

HTML Tutorial

HTML or Hypertext Markup Language, is the most widely used language on Web. Technically, HTML is not a programming language, but rather a markup language. This tutorial gives a complete understanding on HTML.

Before you begin, it's important that you know Windows or Unix. A working knowledge of Windows or Unix makes it much easier to learn HTML.

You should be familiar with:

- Basic word processing using any text editor.
- How to create directories and files.
- How to navigate through different directories.
- Basic understanding on internet browsing using a browser like Internet Explorer or Firefox etc.

Introducing HTML:

HTML stands for **H**ypertext **M**arkup **L**anguage, and it is the most widely used language to write Web Pages. As its name suggests, HTML is a markup language.

- **Hypertext** refers to the way in which Web pages (HTML documents) are linked together. When you click a link in a Web page, you are using hypertext.
- **Markup Language** describes how HTML works. With a markup language, you simply "mark up" a text document with tags that tell a Web browser how to structure it to display.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers.

All you need to do to use HTML is to learn what type of markup to use to get the results you want.

Creating HTML Document:

Creating an HTML document is easy. To begin coding HTML you need only two things: a simple- text editor and a web browser. Notepad is the most basic of simple-text editors and you will probably code a fair amount of HTML with it.

Here are the simple steps to create a basic HTML document:

- Open Notepad or another text editor.
- At the top of the page type <html>.
- On the next line, indent five spaces and now add the opening header tag: <head>.
- On the next line, indent ten spaces and type <title> </title>.
- Go to the next line, indent five spaces from the margin and insert the closing header tag: </head>.
- Five spaces in from the margin on the next line, type<body>.
- Now drop down another line and type the closing tag right below its mate: </body>.
- Finally, go to the next line and type </html>.
- In the File menu, choose Save As.
- In the Save as Type option box, choose All Files.
- Name the file template.htm.
- Click Save.

You have basic HTML document now, to see some result put the following code in title and body tags.

```
<html>
<head>
<title>This is document title</title>
</head>
<body>
<h1>This is a heading</h1>
<p>Document description goes here.....</p>
</body>
</html>
```

Now you have created one **HTML page** and you can use a Web Browser to open this HTML file to see the result. Hope you understood that Web Pages are nothing but they are simple HTML files with some content which can be rendered using Web Browsers.

Here <html>, <head>,...<p>, <h1> etc. are called HTML tags. HTML tags are building blocks of an HTML document and we will learn all the HTML tags in subsequent chapters.

NOTE: One HTML file can have extension as **.htm** or **.html**. So you can use either of them based on your comfort.

HTML Document Structure:

An HTML document starts and ends with <html> and >/html> tags. These tags tell the browser that the entire document is composed in HTML. Inside these two tags, the document is split into two sections:

- The <head>...</head> elements, which contain information about the document such as title of the document, author of the document etc. Information inside this tag does not display outside.
- The <body>...</body> elements, which contain the real content of the document that you see on your screen.

HTML Tags and Elements:

HTML language is a markup language and we use many tags to markup text. In the above example you have seen <html>, <body> etc. are called HTML tags or HTML elements.

Every tag consists of a tag name, sometimes followed by an optional list of tag attributes, all placed between opening and closing brackets (< and >). The simplest tag is nothing more than a name appropriately enclosed in brackets, such as <head> and <i>. More complicated tags contain one or more attributes, which specify or modify the behaviour of the tag.

According to the HTML standard, tag and attribute names are not case-sensitive. There's no difference in effect between <head>, <Head>, <HEAD>, or even <HeaD>; they are all equivalent. But with XHTML, case is important: all current standard tag and attribute names are in lowercase.

HTML is Forgiving?

A very good quality associated with all the browsers is that they would not give any error if you have not put any HTML tag or attribute properly. They will just ignore that tag or attribute and will apply only correct tags and attributes before displaying the result.

We cannot say, HTML is forgiving because this is just a markup language and required to format documents.

HTML Basic Tags

The basic structure for all HTML documents is simple and should include the following minimum elements or tags:

- **<html>** - The main container for HTML pages
- **<head>** - The container for page header information
- **<title>** - The title of the page
- **<body>** - The main body of the page

Remember that before an opening `<html>` tag, an XHTML document can contain the optional XML declaration, and it should always contain a DOCTYPE declaration indicating which version of XHTML it uses.

Now we will explain each of these tags one by one. In this tutorial you will find the terms element and tag are used interchangeably.

The **<html>** Element:

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

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

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

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

The **<head>** Element:

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

Each `<head>` element should contain a `<title>` element indicating the title of the document.

