Introduction to HTML

World Wide Web - Internet

The World Wide Web(WWW) has been defined as ”a global, interactive, dynamic, cross-platform, distributed, graphical, hypertext information system that runs over the Internet”

(a) *Web is Interactive:* Communicate with the publisher of the pages your are reading and with other readers of those pages.

(b) *Web is Dynamic:* Information on the Web and in the site that published can update it any time.

(c) *Web is Cross-Platform:* Can Access Web Information equally through an application called browser.

(d) *Web is Distributed:* Information is distributed globally across thousands of Web sites, A Web site is a location on the Web that publishes some kind of information.

(e) *Graphical:* Displays both text and graphics.

(f) *Hypertext Information System:* Enables you to read and navigate text in a monitor way.

HTML - (HyperText Markup Language)

- *Hypertext* is an ordinary text that has been dressed up with extra features, such as formatting, images, multimedia, and links to other documents.
- *Markup* is the process of taking ordinary text and adding extra symbols(markup tags).
- The markup tags tells the browser how to display the page.

There have been a number of versions of HTML that have been released over the past few years in which HTML5 being the latest release.

HTML5 has a set of new HTML elements added to HTML, plus redefinitions of some of the existing HTML elements.

By using HTML, we can design rich, colorful, hypertext-oriented, interactive screens, forms for web-based applications and websites that include text, images and data. Whereas, by using HTML5 we can design a form for your website, and we can include graphics and import multimedia into your site to create an interactive experience for the users.

Creating Simple HTML Document

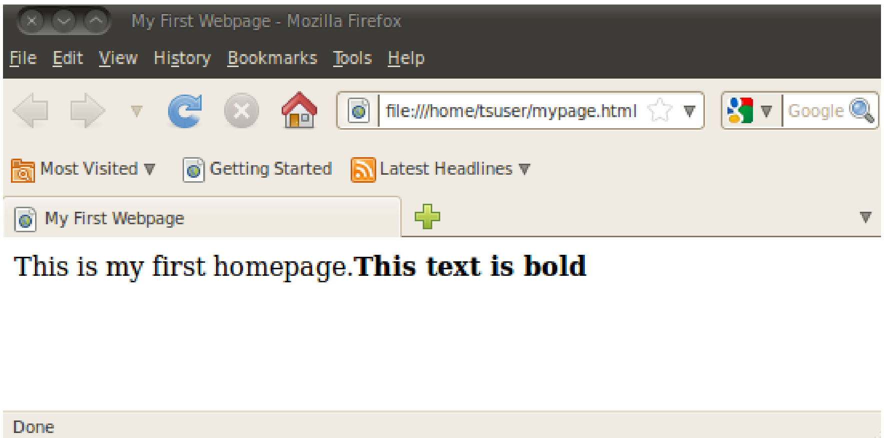
Open your text editor and type the following text:

```
<!DOCTYPE html>
<html>
     <head>
          <meta charset="utf-8">
          <title>My First Webpage</title>
     </head>
     <body>This is my first homepage.<b>This text is bold</b>
     </body>
</html>
```

Steps to follow:

1. Save the file as mypage.html or mypage.htm
2. Click on the Web browser(Mozilla firefox, Google, Internet Explorer)
3. Select Open (or Open Page) in the File menu of your browser. (A dialog box will appear)
4. Select Browse and locate the html file you just created - mypage.html
    - Select it and click Open.
    - Now you should see an address in the dialog box,
      for example - file:///home/tsuser/mypage.html
5. Click OK, The browser will display the page.

Output:


My First Webpage - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///home/tsuser/mypage.html
Google
Most Visited
Getting Started
Latest Headlines
My First Webpage
Done

This is my first homepage.**This text is bold**

HTML Tags

HTML Tags are used to mark-up HTML elements.

- They are surrounded by the two characters $< and >$ (angle brackets).
- Normally in pairs like $< b >$ and $< /b >$ where first is start tag and the second tag is end tag.
- The text between the start and end tags is the element content.
- Not case sensitive

Tag Attributes:

- It provides additional information about the HTML elements.
- The $< tag >$ tells the browser to do something, while the attribute tells the browser how to do it.
- It always come in name/value pairs like this: $name = "value"$.
- name is added to the start tag of an HTML element and the value is surrounded by single or double quotes.

Basic HTML Tags:

Tag Name	Tag	Description
Starting Tag	$< html >.....< /html >$	Used to Display Webpages
Body Tag	$< body >.....< /body >$	Body of the document
Heading Tag	$< h1 >.....< /h1 >$ $< h2 >.....< /h2 >$ .... $< h6 >......< /h6 >$	Descriptive info, for displaying Headings in different formats
Paragraph Tag	$< p >.....< /p >$	Starts and Stops the Paragraph
Break Tag	$< br >$	Used to break a line if needed
Horizontal Rule Tag	$< hr >$	Horizontal Line
Comments...Tag	$<! - sometext.... - - >$	To comment the required text

`<!DOCTYPE>` Declaration

Every HTML document should start with a $<!DOCTYPE >$ declaration that tells the browser what version of HTML the document is written in. Whereas $<!DOCTYPE >$ used in current version of HTML(i.e., HTML5) is much shorter than previous version of HTML(i.e., HTML4).

HTML4:

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

HTML5:

```
<!DOCTYPE HTML>
```

Character encoding declaration

A charater encoding declaration is a mechanism for specifying the character encoding used to store or transmit a document.

The first line inside $< head >$ is the one that defines the character encoding for the document. This is another element that's been simplified.

HTML4:

```
<meta http-equiv="Content-Type" content="text/html;
                                              charset=utf-8">
```

HTML5 improves on this by reducing the character encoding $< meta >$ tag to the bare minimum:

HTML5:

```
<meta charset="utf-8">
```

In nearly all cases, $utf-8$ is the value you will be using in your documents. A full explanation of character encoding is beyond the scope of this chapter, and it probably wont be that interesting to you, either. Nonetheless, if you want to delve a little deeper,

you can go to this link http://www.w3schools.com/html/html_charset.asp

HTML Elements

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

Explanation:

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.

Example:

```
<!DOCTYPE html>
    <html>
        <head>
            <meta charset="utf-8">
            <title> My First Webpage </title>
        </head>
    </html>
```

```
<!DOCTYPE html>
    <html>
        <head>
            <meta charset=" utf −8">
            <title> My First Webpage </title>
        </head>
    </html>
```

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.
- It 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:

```
<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.

Example:

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

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.
- The *value* is what you want the value of the property to be.

The value of the attribute should be put in double quotation marks, and is separated from the name by the equals sign.

Alignment:

- It is possible to align block elements (tables, images, objects, paragraphs, etc.) on the canvas with the **align** attribute$(name = align)$.
- Although this attribute may be set for many HTML elements, its range of possible values sometimes differs from element to element.
- Here we only discuss the meaning of the align attribute for text
- Possible values for the align attribute are as follows:

Possible Values:

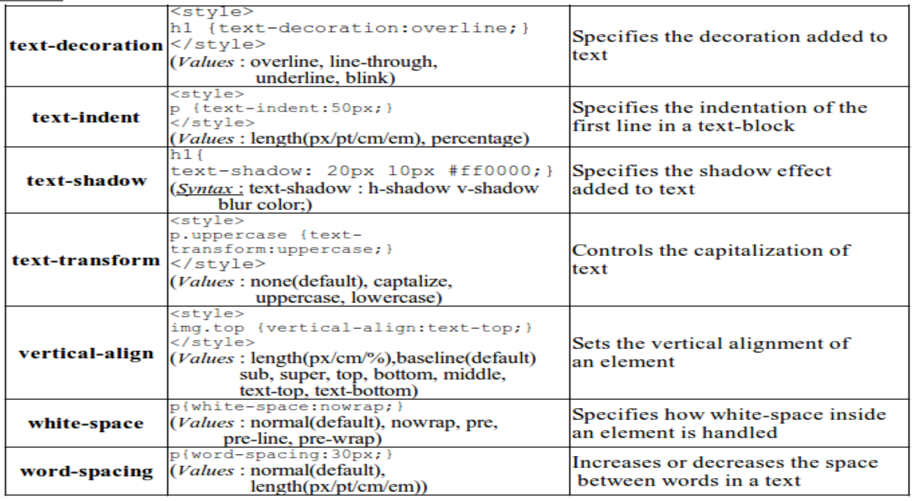
Values	Description
left	text lines are rendered flush left.
center	text lines are centered.
right	text lines are rendered flush right.
justify	text lines are justified to both margins.

Example:

