

## HTML Tags learned

1. **<!DOCTYPE HTML PUBLIC "THIS IS AN EXAMPLE">** = Used to indicate what type of document it is. generally it must contain the "DOCTYPE" keyword.
2. **<HTML> ...</HTML>** - All html files start and end with the tag pair.
3. **<HEAD>...</HEAD>** - All html files have a pair of "HEAD" tags that indicate what the title and other attributes of the page are going to be.
4. **<TITLE>...</TITLE>** - This tag indicates what the title of the HTML file is going to be on the BROWSER window title.
5. **<BODY>...</BODY>** - This tag pair is to logically separate the HTML file into the header and the body. Usually the header contains information regarding the html whereas the body contains information that the html file must actually contain. The body attributes for changing the color and appearance of the hyperlinks is as follows.

**LINK attribute** : This attribute specifies the color of the hyperlink.  
**VLINK attribute** : This attribute specifies the color of a visited hyperlink.  
**ALINK attribute** : This attribute specifies the color of a active hyperlink.

6. **<P>...</P>** - This tag pair is used to indicate the paragraph. Any text that needs to be separated into a paragraph must be put in within a paragraph tag.
7. **<B>...</B>** - This tag pair is used to indicate the text within tag pair must be in bold letters.
8. **<I>...</I>** - This tag pair is used to indicate the text within the tag pair must be in italic letters.
9. **<IMG SRC="..\images\corp.gif" ALT="logo !" HEIGHT = "100" WIDTH = "100" >** - This tag is used to embed images in the html page. The SRC attribute is used to locate the file name under a directory, the ALT attribute is used to indicate the TOOLTIP message that must appear, and HEIGHT and WIDTH indicate the height and the width of the image that is being shown on the html page.

**BORDER attribute** : The border attribute can be used to specify whether a hyperlink colored border is required for the image or not. the possible values for the attribute are "0" or "1".

ISMAP attribute: This is used to specify SERVER SIDE image maps.

USEMAP attribute: This is used to specify CLIENT SIDE image maps.

10. The HTML template must look like,

```
<!DOCTYPE HTML PUBLIC "THIS IS AN EXAMPLE ">
<HTML>
  <HEAD>
    <TITLE> YOUR TITLE GOES HERE. </TITLE>
  </HEAD>
  <BODY> YOUR MAIN INFORMATION GOES HERE.
  </BODY>
</HTML>
```

11. **<HR NOSHADE>** - This tag is used to indicate the Horizontal ruling on the HTML page. If 3D effect of the horizontal bar is not required, then the NOSHADE attribute can be used.

12. **<H1 ALIGN="CENTER"> ...</H1>** - This pair of tags is used to indicate that the text must be the main title for the HTML page. The ALIGN attribute can be used to set the alignment to "center" or "left" or "right".

13. **<!-- YOUR COMMENT GOES HERE -->** - Comments in HTML document can be given like this.

14. **<H1>Heading1</H1>**  
**<H2>Heading2</H2>**  
**<H3>Heading3</H3>** - This set of tags will show the Headings in smaller fonts as the heading increases.

15. **ALIGN** - The align attribute can be used for headings as well. For <P>...</P> tags also, the ALIGN attribute can be used.

16. **<BR>** - Used to insert a carriage return in the HTML file. The attribute to be used for this is the CLEAR attribute.

17. **<CENTER>...</CENTER>** - To center the entire block of text this tags are used.

18. **<DIV>...</DIV>** - Similar to the <CENTER> tag, the <DIV> tag is also used to align the block of text in any fashion. **The use of <DIV>...</DIV> is preferred over <CENTER> tag.**

19. **<BLOCKQUOTE>...</BLOCKQUOTE>** - This tag is used to indent a large body of text on the HTML page.

20. **<PRE>...</PRE>** - These HTML tags are used when you want to preserve the formatting in a HTML block of text. The formatting could mean spaces, returns and alignment etc.