The **<title>** Element:

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

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

Therefore it is important to use a title that really describes the content of your site. The `<title>` element should contain only the text for the title and it may not contain any other elements.

Example:

Here is the example of using title tag.

```
<head>
<title>HTML Basic tags</title>
</head>
```

The <body> Element:

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

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

Most of what you will be learning in this and the following five chapters will be written between the opening <body> tag and closing </body> tag.

Example:

Here is the example of using body tag.

```
<body>
  <p>This is a paragraph tag.</p>
</body>
```

Putting all together:

Now if we will put all these tags together, it will constitute a complete HTML document as follows:

```
<html>
<head>
  <title>HTML Basic tags</title>
</head>
<body>
  <p>This is a paragraph tag.</p>
</body>
</html>
```

To Become more comfortable

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 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 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.

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, hexadecimal, 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.

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 <h> 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:

This is heading 1

This is heading 2

This is heading 3

This is heading 4

This is heading 5

This is heading 6

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:

Here is a paragraph of text.

Here is a second paragraph of text.

Here is a third paragraph of text.

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:

This is left aligned.

This is center aligned.

This is right aligned.

This is justified. This works when you have multiple lines in your paragraph and you want to justify all the lines so that they can look more nice.

Create Line Breaks - The `
` Element:

Whenever you use the `
` 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 `
` 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 `
` it is not valid XHTML

Example:

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

This will produce following result:

Hello

You come most carefully upon your hour.

Thanks

Mahnaz

Centering 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:

This is not in the center.

This is in the center.

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>
    *
 *      *
*      *      *
</pre>
```

This will produce following result:

```
    *
 *      *
*      *      *
```

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:

```
This is paragraph one and should be on top

This is paragraph two and should be at bottom
```

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 hr 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 `` Element:

Anything that appears in a `...` 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.

To Become more comfortable

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.

To Become more comfortable

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.

To Become more comfortable

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.
```

To Become more comfortable -

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.
```

To Become more comfortable

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.

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

This will produce following result:

```
The following word uses a big typeface.
```

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:

```
The following word uses a small typeface.
```

HTML Phrase Tags

While some of these phrase elements are displayed in a similar manner to the , <i>, <pre>, and <tt> elements you have already seen, they are designed for specific purposes. For example, the and 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 Element:

The content of an 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.
```

Strong Text - The Element:

The element is intended to show strong emphasis for its content; stronger emphasis than the element. As with the element, the 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.
```

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>, , , <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 , <i>, <u>, , , <sup>, <sub>, <big>, <small>, , <ins>, , <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 browser. 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.  
-->
```

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>
```

HTML Fonts

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

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](#)

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

HTML Images

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

Insert Image - The Element:

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

```

```

Image Attributes:

Following are most frequently used attributes for 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 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](#) chapter.

A Simple Example:

```

```

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

Now let us try to set some more attributes:

```

```

Wrapping text around images:

Example 1:

```
<p>This is the first paragraph that appears above the paragraph with the image!</p>
```

```
<p>
```

The image will appear along the right hand side of the paragraph. As you can see this is very nice for adding a little eye candy that relates to the specified paragraph.</p>

<p>The left and right image-alignment values tell the browser to place an image against the left or right margin, respectively, of the current text flow. The browser then renders subsequent document content in the remaining portion of the flow adjacent to the image. The net result is that the document content following the image gets wrapped around the image. </p>

Example 2:

You can use vspace or hspace attributes if you want to keep some distance between text and image. Let us revise above example:

```
<p>This is the first paragraph that appears above the paragraph with the image!</p>
```

```
<p>
```

The image will appear along the right hand side of the paragraph. As you can see this is very nice for adding a little eye candy that relates to the specified paragraph.</p>

<p>The left and right image-alignment values tell the browser to place an image against the left or right margin, respectively, of the current text flow. The browser then renders subsequent document content in the remaining portion of the flow adjacent to the image. The net result is that the document content following the image gets wrapped around the image. </p>

HTML Text Links

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

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 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 or 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 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.kes.ac.in/" target="_blank" >TPB 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:

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)

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 can use *border="0"*. The border attribute and other attributes also mentioned in this session are deprecated and they have been replaced by CSS. So it is recommended to use CSS instead of using any attribute directly.

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

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 attributes 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>
```

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>
```

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>
```