```
<!DOCTYPE html>
    <html>
        <head>
            <meta charset="utf-8">
            <title>My First Webpage</title>
        </head>
        <body>
            <h1>This is a heading</h1>
            <h2 align="center">This is a heading</h2>
            <h3 align="right">This is a heading</h3>
            <h4 align="center">This is a heading</h4>
            <h5 align="left">This is a heading</h5>
```

```
                <h6 align="right">This is a heading</h6>
            </body>
        </html>
```

Output:



text-decoration	`<style>`
	`h1 {text-decoration:overline;}`
	`</style>`
	(*Values* : overline, line-through, underline, blink)
		Specifies the decoration added to text

text-indent	`<style>`
	`p {text-indent:50px;}`
	`</style>`
	(*Values* : length(px/pt/cm/em), percentage)
		Specifies the indentation of the first line in a text-block

text-shadow	`h1{`
	`text-shadow: 20px 10px #ff0000;}`
	(*Syntax* : text-shadow : h-shadow v-shadow blur color;)
		Specifies the shadow effect added to text

text-transform	`<style>`
	`p.uppercase {text-transform:uppercase;}`
	`</style>`
	(*Values* : none(default), captalize, uppercase, lowercase)
		Controls the capitalization of text

vertical-align	`<style>`
	`img.top {vertical-align:text-top;}`
	`</style>`
	(*Values* : length(px/cm/%),baseline(default) sub, super, top, bottom, middle, text-top, text-bottom)
		Sets the vertical alignment of an element

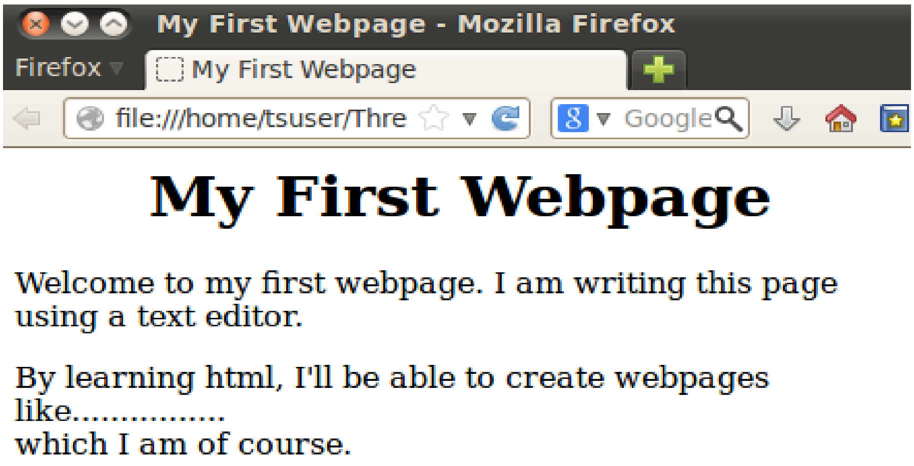
white-space	`p{white-space:nowrap;}` (*Values* : normal(default), nowrap, pre, pre-line, pre-wrap)	Specifies how white-space inside an element is handled
word-spacing	`p{word-spacing:30px;}` (*Values* : normal(default), length(px/pt/cm/em))	Increases or decreases the space between words in a text

Example:

Creating a sample webpage by putting all tags together tags in one single document.

```
<!DOCTYPE html>
    <html>
        <head>
            <meta charset="utf-8">
            <title> My First Webpage </title>
         <body>
             <h1 align="center">My First Webpage</h1>
             <p>Welcome to my first webpage. I am writing
                this page using a text editor.<!-- These comments
                doesn't affect the web presentation -->
             </p>
             <p>By learning html, I'll be able to create webpages
                      like...............<br>
                      which I am of course.
             </p>
        </body>
        </head>
    </html>
```

Output:



My First Webpage

Welcome to my first webpage. I am writing this page using a text editor.

By learning html, I'll be able to create webpages like................
which I am of course.

0.0.1 Difference Between Elements and Attributes

Elements	Attributes
represents structure or semantics	Defines a property for an element
consists of a start tag, content, and an end tag	Consists of an attribute / value pair and appears within element's start tag

Example:

$< p\ align = \text{“}right\text{”} >$ This is the content of the paragraph element. $< /p >$

$< p >$ ⇒ Element

align ⇒ Attribute

“right” ⇒ Value

Whats new in HTML5

HTML5 was created to make the coding process easier and more logical. The unique and impressive features HTML5 comes with are in the multimedia department. Many of the features it comes with have been created with the consideration that users should be able to run heavy content on low-powered devices. The syntactic features include the new $< video >$, $< audio >$ and $< canvas >$ elements, but also integration of vector graphics content (what we knew before as being the $< object >$ tags). This means that multimedia and graphic content on the web will be handled and executed easier and faster, without the need of plugins or APIs.

New Elements

Tag	Description
$< article >$	Defines an article in the document
$< aside >$	Defines content aside from the page content
$< audio >$	Defines sound, such as music or other audio streams.
$< video >$	Defines video, such as a movie clip or other video streams.
$< source >$	is used to specify multiple media resources for media elements, such as $< video >$ and $< audio >$.
$< bdi >$	Defines a part of text that might be formatted in a different direction from other text outside it
$< details >$	Defines additional details that the user can view or hide
$< dialog >$	Defines a dialog box or window
$< figcaption >$	Defines a caption for a $< figure >$ element
$< figure >$	Defines self-contained content, like illustrations, diagrams, photos, code listings, etc.
$< footer >$	Defines a footer for the document or a section
$< header >$	Defines a header for the document or a section
$< main >$	Defines the main content of a document
$< mark >$	Defines marked or highlighted text
$< menuitem >$	Defines a command/menu item that the user can invoke from a popup menu
$< meter >$	Defines a scalar measurement within a known range
$< nav >$	Defines navigation links in the document
$< progress >$	Defines the progress of a task
$< rp >$	Defines what to show in browsers that do not support ruby annotations
$< rt >$	Defines an explanation/pronunciation of characters (for East Asian typography)

Tag	Description
$< ruby >$	Defines a ruby annotation (for East Asian typography)
$< section >$	Defines a section in the document
$< summary >$	Defines a visible heading for a $< details >$ element
$< time >$	Defines a date/time
$< wbr >$	Defines a possible line-break

Basic Tags in HTML

Formatting Tags

The HTML tags are used to format the appearance of the text on your web page. This can jazz up the look of the web page, however, too much variety in the text formatting can also look displeasing.

Some of the text formatting tags are as follows:

Bold Text $< b >$ .... $< /b >$

The text in between the tags will be bold, and stand out against text around it, the same as in a word processor.

Example:

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

Result:

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

Italic Text $< i >$ .... $< /i >$

Also working the same way as a word processor, italics displays the text at a slight angle.

Example:

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

Result:

The following word uses a *italicized* typeface.

Underline Text $< u >$ ...... $< /u >$

Again, the same as underline in a word processor. Note that html links are already underlined and don't need the extra tag. Also $< ins >$ tag is also used similar to this.

Example:

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

Result:

The following word uses a underlined typeface.

Strike-out Text $< strike >$ ...... $< /strike >$

Puts a line right through the centre of the text, crossing it out. Often used to show that text is old and no longer relevant. Also works by using $< s >$ ...... $< /s >$ instead. Also to delete text $< del >$ works as same.

Example:

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

Result:

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

Preformatted Text $< pre >$ ...... $< /pre >$

Any text between the pre tags, including spaces, carriage returns and punctuation, will appear in the browser as it would in a text editor (normally browsers ignore multiple spaces)

Example:

```
< pre >
    function testFunction( strText ){
        alert (strText);
    }
< /pre >
```

Result:

```
function testFunction( strText ){
     alert (strText)
}
```

Emphasis $< em >$ ..... $< /em >$

Used to emphasize text, which usually appears in italics, but can vary according to your browser.

Example:

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

Result:

The following word uses a *emphasis* typeface.

Strong Emphasis $< strong >$ ..... $< /strong >$

Used to emphasize text more, which usually appears in bold, but can vary according to your browser.

Example:

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

Result:

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

Smaller Text $< small >$ ...... $< /small >$

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

Example:

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

Result:

The following word uses a small typeface.

Larger Text $< big >$ ..... $< /big >$

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

Example:

$< p >$The following word uses a $< big >$big$< /big >$ typeface.$< /p >$ This will produce following result:

Result:

The following word uses a big typeface.

Superscript Text $< sup >$ ..... $< /sup >$

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 characters height above the other characters.

Example:

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

Result:

The following word uses a ^{superscript} typeface.

Subscript Text $< sub >$ ..... $< /sub >$

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 characters height beneath the other characters.

Example:

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

Result:

The following word uses a $_{subscript}$ typeface.

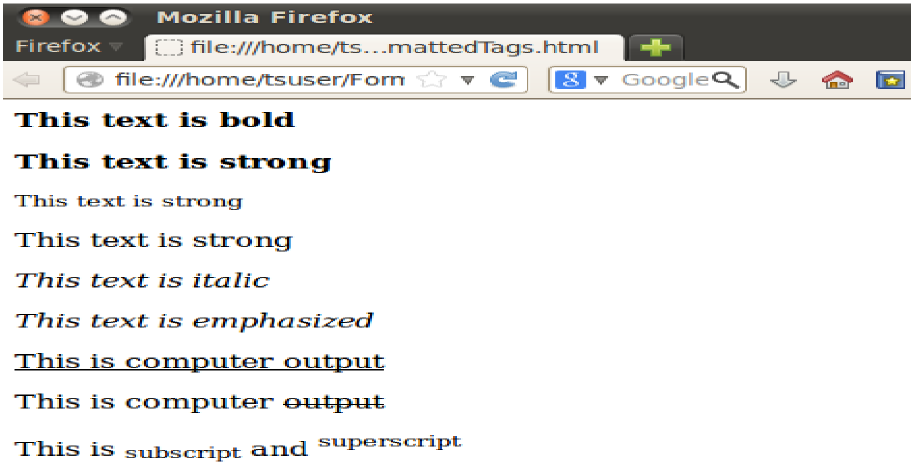
Tags	Description	Formatted Text
$< b >$	Bold Text	**bold**
$< i >$	Italic Text	*italicized*
$< u >$	Underline Text	underlined
$< strike >$	Strike-out Text	~~strikethrough~~
$< em >$	Emphasis	*emphasis*
$< strong >$	Strong Emphasis	**strong emphasis**
$< small >$	Smaller Text	small
$< large >$	Larger Text	large
$< sup >$	Superscript	$25^{th}Aug$
$< sub >$	Subscript	H_2O

Example:

Simple example of HTML document using formatted tags.

```
<!DOCTYPE HTML>
    <html>
        <body>
            <p><b>This text is bold</b></p>
            <p><strong>This text is strong</strong></p>
            <p><i>This text is italic</i></p>
            <p><em>This text is emphasized</em></p>
            <p><u>This is computer output</u></p>
            <p>This is<sub> subscript</sub>
                     and <sup>superscript</sup></p>
        </body>
    </html>
```

Output:



```
This text is bold
This text is strong
This text is strong
This text is strong
This text is italic
This text is emphasized
This is computer output
This is computer output
This is subscript and superscript
```

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.

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.

Attribute	Value	Description
Size	number	Specifies the size of text
face	font_family	Specifies the font of text
color	rgb(x,x,x)#xxxxxx colorname	Specifies the color of text

Attributes

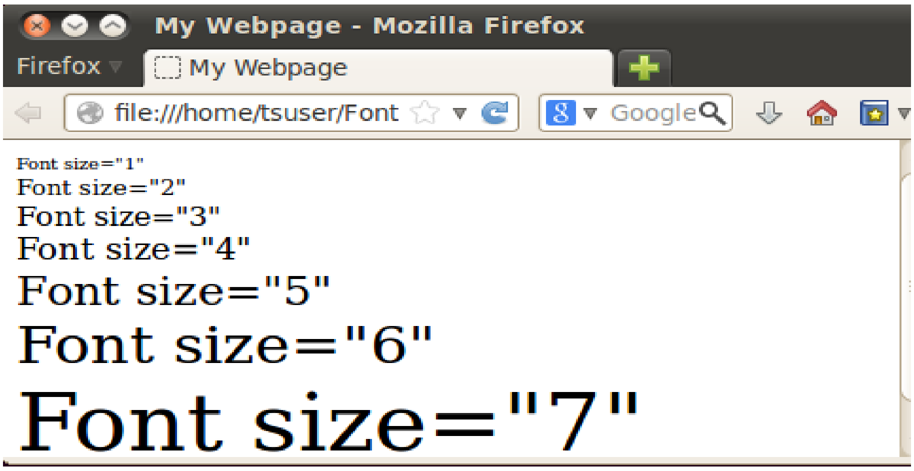
Font Size

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:

```
<!DOCTYPE HTML>
    <html>
        <head>
            <title>My Webpage</title>
        </head>
        <body>
            <font size="1">Font size="1"</font></br>
            <font size="2">Font size="2"</font></br>
            <font size="3">Font size="3"</font></br>
            <font size="4">Font size="4"</font></br>
            <font size="5">Font size="5"</font></br>
            <font size="6">Font size="6"</font></br>
            <font size="7">Font size="7"</font>
        </body>
    </html>
```

Output:



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

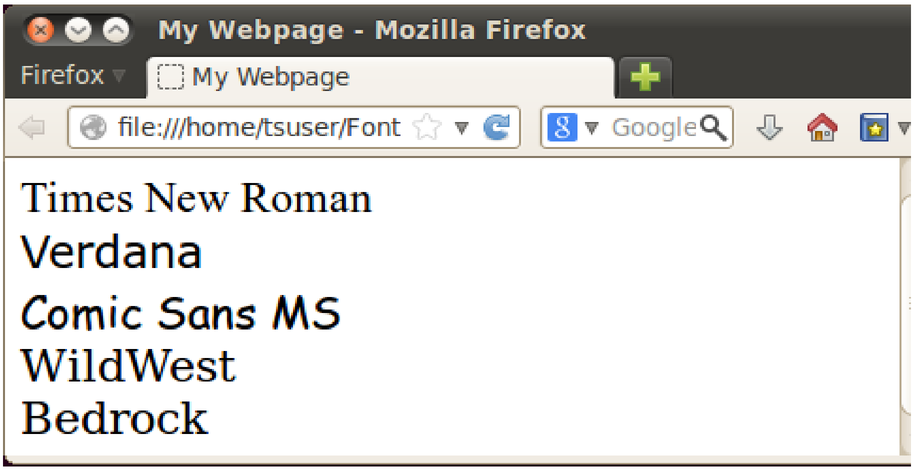
Font Face

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:

```
<!DOCTYPE HTML>
    <html>
        <head>
            <title>My Webpage</title>
        </head>
        <body>
            <font face="Times New Roman" size="5">
                Times New Roman
            </font></br>
            <font face="Verdana" size="5">
                Verdana
            </font></br>
            <font face="Comic sans MS" size="5">
                Comic Sans MS
            </font></br>
            <font face="WildWest" size="5">
                WildWest</font></br>
            <font face="Bedrock" size="5">
                Bedrock
            </font>
        </body>
    </html>
```

Output:



Times New Roman
Verdana
Comic Sans MS
WildWest
Bedrock

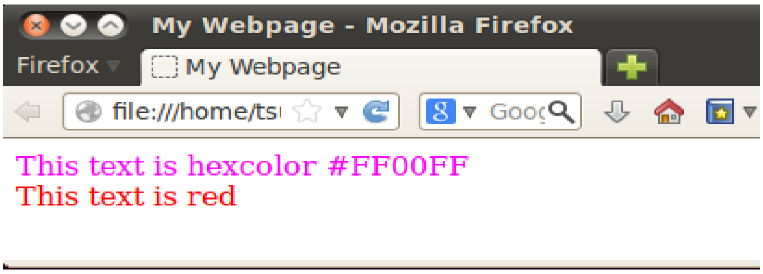
Font Color

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 Color Name with Codes.

Example:

```
<!Doctype HTML>
    <html>
        <head>
            <title>My Webpage</title>
        </head>
        <body>
            <font color="#FF00FF">
                This text is hexcolor #FF00FF
            </font><br />
            <font color="red">
                This text is red
            </font>
        </body>
    </html>
```

Output:



This text is hexcolor #FF00FF
This text is red

bgcolor Attribute

- The bgcolor attribute specifies the background color of a document.
- It is an attribute for $< body >$

Syntax:

$< body\ bgcolor = "color\_name|hex\_number|rgb\_number" >$

Attributes Values

Table showing attribute values:

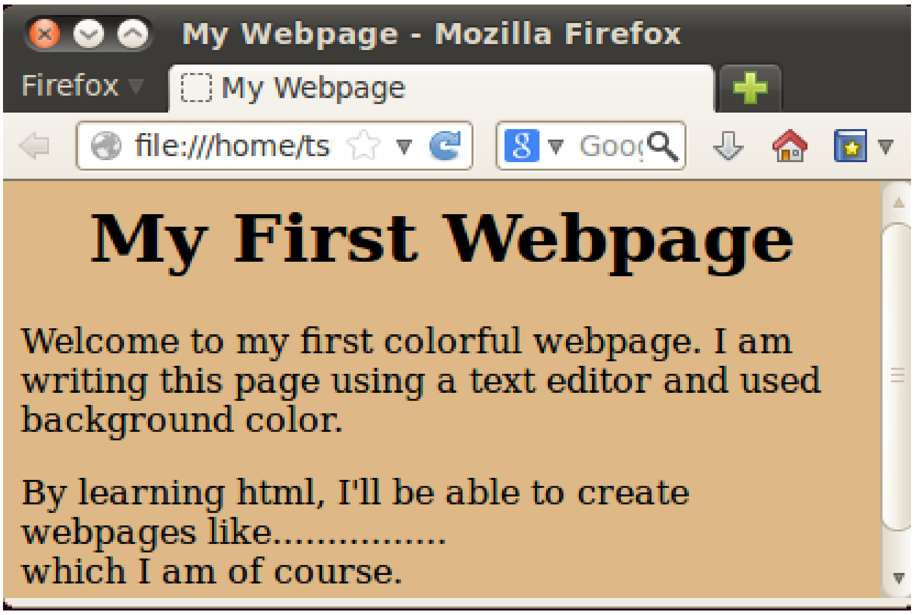
Value	Description
color_name	Specifies the background color with a color name (like "red")
hex_number	Specifies the background color with a hex code (like "#ff0000")
rgb_number	Specifies the background color with an rgb code (like "rgb(255,0,0)")

Example:

```
<!DOCTYPE HTML>
    <html>
        <head>
            <title>My Webpage</title>
        </head>
```

```
              <body bgcolor="#DEB887">
                  <h1 align="center">
                      My First Webpage
                  </h1>
                  <p>
                      Welcome to my first colorful webpage.
                      I am writing this page using a text editor
                      and used background color.
                      <!-- These comments doesn't affect the web
                                                  presentation -->
                  </p>
                  <p>
                      By learning html, I'll be able to create webpages
                          like................<br>
                          which I am of course.
                  </p>
              </body>
          </html>
```

Output:



My First Webpage

Welcome to my first colorful webpage. I am writing this page using a text editor and used background color.

By learning html, I'll be able to create webpages like................
which I am of course.

Header Tag $< header >$

- $< header >$ tag specifies a header for a document or section.
- used as a container for introductory content or set of navigational links.
- multiple $< header >$ elements are allowed

Footer Tag $< footer >$

- $< footer >$ tag defines a footer for a document or section.
- contains the author of the document, copyright information, links to terms of use, contact information...
- multiple $< footer >$ elements are allowed

Lists

HTML lists appear in web browsers as bulleted lines of text. There are actually three different types of HTML lists,They are as follows,

1. Unordered lists (bullets)
2. Ordered lists (numbers)
3. Definition lists (think: dictionaries).

Each list type utilizes its own unique list tag, which we'll demonstrate below.

Unordered lists

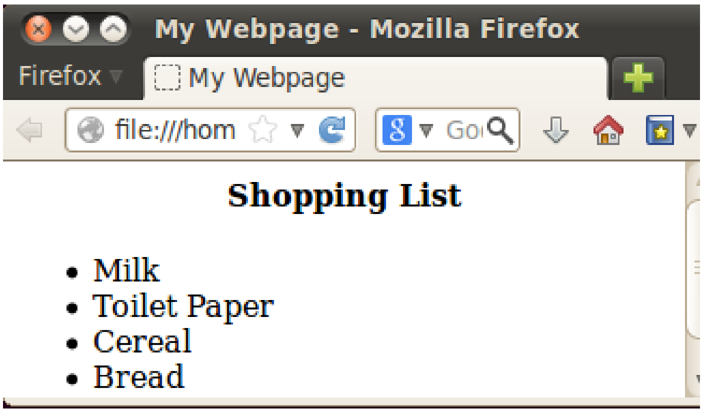
An unordered list ($< ul >$) signifies to a web browser that all list items contained inside the $< ul >$ and$< \backslash ul >$ tag should be rendered with a bullet preceding the text. The default bullet type for most web browsers is a full disc (black circle), but this can be adjusted using an HTML attribute called *"type"*.

Default Bullet List:

```
<!DOCTYPE HTML>
    <html>
        <head>
            <title>My Webpage</title>
        </head>
        <body>
            <h4 align="center">Shopping List</h4>
```

```
                <ul>
                    <li>Milk</li>
                    <li>Toilet Paper</li>
                    <li>Cereal</li>
                    <li>Bread</li>
                </ul>
            </body>
        </html>
```

Output:



Shopping List

- Milk
- Toilet Paper
- Cereal
- Bread

To render a list with a different bullet type, add a type attribute to the unordered list element. Using the same code in the example above, replace the $< ul >$ line from the previous example with any of the lines listed below to change the bullet from disc shape to another shape.

Values for type attributes:

<ul type = "square"> - A square hollow bullet

<ul type = "disc"> - A solid bullet (the default)

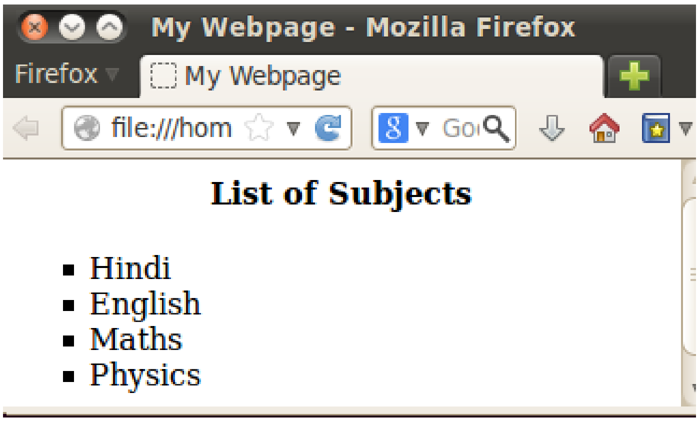
<ul type = "circle"> - A hollow bullet

Example: <ul type = "square">

```
<!DOCTYPE HTML>
    <html>
        <head>
            <title>My Webpage</title>
        </head>
```

```
                <body>
                    <h4 align = "center">List of Subjects</h4>
                    <ul type = "square">
                        <li>Hindi</li>
                        <li>English</li>
                        <li>Maths</li>
                        <li>Physics</li>
                    </ul>
                </body>
            </html>
```

Output:



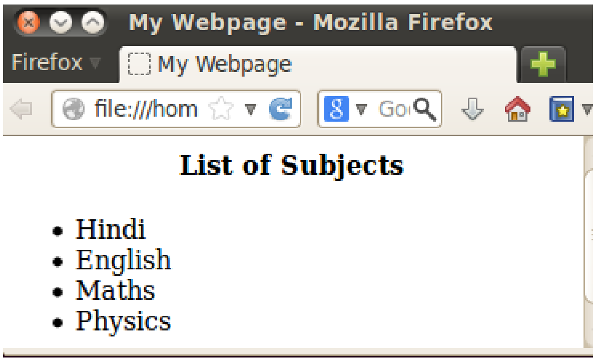
List of Subjects

- Hindi
- English
- Maths
- Physics

Example: <ul type = "disc">

```
        <!DOCTYPE HTML>
            <html>
                <head>
                    <title>My Webpage</title>
                </head>
                <body>
                    <h4 align = "center">List of Subjects</h4>
                    <ul type = "disc">
                        <li>Hindi</li>
                        <li>English</li>
                        <li>Maths</li>
                        <li>Physics</li>
                    </ul>
                </body>
            </html>
```

Output:



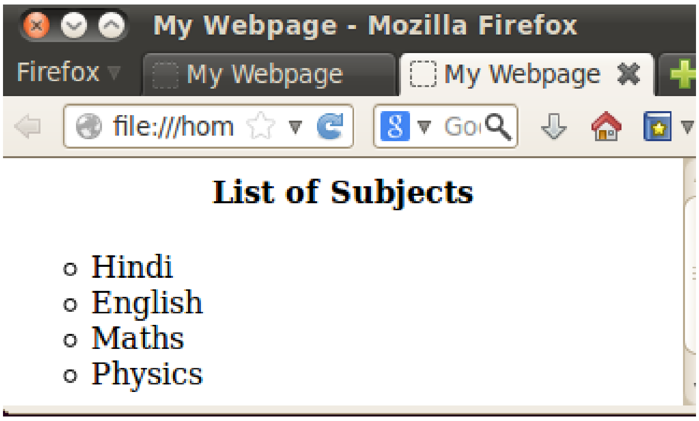
List of Subjects

- Hindi
- English
- Maths
- Physics

Example: <ul type = "circle">

