgrounds or colorful look for the pages.

# HTML Layout - Using Tables:

The simplest and most popular way of creating layouts is using HTML <table> tag. These tables are arranged in columns and rows, so you can utilize these rows and columns in whatever way you like.

For example, the following HTML layout example is achieved using a table with 3 rows and 2 columns - but the header and footer column spans both columns using the colspan attribute:

|  |
| --- |
| <table width="100%" border="0">  <tr>  <td colspan="2" style="background-color:#CC99FF;">  <h1>This is Web Page Main title</h1>  </td>  </tr>  <tr valign="top">  <td style="background-color:#FFCCFF;  width:100px;text-align:top;">  <b>Main Menu</b><br />  HTML<br />  PHP<br />  PERL...  </td>  <td style="background-color:#eeeeee;height:200px;  width:300px;text-align:top;">  Technical and Managerial Tutorials  </td>  </tr>  <tr>  <td colspan="2" style="background-color:#CC99FF;">  <center>  Copyright © 2007 Tutorialspoint.com  </center>  </td>  </tr>  </table> |

This will produce following result:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | This is Web Page Main title | | | **Main Menu** HTML PHP PERL... | Technical and Managerial Tutorials | | Copyright © 2007 Tutorialspoint.com | | |

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_table_layout)

# Multiuple Columns Layouts - Using Tables

You can design your webpage to put your web content in multiple pages. You can keep your content in middle column and you can use left column to use menu and right column can be used to put advertisement or some other stuff. It will be very similar to our site tutorialspoint.com.

Here is an example to create three column layout:

|  |
| --- |
| <table width="100%" border="0">  <tr valign="top">  <td style="background-color:#FFCCFF;width:20%;  text-align:top;">  <b>Main Menu</b><br />  HTML<br />  PHP<br />  PERL...  </td>  <td style="background-color:#eeeeee;height:200px;  width:60%;text-align:top;">  Technical and Managerial Tutorials  </td>  <td style="background-color:#FFCCFF;  width:20%;text-align:top;">  <b>Right Menu</b><br />  HTML<br />  PHP<br />  PERL...  </td>  </tr>  <table> |

This will produce following result:

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **Main Menu** HTML PHP PERL... | Technical and Managerial Tutorials | **Right Menu** HTML PHP PERL... | |

# 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, tables weren't really designed as a layout tool. Tables are more suited to presenting tabular data.

You can achieve same result whatever you have achieved using <table> tag in previous example.

|  |
| --- |
| <div style="width:100%">  <div style="background-color:#CC99FF;">  <b style="font-size:150%">This is Web Page Main title</b>  </div>  <div style="background-color:#FFCCFF;  height:200px;width:100px;float:left;">  <b>Main Menu</b><br />  HTML<br />  PHP<br />  PERL...  </div>  <div style="background-color:#eeeeee;  height:200px;width:300px;float:left;">  Technical and Managerial Tutorials  </div>  <div style="background-color:#CC99FF;clear:both">  <center>  Copyright © 2007 Tutorialspoint.com  </center>  </div>  </div> |

This will produce following result:

|  |
| --- |
| **This is Web Page Main title** |

# HTML Forms Tutorial

HTML Forms are required when you want to collect some data from the site visitor. For example registration information: name, email address, credit card, etc.

A form will take input from the site visitor and then will post your back-end application such as CGI, ASP Script or PHP script etc. Then your back-end application will do required processing on that data in whatever way you like.

Form elements are like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc. which are used to take information from the user.

A simple syntax of using <form> is as follows:

|  |
| --- |
| <form action="back-end script" method="posting method">  form elements like input, textarea etc.  </form> |

Most frequently used form attributes are:

* **name:** This is the name of the form.
* **action:** Here you will specify any script URL which will receive uploaded data.
* **method:** Here you will specify method to be used to upload data. It can take various values but most frequently used are GET and POST.
* **target:** It specifies the target page where the result of the script will be displayed. It takes values like \_blank, \_self, \_parent etc.
* **enctype:** You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server. Possible values are like:
  + **application/x-www-form-urlencoded** - This is the standard method most forms use. It converts spaces to the plus sign and non-alphanumeric characters into the hexadecimal code for that character in ASCII text.
  + **mutlipart/form-data** - This allows the data to be sent in parts, with each consecutive part corresponding the a form control, in the order they appear in the form. Each part can have an optional content-type header of its own indicating the type of data for that form control.

Please refer to [PERL & CGI](http://www.tutorialspoint.com/perl/perl_cgi.htm) for a detail on data uploading using CGI.