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>
```

Table height and width:

You can set a table width and height using *width* and *height* attributes. 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	

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 deprecated 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	

HTML Colors

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

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

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

NOTE: It is recommended to use CSS to set background or text colors.

HTML Color Coding Methods:

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

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

















Now we will see these coloring schemes one by one.

HTML Colors - Color Names:

You can specify direct a color name to set text or background color. W3C has listed 16 basic color names that will validate with an HTML validator but there are over 200 different color names supported by Netscape and IE. Check a complete list of [HTML Color Name](#).

W3C Standard 16 Colors:

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










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

HTML Colors - Hex Codes:

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

A hexadecimal value can be taken from any graphics software like Adobe Photoshop, Jasc Paintshop Pro or even using Advanced Paint Brush.

Each hexadecimal code will be preceded by a pound or hash sign #. Following are the examples to use Hexadecimal notation.

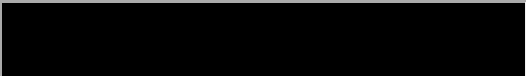






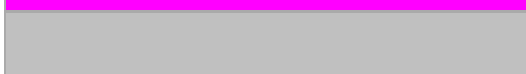

Color	Color HEX
	#000000
	#FF0000
	#00FF00
	#0000FF
	#FFFF00
	#00FFFF
	#FF00FF
	#C0C0C0
	#FFFFFF

HTML Colors - RGB Values:

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

NOTE: All the browsers does not support rgb() property of color so it is recommended not to use it.

Following is the example to show few colors using RGB values.

Color	Color RGB
	rgb(0,0,0)
	rgb(255,0,0)
	rgb(0,255,0)
	rgb(0,0,255)
	rgb(255,255,0)
	rgb(0,255,255)
	rgb(255,0,255)
	rgb(192,192,192)
	rgb(255,255,255)

Building Color Codes:

You can build millions of color codes using our Color Code Builder.

Browser Safe Colors:

Here is the list of 216 colors which are supposed to be most safe and computer independent colors. These colors vary from hexa code 000000 to FFFFFFFF. These color are safe to use because they ensure that all computers would display the colors correctly when running a 256 color palette:

000000	000033	000066	000099	0000CC	0000FF
003300	003333	003366	003399	0033CC	0033FF
006600	006633	006666	006699	0066CC	0066FF
009900	009933	009966	009999	0099CC	0099FF

00CC00	00CC33	00CC66	00CC99	00CCCC	00CCFF
00FF00	00FF33	00FF66	00FF99	00FFCC	00FFFF
330000	330033	330066	330099	3300CC	3300FF
333300	333333	333366	333399	3333CC	3333FF
336600	336633	336666	336699	3366CC	3366FF
339900	339933	339966	339999	3399CC	3399FF
33CC00	33CC33	33CC66	33CC99	33CCCC	33CCFF
33FF00	33FF33	33FF66	33FF99	33FFCC	33FFFF
660000	660033	660066	660099	6600CC	6600FF
663300	663333	663366	663399	6633CC	6633FF
666600	666633	666666	666699	6666CC	6666FF
669900	669933	669966	669999	6699CC	6699FF
66CC00	66CC33	66CC66	66CC99	66CCCC	66CCFF
66FF00	66FF33	66FF66	66FF99	66FFCC	66FFFF
990000	990033	990066	990099	9900CC	9900FF
993300	993333	993366	993399	9933CC	9933FF
996600	996633	996666	996699	9966CC	9966FF
999900	999933	999966	999999	9999CC	9999FF
99CC00	99CC33	99CC66	99CC99	99CCCC	99CCFF
99FF00	99FF33	99FF66	99FF99	99FFCC	99FFFF
CC0000	CC0033	CC0066	CC0099	CC00CC	CC00FF
CC3300	CC3333	CC3366	CC3399	CC33CC	CC33FF
CC6600	CC6633	CC6666	CC6699	CC66CC	CC66FF

CC9900	CC9933	CC9966	CC9999	CC99CC	CC99FF
CCCC00	CCCC33	CCCC66	CCCC99	CCCCCC	CCCCFF
CCFF00	CCFF33	CCFF66	CCFF99	CCFFCC	CCFFFF
FF0000	FF0033	FF0066	FF0099	FF00CC	FF00FF
FF3300	FF3333	FF3366	FF3399	FF33CC	FF33FF
FF6600	FF6633	FF6666	FF6699	FF66CC	FF66FF
FF9900	FF9933	FF9966	FF9999	FF99CC	FF99FF
FFCC00	FFCC33	FFCC66	FFCC99	FFCCCC	FFCCFF
FFFF00	FFFF33	FFFF66	FFFF99	FFFFCC	