```
<!DOCTYPE HTML>
    <html>
        <head>
            <title>My Webpage</title>
        </head>
        <body>
            <h4 align = "center">List of Subjects</h4>
            <ul type = "circle">
                <li>Hindi</li>
                <li>English</li>
                <li>Maths</li>
                <li>Physics</li>
            </ul>
        </body>
    </html>
```

Output:



List of Subjects

- Hindi
- English
- Maths
- Physics

Ordered lists

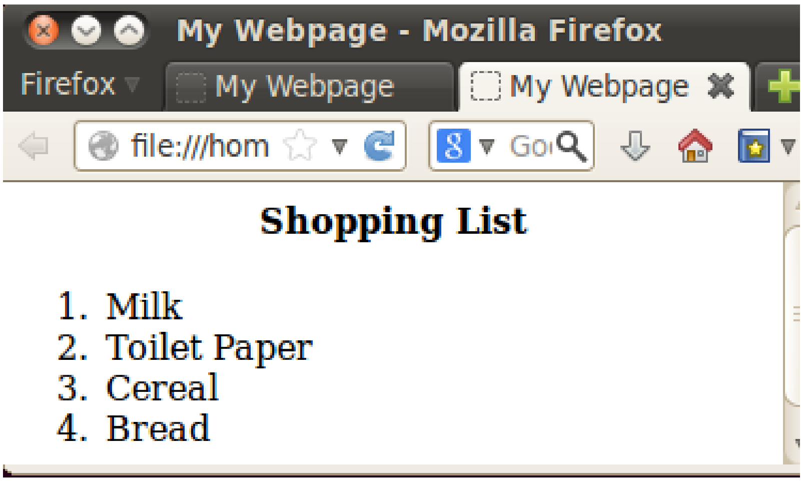
An ordered list is defined using the $< ol >$ tag, and list items placed inside of an ordered list are preceded with numbers instead of bullets.

The numbering of an HTML list can be changed to letters or Roman Numerals by once again adjusting the *type* attribute.

Default Number List:

```
<!DOCTYPE HTML>
    <html>
        <head>
            <title>My Webpage</title>
        </head>
        <body>
            <h4 align="center">Shopping List</h4>
            <ol>
                <li>Milk</li>
                <li>Toilet Paper</li>
                <li>Cereal</li>
                <li>Bread</li>
            </ol>
        </body>
    </html>
```

Output:



Shopping List

1. Milk
2. Toilet Paper
3. Cereal
4. Bread

Values of type Attribute:

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

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

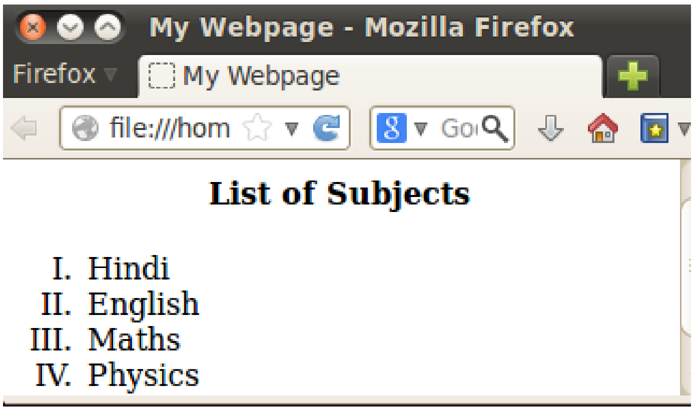
<ol type="a"> - Lower-Case Letters.

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

Example: <ol type = "I">

```
<!DOCTYPE HTML>
    <html>
        <head>
            <title>My Webpage</title>
        </head>
        <body>
            <h4 align = "center">List of Subjects</h4>
            <ol type = "I">
                <li>Hindi</li>
                <li>English</li>
                <li>Maths</li>
                <li>Physics</li>
            </ol>
        </body>
    </html>
```

Output:



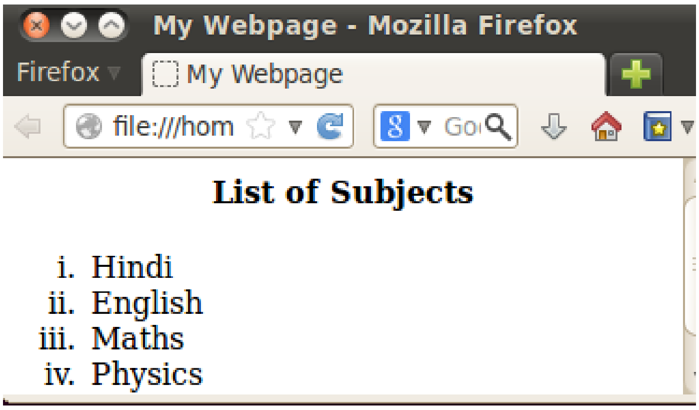
List of Subjects

I. Hindi
II. English
III. Maths
IV. Physics

Example: <ol type = "i">

```
<!DOCTYPE HTML>
    <html>
        <head>
            <title>My Webpage</title>
        </head>
        <body>
            <h4 align = "center">List of Subjects</h4>
            <ol type = "i">
                <li>Hindi</li>
                <li>English</li>
                <li>Maths</li>
                <li>Physics</li>
            </ol>
        </body>
    </html>
```

Output:



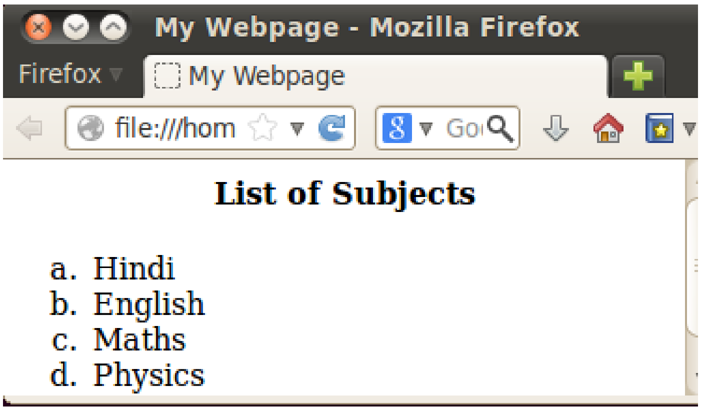
List of Subjects

i. Hindi
ii. English
iii. Maths
iv. Physics

Example: <ol type = "a">

```
<!DOCTYPE HTML>
    <html>
        <head>
            <title>My Webpage</title>
        </head>
        <body>
            <h4 align = "center">List of Subjects</h4>
            <ol type = "a">
                <li>Hindi</li>
                <li>English</li>
                <li>Maths</li>
                <li>Physics</li>
            </ol>
        </body>
    </html>
```

Output:



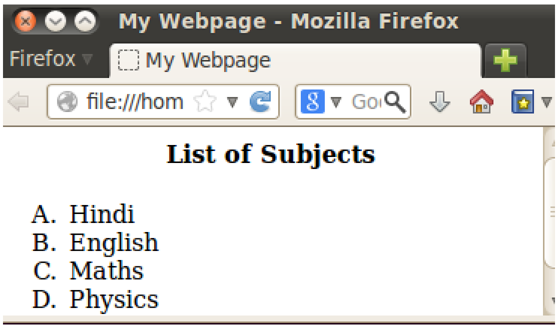
List of Subjects

a. Hindi
b. English
c. Maths
d. Physics

Example: <ol type = "A">