### 21. Lists:

a. **Ordered Lists:** **<OL>..</OL>** - Defines a list where order must matter.

Example:

```
<OL>
<LI>Item1
<LI>Item2
<LI>Item3
</OL>
```

**NOTE: "START" - Used to specify the offset to start with ordering.**

**&**

**"TYPE" - Used to specify the type that could be used for the ordering. It could be 'numerals' or 'letters' (caps/small) or 'roman' (caps/small), attributes can be used with the list tags.**

b. **Unordered Lists:** **<UL>...</UL>** - Unordered lists, the order does not matter.

Example:

```
<UL>
<LI>Unordered Lists
  <UL START="50" TYPE="square">
    <LI> Can be nested.
  </UL>
  <UL START="150" TYPE="disc">
    <LI> Can be nested like this.
  </UL>
```

```
<UL START="25" TYPE="circle">
  <LI>And can also be nested like this.
</UL>
```

```
</UL>
```

c. **Definition List:** This is used in conjunction with places where there is, the header and the details places.  
Example:

```
<DL>
  <DT> This is the TERM or Header
  </DT>
  <DD> This forms the body.
  </DD>
</DL>
```

d. **Directory** - **<DIR>** - This is supposed to give a directory structure kind of look.  
Example:

```
<DIR>
  <LI>dir1
  <LI>dir2
  <LI>dir3
</DIR>
```

e. **Menus** - **<MENU>** - This is similar to directory tag.  
Example:

```
<MENU>
  <LI>dir1
  <LI>dir2
  <LI>dir3
</MENU>
```

22. **<ADDRESS>** - This tag is used to specify address. This is a special tag.

23. **Attributes** -

<b>&lt;TT&gt;...&lt;/TT&gt;</b>	- Typewriter Monospaced.
<b>&lt;U&gt;...&lt;/U&gt;</b>	- Underlined
<b>&lt;STRIKE&gt;...&lt;/STRIKE&gt;</b>	- Strike through
<b>&lt;S&gt;...&lt;/S&gt;</b>	- Same as Strike through
<b>&lt;SUB&gt;...&lt;/SUB&gt;</b>	- Subscript
<b>&lt;SUP&gt;...&lt;/SUP&gt;</b>	- Superscript
<b>&lt;BIG&gt;...&lt;/BIG&gt;</b>	- Bigger Font ( than current)
<b>&lt;SMALL&gt;...&lt;/SMALL&gt;</b>	- Smaller Font ( than current).

24. **<MARQUEE>** - This is a MS-IE specific tag.

25. **<LAYER>** - This is a Netscape specific tag.

26. **<A>...</A>** - Anchor Tags. These tags are used for linking namely hyperlinking.  
Example:

```
<A HREF="http://www.ibm.com">Visit IBM Web page</A>
```

This will place a hyperlink on the page at the text "Visit IBM Web page" and the link will refer to the www.ibm.com website.

**HREF Attribute :** This attribute can be used to specify not only http hyperlinks but also, locations of directories, "ftp", "nntp", and even "mailto".

Example:

```
<A HREF="http://www.ibm.com/spec.html">Visit SPECS page</A>
```

```
<A HREF="http://www.ibm.com/java">Visit IBM-JAVA Web page</A>
```

```
<A HREF="ftp://ftp.ibm.com">Visit IBM FTP server.</A>
```

```
<A HREF="mailto:webmaster@ibm.com">Send a mail to IBM Web Master</A>
```

**NAME attribute** : Names the anchor element so that the target may be another anchor element.

**ID attribute** : Identifies the anchor for target by another anchor, style sheet access and scripting. After naming the anchor by using the "NAME" attribute, using the ID attribute we can use it to refer to the particular anchor element.

**TARGET attribute:** Defines the FRAME destination of the link.

**TITLE attribute:** Sets the hint text for the link.

**ACCESSKEY attribute:** Sets the key for keyboard access to the link. Preferably used where tet based browsers are used such as "LYNX".