There are different types of form controls that you can use to collect data from a visitor to your site.

* Text input controls
* Buttons
* Checkboxes and radio buttons
* Select boxes
* File select boxes
* Hidden controls
* Submit and reset button

# HTML Forms - Text Input Controls:

There are actually three types of text input used on forms:

* **Single-line text input controls:** Used for items that require only one line of user input, such as search boxes or names. They are created using the <input> element.
* **Password input controls:** Single-line text input that mask the characters a user enters.
* **Multi-line text input controls:** Used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created with the <textarea> element.

# Single-line text input controls:

Single-line text input controls are created using an <input> element whose type attribute has a value of text.

Here is a basic example of a single-line text input used to take first name and last name:

|  |
| --- |
| <form action="/cgi-bin/hello\_get.cgi" method="get">  First name:  <input type="text" name="first\_name" />  <br>  Last name:  <input type="text" name="last\_name" />  <input type="submit" value="submit" />  </form> |

This will produce following result:

|  |
| --- |
| Top of Form  First name:  Last name:  Bottom of Form |

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

* **type:** Indicates the type of input control you want to create. This element is also used to create other form controls such as radio buttons and checkboxes.
* **name:** Used to give the name part of the name/value pair that is sent to the server, representing each form control and the value the user entered.
* **value:** Provides an initial value for the text input control that the user will see when the form loads.
* **size:** Allows you to specify the width of the text-input control in terms of characters.
* **maxlength:** Allows you to specify the maximum number of characters a user can enter into the text box.

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_input_text)

# Password input controls::

This is also a form of single-line text input controls are created using an <input> element whose type attribute has a value of password.

Here is a basic example of a single-line password input used to take user password:

|  |
| --- |
| <form action="/cgi-bin/hello\_get.cgi" method="get">  Login :  <input type="text" name="login" />  <br>  Password:  <input type="text" name="password" />  <input type="submit" value="submit" />  </form> |

This will produce following result:

|  |
| --- |
| Top of Form  Login :  Password :  Bottom of Form |

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_input_password)

# Multiple-Line Text Input Controls:

If you want to allow a visitor to your site to enter more than one line of text, you should create a multiple-line text input control using the <textarea> element.

Here is a basic example of a multi-line text input used to take item description:

|  |
| --- |
| <form action="/cgi-bin/hello\_get.cgi" method="get">  Description : <br />  <textarea rows="5" cols="50" name="description">  Enter description here...  </textarea>  <input type="submit" value="submit" />  </form> |

This will produce following result:

|  |
| --- |
| Top of Form  Description :  Bottom of Form |

Following is the detail of above used attributes for <textarea> tag.

* **name:** The name of the control. This is used in the name/value pair that is sent to the server.
* **rows:** Indicates the number of rows of text area box.
* **cols:** Indicates the number of columns of text area box.

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_input_textarea)

# HTML Forms - Creating Button:

There are various ways in HTML to create clickable buttons. You can create clickable button using <input> tag.

When you use the <input> element to create a button, the type of button you create is specified using the type attribute. The type attribute can take the following values:

* **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.

Here is the example:

|  |
| --- |
| <form action="http://www.example.com/test.asp" method="get">  <input type="submit" name="Submit" value="Submit" />  <br /><br />  <input type="reset" value="Reset" />  <input type="button" value="Button" />  </form> |

This will produce following result:

|  |
| --- |
| Top of Form    Bottom of Form |

You can use an image to create a button. Here is the syntax:

|  |
| --- |
| <form action="http://www.example.com/test.asp" method="get">  <input type="image" name="imagebutton" src="URL" />  </form> |

Here *src* attribiute specifies a location of the image on your webserver.

You can use <button> element to create various buttons. Here is the syntax:

|  |
| --- |
| <form action="http://www.example.com/test.asp" method="get">  <button type="submit">Submit</button>  <br /><br />  <button type="reset"> Reset </button>  <button type="button"> Button </button>  </form> |

This will produce following result:

|  |
| --- |
| Top of Form  Submit   ResetButton  Bottom of Form |

# HTML Forms - Checkboxes Control:

Checkboxes are used when more than one option is required to be selected. They are created using <input> tag as shown below.

Here is example HTML code for a form with two checkboxes

|  |
| --- |
| <form action="/cgi-bin/checkbox.cgi" method="get">  <input type="checkbox" name="maths" value="on"> Maths  <input type="checkbox" name="physics" value="on"> Physics  <input type="submit" value="Select Subject" />  </form> |

The result of this code is the following form

|  |
| --- |
| Top of Form  Maths Physics  Bottom of Form |

Following is the list of important checkbox attributes:

* **type:** Indicates that you want to create a checkbox.
* **name:** Name of the control.
* **value:** The value that will be used if the checkbox is selected. More than one checkbox should share the same name only if you want to allow users to select several items from the same list.
* **checked:** Indicates that when the page loads, the checkbox should be selected.