```
<!DOCTYPE HTML>
    <html>
        <head>
            <title>My Webpage</title>
        </head>
        <body>
            <h4 align = "center">List of Subjects</h4>
            <ol type = "A">
                <li>Hindi</li>
                <li>English</li>
                <li>Maths</li>
                <li>Physics</li>
            </ol>
        </body>
    </html>
```

Output:



List of Subjects

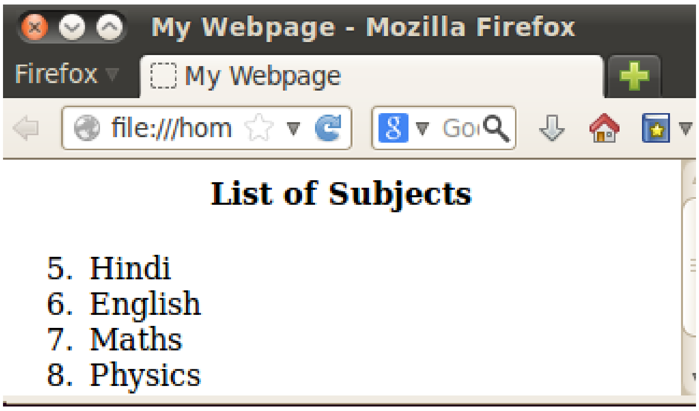
A. Hindi
B. English
C. Maths
D. Physics

The *start* attribute allows you to further customize an HTML ordered list by setting a new starting digit for the ordered list element.

Example:

```
<!DOCTYPE HTML>
    <html>
        <head>
            <title>My Webpage</title>
        </head>
        <body>
            <h4 align = "center">List of Subjects</h4>
            <ol start = "5">
                <li>Hindi</li>
                <li>English</li>
                <li>Maths</li>
                <li>Physics</li>
            </ol>
        </body>
    </html>
```

Output:



List of Subjects

5. Hindi
6. English
7. Maths
8. Physics

Definition term lists

HTML definition lists ($< dl >$) are list elements that have a unique array of tags and elements; the resulting listings are similar to those you'd see in a dictionary.These lists are important because they provide a context for the text with which they are associated.

$< dl >$ - opening clause that defines the start of the list

$< dt >$ - list item that defines the definition term

$< dd >$ - definition of the list item

DL $< dl >$

A definition list is the container element for DT($< dt >$) and DD($< dd >$) elements. The DL($< dl >$ element should be used when you want incorporate a definition of a term in your document, it is often used in glossaries to define many terms, it is also used in normal documents when the author wishes to explain a term in a more detail (Like this definition).

DT $< dt >$

The term currently being defined in the definition list. This element contains inline data.

DD $< dd >$

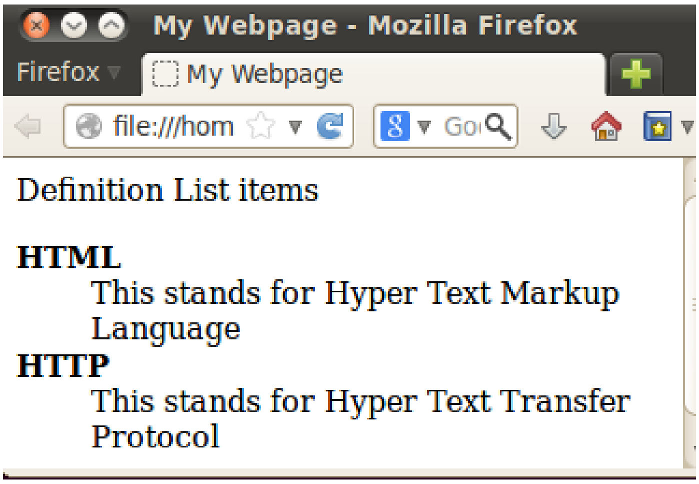
The definition description element contains data that describes a definition term. This data may be either inline, or it may be block level.

These lists displace the word term ($< dt >$) in such a way that it rests just above the definition element ($< dd >$) to offer a very unique look. For best, results we suggest bolding the definition terms with the bold tag $< b >$.

Example:

```
<!DOCTYPE HTML>
    <html>
        <head>
            <title>My Webpage</title>
        </head>
        <body>
            <p>Definition List items </p>
            <dl>
                <dt>
                    <b>
                        HTML
                    </b>
                </dt>
                    <dd>
                        This stands for
                        Hyper Text Markup Language
                    </dd>
                <dt>
                    <b>
                        HTTP
                    </b>
                </dt>
                    <dd>
                        This stands for
                        Hyper Text Transfer Protocol
                    </dd>
            </dl>
        </body>
        </html>
```

Output:



Definition List items

HTML
This stands for Hyper Text Markup Language
HTTP
This stands for Hyper Text Transfer Protocol

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.

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

It 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.

From this 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).

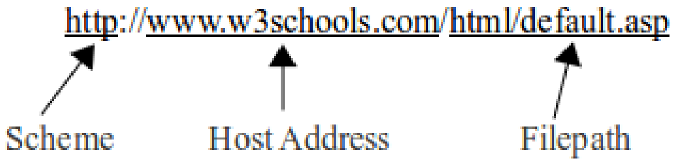
URL

Every document on the Web has a unique address. This address is known as **U**niform **R**esource **L**ocator (URL).

URL Elements

A URL is made of up several parts, each of which offers information to the Web browser to help find the page you are after. It is easier to learn the parts of a URL if you look at the most common ones first. If you look at the example URL given below, there are three key parts: the scheme, the host address, and the file path. The following sections discuss each of these in turn.

Example:



The Scheme The scheme identifies the type of protocol and URL you are linking to and therefore how the resource should be retrieved. For example, most Web pages use something called the *Hypertext Transfer Protocol (HTTP)* to pass information to you which is why most Web pages start with *http://*.

There are other schemes available and you can use based on your requirement:

Description:

http:// Hypertext Transfer Protocol (HTTP) is used to request pages from Web servers and send them back from Web servers to browsers.

https:// Secure Hypertext Transfer Protocol (HTTPS) encrypts the data sent between the browser and the Web server using a digital certificate.

ftp:// File Transfer Protocol is another method for transferring files on the Web. While HTTP is a lot more popular for viewing Web sites because of its integration with browsers, FTP is still commonly used protocol to transfer large files across the Web and to upload source files to your Web server.

file:// Used to indicate that a file is on the local hard disk or a shared directory on a LAN.

The Host Address

The host address is where a Web site can be found, either the IP address (four sets of numbers between 0 and 258, for example 68.178.157.132) or more commonly the domain name for a site such as *www.tutorialspoint.com*. Note that *"www"* is not actually part of the domain name although it is often used in the host address.it has nothing to do with the protocol used.

The Filepath

The filepath always begins with a forward slash character, and may consist of one or more directory or folder names. Each directory name is separated by forward slash characters and the filepath may end with a filename at the end. Here *index.htm* is the filename which is available in html directory.

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 $< \backslash a >$ tag becomes part of the link and a user can click that part to reach to the linked document.

Syntax:

$< a\ href = "DocumentURL" attr\_name = "attr\_value" ...more\ attributes/ >$

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.

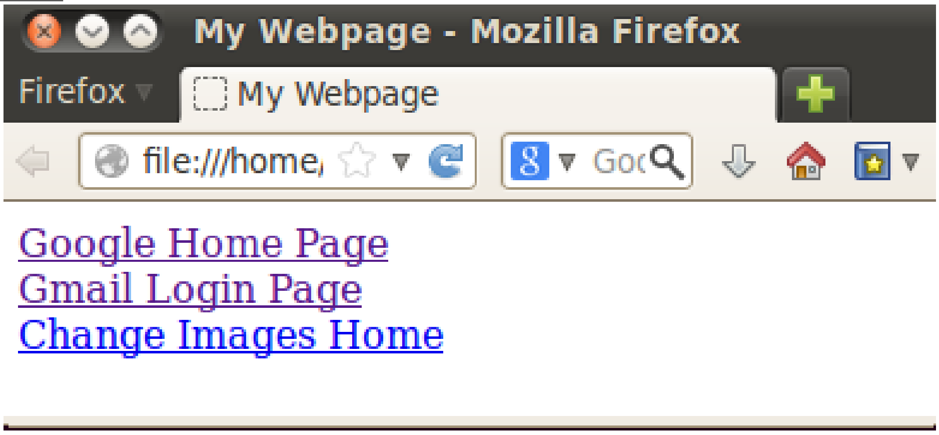
  For example, you could make the T key an access key so that when the user presses either the Alt or Ctrl key on his keyboard (depending on his operating system) along with the T key, the link gets activated.

Example:

```
<html>
     <head>
```

```
                <title>My Webpage</title>
        </head>
        <body>
                <a href="http://www.google.com/" target="_blank" >
                                                 Google Home Page</a></br>
                <a href="http://www.gmail.com/" target="_self" >
                                                 Gmail Login Page</a></br>
                <a href="http://www.change-images.com/" target="_top" >
                                                Change Images Home</a>
        </body>
</html>
```

Output:



Google Home Page
Gmail Login Page
Change Images Home

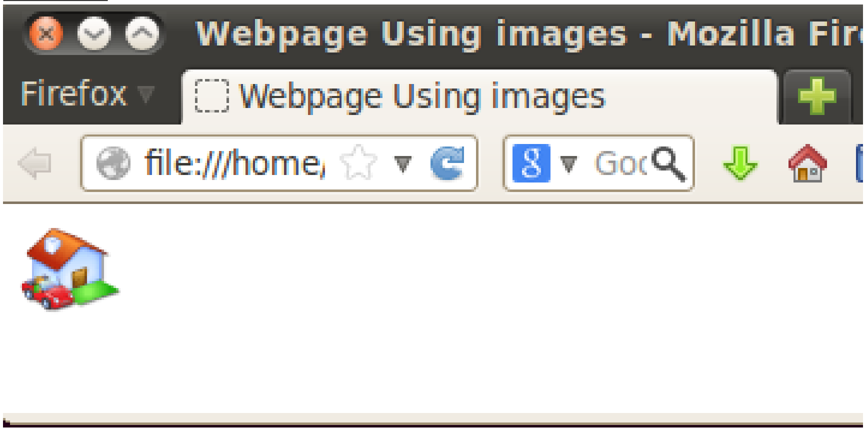
As now we know that how to create hyper text link using text and how to use images in your web page. Now we will learn how to use images to create hyper links.

Example:

```
<html>
        <head>
                <title>Webpage Using images</title>
        </head>
        <body>
                <a href="http://www.google.com/index.htm"
                                                target="_self" ></br>
                    <img src="/images/home.gif" alt="Google Home Page"
                                                       border="0"/>
                </a>
        </body>
    </html>
```

This will create following hyperlink at google.com home.

Output:



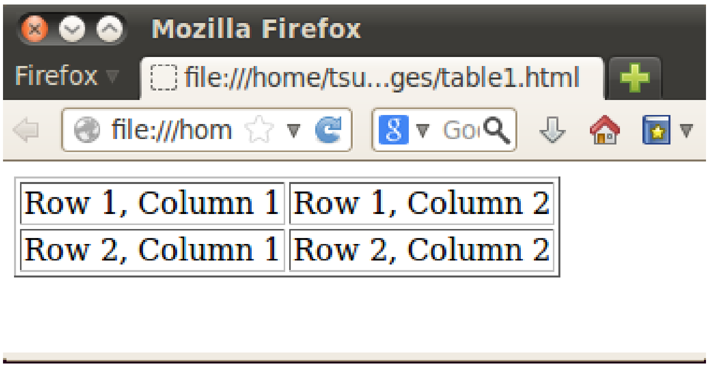
The HTML table model allows authors to arrange data – text, preformatted text, images, links, forms, form fields, other tables, etc. – into rows and columns of cells.

A simple table

Here is a trivial table of two rows and two columns:

```
    <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>
```

Output:



Row 1, Column 1	Row 1, Column 2
Row 2, Column 1	Row 2, Column 2

- The entire table is contained between $< table >$ and $< /table >$ tags.
- Each table row starts with a $< tr >$ tag.
- Each cell starts with a $< td >$ tag.

Table Heading

To add the column heading to the columns in the table, place the heading in $< th >$ and $< /th >$ tags.

Captioning a table

To add a title to a table, place the caption text between $< caption >$ and $< /caption >$ tags just after the $< table >$ tag.

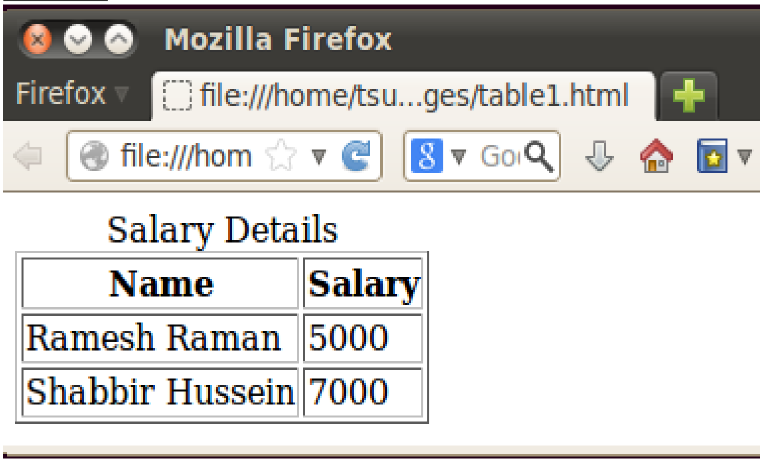
You can use an *align* = *"bottom"* attribute in the $< caption >$ tag to position the caption below the table. The default placement is at the top.

```
<caption align = "bottom"> Salary Details </caption>
```

Example:

```
<table border = "1">
    <caption> Salary Details </caption>
    <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>
```

Output:



Salary Details

Name	Salary
Ramesh Raman	5000
Shabbir Hussein	7000

Attributes of the $< table >$ tag

The $< table >$ tag allows a number of optional attributes that specify the properties of the entire table.

- <**table align = ”left”**> positions the table at the left side of the viewer window, *align* = *”center”* centers it, and *align* = *”right”* places it on the right side.
- The <**table width = ”dimension”**> attribute sets the width of the entire table, it is discussed above in the section on dimensions.
- The <**table border = ”dimension”**> tells the browser to draw a border around the entire table. The dimension is optional and specifies how thick to make the border. For example: $< tableborder = ”5px” >$
  specifies a five-pixel border around the table.

Aligning Text in Cells

Horizontal alignment of text in cells

Used to Justify the text in the cells, by default it justifies to left.
Any cell can include an align="align-type" attribute to tell the browser how to position the item in that cell.

The align-type can be any of:

- <**td align = "left"**> to left-justify the contents of the cell.
- <**td align = "center"**> to center the contents of the cell.
- <**td align = "right"**> to right-justify the contents of the cell.
- <**td align = "justify"**> to to justify the text; that is, if the item is a text string too long to fit on one line, all the lines but the last will be stretched to form straight left and right margins.
- You can use $< tdalign = "char" >$ for columns of numbers with decimals so that they are aligned on the decimal point.

Vertical alignment for a whole row

Used specify where the items are placed vertically in their cells by using a valign attribute in the $< tr >$ tag for that row.

The valign attribute can take these values:

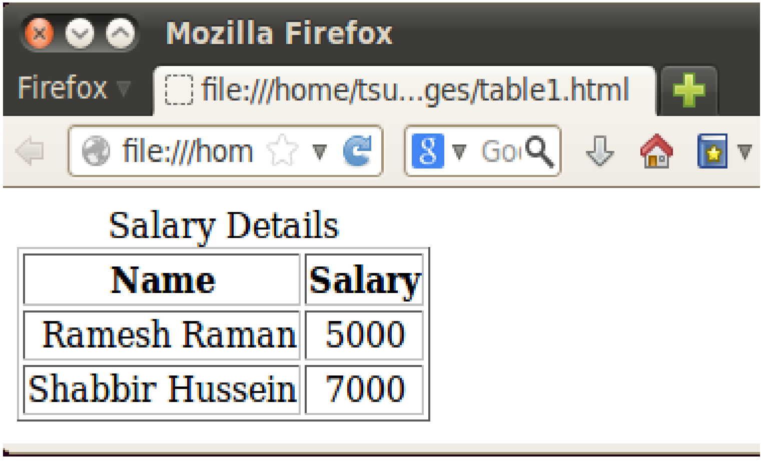
- <**tr valign = "top"**> says to put each item at the top of its cell.
- <**tr valign = "middle"**> says to center each item vertically within its cell.
- <**tr valign = "bottom"**> says to put each item at the bottom of its cell.

Example:

```
<table border = "1">
    <caption> Salary Details </caption>
    <tr>
        <th> Name </th>
```