**TABINDEX attribute** : Sets the numeric order of tab for the particular link by using the TAB key.

**REL attribute** : Defines the relationship of the object being linked to.

**REV attribute** : Defines the relationship of the current object to the object being linked to.

27. **Markers** `<A NAME="top"></A>` - These type of markers specify, unique locations on the page. Using these marker names along with "#" symbol the specific location in the document can be reached. Example1:

```
<A NAME="topmarker"></A>
```

```
....  
....  
....  
....
```

```
<A NAME="bottommarker" HREF="#topmarker"> Return to top marker position</A>
```

Example 2: Likewise, we can specify links to specific markers on all html pages that might be linked containing markers.

```
<A NAME="bottommarker" HREF="http://www.ibm.com/index.html#JAVA_marker"> Return to top marker position</A>
```

28. **Image Links** : The images could also be hyperlinked. By using the following tags shown below.

```
<A HREF="http://www.ibm.com/spec.html" TITLE="IBM Web site link" ACCESSKEY="B">  
<IMG SRC="..\images\corp.gif" ALT="Corporate Web page" HEIGHT="50" WIDTH="50">
```

Visit [CORP. webpage](#)

29. **Image Maps** : These are hyperlinked images. Clicking on a particular spot will load another html page. There are 2 types of image maps.

**a. Server-side Image Maps** : These are image maps that will contact the server for linking to a different URL. Features,

- The user clicks somewhere within the image.
- The browser sends a request to the server with the (x,y) co-ordinates. The data is decoded by the server and sends a URL which maps to the current spot clicked.
- After getting the information back from the server, the browser requests the particular URL.

**b. Client-side Image Maps** : There is no need to visit the server getting the URL that corresponding to the spot clicked on the image.

- There is no need to visit the server for determining the page requested.
- Destination URL's can be shown as the user moves over the image.
- Image maps can be developed easily without testing with a server.

Example:

```
<IMG SRC="..\images\corp.gif" USEMAP="#shapes" BORDER="0" WIDTH="100"
HEIGHT="150">
...
...
...
<!-- here we can have the image map information -->
<MAP NAME="shapes">
    <AREA SHAPE="RECT" COORDS="6,50,140,143" HREF="rect.html"
ALT="rectangle">
    <AREA SHAPE="CIRCLE" COORDS="195,100,50" HREF="cir.html" ALT="circle">
    <AREA SHAPE="POLY" COORDS="6,50,140,143,123,456" HREF="poly.html"
ALT="polygon">
</MAP>
```

30. **<LINK> tag** - Most often, the usage of `<A>...</A>` tag that is an anchor. Using this we can develop links to other pages. But often there is no information on the relationship of the current document and the destination document. In order to provide information on the relationship between the current document and the destination document, the `<LINK>` attribute has come into being.

Example:

```
<LINK HREF="http://www.bigblue.com/index.html" REL="INDEX">
```

**REL attribute : has alot of significance.** Possible REL value are,

- |          |   |
|----------|---|
| TOP      | = The page that is similar to index of your site.                               |
|          | <code>&lt;LINK HREF="http://www.bigblue.com/index.html" REL="TOP"&gt;</code>    |
| CONTENTS | = The page that acts as the "Table of Contents".                                |
|          | <code>&lt;LINK HREF="http://www.bigblue.com/toc.html" REL="CONTENTS"&gt;</code> |
| INDEX    | = The page that acts as the index for the site.                                 |
|          | <code>&lt;LINK HREF="http://www.bigblue.com/index.html" REL="INDEX"&gt;</code>  |
| GLOSSARY | =The page that acts as the glossary page.                                       |