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_input_checkbox)

# HTML Forms - Raidobox Control:

Radio Buttons are used when only one option is required to be selected. They are created using <input> tag as shown below:

Here is example HTML code for a form with two radio button:

|  |
| --- |
| <form action="/cgi-bin/radiobutton.cgi" method="post">  <input type="radio" name="subject" value="maths" /> Maths  <input type="radio" name="subject" value="physics" /> Physics  <input type="submit" value="Select Subject" />  </form> |

The result of this code is the following form

|  |
| --- |
| Top of Form  Maths Physics  Bottom of Form |

Following is the list of important radiobox attributes:

* **type:** Indicates that you want to create a radiobox.
* **name:** Name of the control.
* **value:** Used to indicate the value that will be sent to the server if this option is selected.
* **checked:** Indicates that this option should be selected by default when the page loads.

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_input_radiobox)

# HTML Forms - Select box Control:

Drop Down Box is used when we have many options available to be selected but only one or two will be selected..

Here is example HTML code for a form with one drop down box

|  |
| --- |
| <form action="/cgi-bin/dropdown.cgi" method="post">  <select name="dropdown">  <option value="Maths" selected>Maths</option>  <option value="Physics">Physics</option>  </select>  <input type="submit" value="Submit" />  </form> |

The result of this code is the following form

|  |
| --- |
| Top of Form    Bottom of Form |

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

* **name:** This is the name for the control.
* **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>:

* **value:** The value that is sent to the server if this option is selected.
* **selected:** Specifies that this option should be the initially selected value when the page loads.
* **label:** An alternative way of labeling options.

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_input_selectbox)

# HTML Forms - File Select Boxes:

If you want to allow a user to upload a file to your web site from his computer, you will need to use a file upload box, also known as a file select box. This is also created using the <input> element.

Here is example HTML code for a form with one file select box

|  |
| --- |
| <form action="/cgi-bin/hello\_get.cgi" method="post"  name="fileupload" enctype="multipart/form-data">  <input type="file" name="fileupload" accept="image/\*" />  </form> |

The result of this code is the following form

|  |
| --- |
| Top of Form  Bottom of Form |

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_input_file)

# HTML Forms - Hidden Controls:

If you will want to pass information between pages without the user seeing it. Hidden form controls remain part of any form, but the user cannot see them in the Web browser. They should not be used for any sensitive information you do not want the user to see because the user could see this data if she looked in the source of the page.

Following hidden form is being used to keep current page number. When a user will click next page then the value of hidden form will be sent to the back-end application and it will decide which page has be displayed next.

|  |
| --- |
| <form action="/cgi-bin/hello\_get.cgi"  method="get" name="pages">  <p>This is page 10</p>  <input type="hidden" name="pgaenumber" value="10" />  <input type="submit" value="Next Page" />  </form> |

This will produce following result:

|  |
| --- |
| Top of Form  This is page 10    Bottom of Form |

To Become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_input_hidden)

# HTML Forms - Submit and Reset Button:

These are special buttons which can be created using <input> When submit button is clicked then Forms data is submitted to the back-end application. When reset button is clicked then all the forms control are reset to default state.

You already have seen submit button above, we will give one reset example here:

|  |
| --- |
| <form action="/cgi-bin/hello\_get.cgi" method="get">  First name:  <input type="text" name="first\_name" />  <br>  Last name:  <input type="text" name="last\_name" />  <input type="submit" value="Submit" />  <input type="reset" value="Reset" />  </form> |

This will produce following result. Type something and click reset button.

|  |
| --- |
| Top of Form  First name:  Last name:  Bottom of Form |

# HTML Styles

Style sheets describe how documents are presented on screens, in print, or perhaps how they are pronounced. W3C has actively promoted the use of style sheets on the Web since the Consortium was founded in 1994.

Cascading Style Sheets (CSS) is a style sheet mechanism that has been specifically developed to meet the needs of Web designers and users.