```
            <th> Salary </th>
        </tr>
        <tr align = "center">
            <td align = "right"> Ramesh Raman </td>
            <td align = "center">5000</td>
        </tr>
        <tr align = "center">
            <td align = "right"> Shabbir Hussein </td>
            <td align = "center"> 7000</td>
        </tr>
    </table>
```

Output:



Salary Details

Name	Salary
Ramesh Raman	5000
Shabbir Hussein	7000

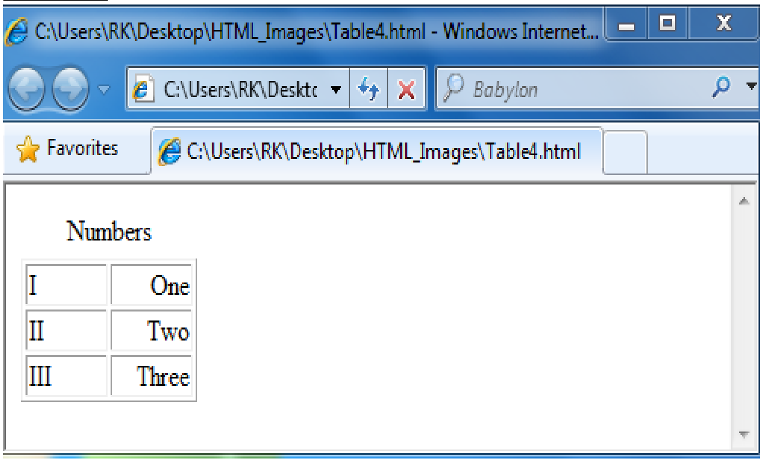
Alignment for an entire column

It is difficult to add an align = "..." attribute in each cell of a column. A better way is use a group of <**col**> tags before the first row of the table, each of these tags specifies the alignment for that whole column.

Example:

```
     <table border =  "1">
        <caption>Numbers</caption>
        <col align = "left" width = "50">
        <col align = "right" width = "50">
        <tr>
            <td> I </td>
            <td> One</td>
        </tr>
        <tr>
            <td> II </td>
            <td> Two</td>
        </tr>
        <tr>
            <td> III </td>
            <td> Three</td>
        </tr>
    </table>
```

Output:


C:\Users\RK\Desktop\HTML_Images\Table4.html - Windows Internet...
C:\Users\RK\Desktc
Babylon
Favorites
C:\Users\RK\Desktop\HTML_Images\Table4.html

Numbers

I	One
II	Two
III	Three

The first $< col >$ tag says that the first column's cells will be left-justified. The second $< col >$ tag says to right-justify the cells of the second column.

Column grouping

If you have groups of columns with the same attribute values, you can use a $< colgroup >$ tag before each group of $< col >$ tags to specify attributes for that group.

```
<table>
    <colgroup align = "center">
        <col width = "3pi">
        <col width = "5pi" span = "4">
    <colgroup align = "right">
```

```
            <col width = "4pi">
            <col width = "8pi">
             ...
    </table>
```

The example above defines seven columns. The first five are all centered and the next two are right-justified.

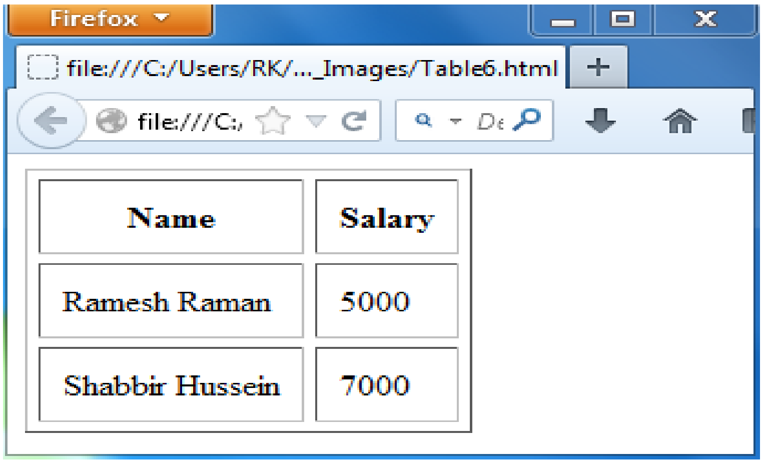
Cellpadding and Cellspacing

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.

Example:

```
    <table border = "1" cellpadding = "15" 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>
```

Output:



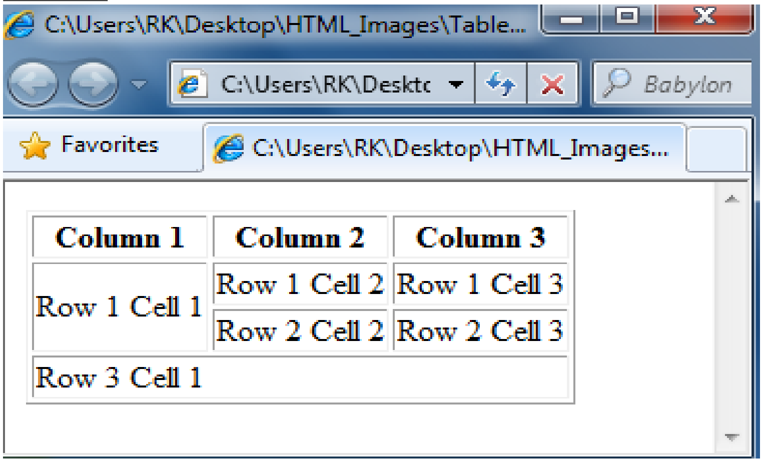
Name	Salary
Ramesh Raman	5000
Shabbir Hussein	7000

Colspan and Rowspan Attributes

Colspan attribute is used to merge two or more columns into a single column and rowspan attribute is used to merge two or more rows.

```
<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>
```

Output:


C:\Users\RK\Desktop\HTML_Images\Table...
C:\Users\RK\Desktc
Babylon
Favorites
C:\Users\RK\Desktop\HTML_Images...

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

Tables Backgrounds

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

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

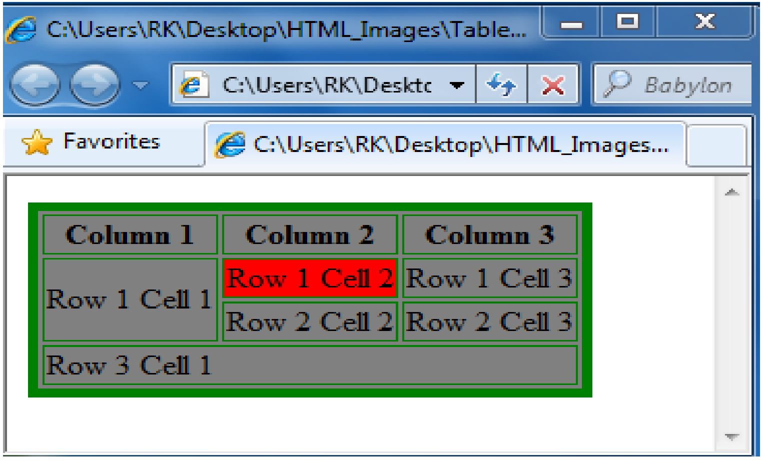
NOTE : You can set border color also using bordercolor attribute.

Example:

```
<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>
```

Output:


C:\Users\RK\Desktop\HTML_Images\Table...
C:\Users\RK\Desktc
Babylon
Favorites
C:\Users\RK\Desktop\HTML_Images...

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		

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.

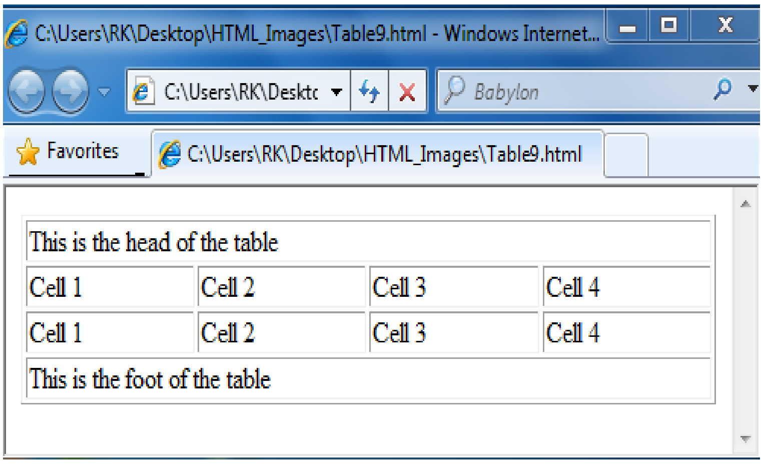
The three elements for separating the head, body, and foot of a table are:

- <**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>
```

Output:


C:\Users\RK\Desktop\HTML_Images\Table9.html - Windows Internet...
C:\Users\RK\Desktc
Babylon
Favorites
C:\Users\RK\Desktop\HTML_Images\Table9.html

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