<LINK HREF="http://www.bigblue.com/index.html" REL="GLOSSARY">	
COPYRIGHT	=The page that acts as the copyright page.
NEXT	=The page that refers to the next document in the list.
PREVIOUS	=The page that refers to the previous document in the list.
HELP	=The page that refers to the HELP page in the list.
SEARCH	=The page that refers to the SEARCH page in the list.
NAVIGATE	=The doc. that contains information of how to navigate the site.
CHILD	=The doc. that is the child doc. to the doc. being viewed.
SIBLING	=This identifies the do. under the same hierarchy.
PARENT	=This is the parent relationship doc.
BEGIN/FIRST	=Identifies the sequence of the document.
END/LAST	=Identifies the sequence as the last of the docs.
BIBLIOENTRY	=
DEFINITION	=
FOOTNOTE	=
MADE	=Used to identify the maker/author of the documents.
AUTHOR	=Doc. on how to contact the author.
EDITOR	=Link to the editor's home/personal home page.
PUBLISHER	=The company who is publishing the pages.
DISCLAIMER	=The disclaimer that identifies the page for the "disclaimer note".
TRADEMARK	=The page containing the Trademark info.
META	=This contains info. about info. contained in this site.
TRANSLATION	= This contains the relationship that has the translation of the document(s).
STYLESHEET	=It identifies the stylesheet for the current document.

31. **<META> tag** - This tag is to present information on information of the document.

Example:

```
<META NAME="author" CONTENT="Venkatesan C">  
<META NAME="Description" CONTENT="Is Learning HTML">
```

32. **Client-PULL Concept**: This allows to reload a particular page after a definite amount of time. This concept is known as client-pull.

Example:

```
<META HTTP-EQUIV="REFRESH" CONTENT="10;URL=secondpage.html">
```

This specifies to download the page named "secondpage.html" after 10 seconds.

33. **Images Basics**: Image Tag is used to embed images in the html document. The general syntax is,

```
<IMG SRC="logo.gif">
```

**ALT Attribute**: This attribute is used to set the alternative text in the place of images on the browser that is text based or images are turned off. It could be used to display help as a tooltip also.

**ALIGN attribute**: This attribute is used to set the alignment of the image against the text.

```
<IMG SRC="logo1.gif" ALIGN="TOP" BORDER="1" >
```

```
<IMG SRC="logo2.gif" ALIGN="MIDDLE" BORDER="1" >
```

```
<IMG SRC="logo3.gif" ALIGN="BOTTOM" BORDER="1" >
```

```
<IMG SRC="logo4.gif" ALIGN="LEFT" BORDER="1" >
```

<IMG SRC="logo5.gif" ALIGN="RIGHT" BORDER="1" >

Other ALIGN attributes that can take values are TEXTOP, BASELINE, ABSMIDDLE, ABSBOTTOM. Try to avoid these tags.

**HSPACE & VSPACE attribute** : These are provided to introduce spaces around images to the extent we want. The horizontal space between the image and the wrap around text is specified by the HSPACE and similiarly for the vertical spacing we use the VSPACE.

**HEIGHT & WIDTH attributes** : These are used to set the visible height and the width of the image that we want to display.

**LOWSRC attribute**: This is used to load a low content image before the actual say, hig resolution image is loaded.

Example :

<IMG SRC="hi-res-img.gif" LOWSRC="b\_w\_img.gif" HEIGHT = "100" WIDTH="100"  
ALT="Image photo">

34. **Images as BUTTONS** : Inorder to make the images as buttons, we have to enclose them within the <A>...</A> anchor tag.

<A HREF="http://www.ibm.com"><IMG SRC="logo.gif"></A>

To turn off the border in this case is simply to set the BORDER attribute to 0.

<A HREF="http://www.ibm.com"><IMG SRC="logo.gif" BORDER="0"></A>

35. **<NOBR>... </NOBR> Tag** : This specifies no word wrapping must take place.

Example.

<NOBR><H1> This line of text is not meant to be broken, This line of text is not meant to be broken , This line of text is not meant to be broken , This line of text is not meant to be broken</H1></NOBR>

This mean that the entire line will be shown in a single line by the browser with a horizontal scroll bar.

36. **<SPACER> Tag** : This is used to produce spacing between/within lines typically.

Example:

<H1>This is a line of text, <SPACER TYPE="HORIZONTAL" SIZE="75"> and this is the end of it.</H1>

This will produce a line with about 75 pixels spacing.