With CSS, you can specify a number of style properties for a given HTML element. Each property has a name and a value, separated by a colon (:). Each property declaration is separated by a semi-colon (;).

|  |
| --- |
| <p style="color:red;font-size:24px;">Using Style Sheet Rules</p> |

This will produce following result:

|  |
| --- |
| Using Style Sheet Rules |

There are three ways of using a style sheet in an HTML document:

# External Style Sheet:

If you have to give same look and feel to many pages then it is a good idea to keep all the style sheet rules in a single style sheet file and include this file in all the HTML pages. You can incluse a style sheet file into HTML document using <link> element. Below is an example:

|  |
| --- |
| <head>  <link rel="stylesheet" type="text/css"  href="yourstyle.css">  </head> |

# Internal Style Sheet:

If you want to apply Style Sheet rules to a single document only then you can include those rules into that document only. Below is an example:

|  |
| --- |
| <head>  <style type="text/css">  body{background-color: pink;}  p{color:blue; 20px;font-size:24px;}  </style>  </head> |

To become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_style_tag)

# Inline Style Sheet:

You can apply style sheet rules directly to any HTML element. This should be done only when you are interested to make a particular change in any HTML element only. To use inline styles you use the style attribute in the relevant tag. Below is an example:

|  |
| --- |
| <p style="color:red;font-size:24px;">Using Style Sheet Rules</p> |

This will produce following result:

|  |
| --- |
| Using Style Sheet Rules |

# HTML Scripts

A *script* is a small piece of program that can add interactivity to your website. For example, a script could generate a pop-up alert box message, or provide a dropdown menu. This script could be Javascript or VBScript.

You can write your Event Handlers using any of the scripting language and then you can trigger those functions using HTML attributes.

There are two ways of using a style sheet in an HTML document:

# External Script:

If you have to use a single script functionality among many HTML pages then it is a good idea to keep that function in a single script file and then include this file in all the HTML pages. You can incluse a style sheet file into HTML document using <script> element. Below is an example:

|  |
| --- |
| <head>  <script src="yourfile.js" type="text/javascript" />  </head> |

# Internal Script:

You can write your script code directly into your HTML document. Usually we keep script code in header of the document using <script> tag, otherwise there is no restriction and you can put your source code anywhere in the document. You can specify whether to make a script run automatically (as soon as the page loads), or after the user has done something (like click on a link). Below is an example this would write a *Hello Javascript!* message as soon as the page loads.:

|  |
| --- |
| <head>  <title>Internal Script</title>  </head>  <body>  <script type="text/javascript">  document.write("Hello Javascript!")  </script>  </body> |

This will produce following result:

|  |
| --- |
| Hello Javascript! |

To become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_script_tag)

# Writing Event Handler:

It is very easy to write an event handler. Following example explains how to write an event handler. Let's write one simple function *myAlert* in the header of the document. We will call this function when any user will bring mouse over a paragraph written in the example.

|  |
| --- |
| <head>  <title>Event Handler Example t</title>  <script type="text/javascript">  function myAlert()  {  alert("I am an event handler....");  return;  }  </script>  </head>  <body>  <span onmouseover="myAlert();">  Bring your mouse here to see an alert  </span>  </body> |

Now this will produce following result. Bring your mouse over this line and see the result:

|  |
| --- |
| Bring your mouse here to see an alert |

To become more comfortable - [Do Online Practice](http://www.tutorialspoint.com/cgi-bin/practice.cgi?file=html_event_tag)

# Hide Scripts from Older Browsers

Athough most (if not all) browsers these days support scripts, some older browsers don't. If a browser doesn't support JavaScript, instead of running your script, it would display the code to the user. To prevent this from happening, you can simply place HTML comments around the script. Older browsers will ignore the script, while newer browsers will run it.

|  |
| --- |
| **JavaScript Example:**  <script type="text/javascript">  <!--  document.write("Hello Javascript!");  //-->  </script>  **VBScript Example:**  <script type="text/vbscript">  <!--  document.write("Hello VBScript!")  '-->  </script> |

# The <noscript> Element:

You can also provide alternative info for users whose browsers don't support scripts and for users who have disabled scripts. You do this using the *<noscript>* tag.

|  |
| --- |
| **JavaScript Example:**  <script type="text/javascript">  <!--  document.write("Hello Javascript!");  //-->  </script>  <noscript>Your browser does not support Javascript!</noscript>  **VBScript Example:**  <script type="text/vbscript">  <!--  document.write("Hello VBScript!")  '-->  </script>  <noscript>Your browser does not support VBScript!</noscript> |

## Default Scripting Language

You can specify a default scripting language for all your *script* tags to use. This saves you from having to specify the language everytime you use a script tag within the page. Below is the example:

|  |
| --- |
| <meta http-equiv="Content-Script-Type" content="text/JavaScript" /> |

Note that you can still override the default by specifying a language within the script tag.