Syntax :

<SPACER>

Attributes:

ALIGN = LEFT | RIGHT | TOP | ABSMIDDLE | ABSBOTTOM | TEXTTOP | MIDDLE |  
BASELINE | BOTTOM (use with block)

HEIGHT = # of pixels. (use with block)

SIZE = # of pixels. (use with HORIZONTAL, VERTICAL ALIGNMENT only)

TYPE = HORIZONTAL | VERTICAL | BLOCK

WIDTH = # of pixels. (use with block)

37. **<MULTICOL> Tag:** This is used typically to produce the multi column page like a newspaper.  
Example:

```
<MULTICOL COLS="2" GUTTER="50" WIDTH="80%">  
This is a full fledged article on the global warming scenario. It is  
reported that almost the earth warms up be a fraction of a degree  
centigrade due to the pollution and the change in the eco-system. The  
"Al-Nino" caused a lot of damage to the Florida coast of the US. This is  
due to "Al-Nino". There is a complement of "Al-Nino" called the "Al-  
Niniya".  
</MULTICOL>
```

**Syntax:** <MULTICOL>

Attributes:

COLS = Number of columns.

GUTTER = gutter width in pixels or %. Gutter is the space between 2 column.

WIDTH = column width in pixels or %.

38. **Tables <TABLE> ...</TABLE>:** This is used to specify the table typeof layout in the HTML document.

Example:

```
<TABLE BORDER="1">  
<CAPTION> This is the CAPTION of the TABLE</CAPTION>  
<TR>  
  <TH>Car</TH>  
  <TH>Company</TH>  
</TR>  
<TR>  
  <TD>Camry</TD>  
  <TD>Toyota</TD>  
</TR>  
<TR>  
  <TD>Taurus</TD>  
  <TD>Ford</TD>  
</TR>  
<TR>  
  <TD>Concorde</TD>  
  <TD>Chrysler</TD>  
</TR>  
</TABLE>
```

**Syntax:** <TABLE> ...</TABLE>

Attributes:

BORDER=Width of the border in pixels.

BACKGROUND = URL of image.

BGCOLOR = color

ALIGN = LEFT | RIGHT | CENTER

BORDERCOLOR = color

BORDERCOLORDARK = color

BORDERCOLORLIGHT = color

CELLBORDER = pixels

CELLSPACING = pixels

CELLPADDING = pixels

CLASS = name of style sheet access

COLS = number of columns.  
DATAPAGESIZE = n (IE)  
DATASRC = #ID (IE)  
DIR = text direction  
FRAME = ABOVE | BELOW | BORDER | BOX | INSIDES | LHS | RHS | VOID |

VSIDES

HEIGHT = pixels or %.  
HREF = url to reference.  
ID = unique value.  
LANG = ISo Lang. Abbr.  
RULES = ALL | COLS | GROUPS | NONE | ROWS ( these specify dividing lines  
within table).

STYLE = style sheet properties.  
TITLE advisory text  
TRANSPERANCY = 0 - 100.  
WIDTH = pixels or %.

...

Table Elements like <CAPTION>, <TR>,<TD> etc.

Groups of rows may contain these tag elements. They are,

<THEAD> - Header for the group.  
<TFOOT> - Footer for the group.  
<TBODY> - Body for the group.

39. **Fonts : The <FONT>...</FONT>** tag is used to specify text in a particular font.  
Example:

**<FONT SIZE="10">**This is a line of text with size 10**</FONT>**

Syntax: <FONT>...</FONT>

Attributes:

COLOR = color  
SIZE = number if pixels height.  
FACE = used to specify the font names with "," comma seprator.  
EFFECT = EMBOSS | RELIEF | SHADOW ( WebTV).  
ID = value  
LANG = ISO Lang. Abbr.  
POINTSIZE = points.  
SIZE = 1-7, +n, -n.  
STYLE = style information  
TRANSPERANCY = 0-100 (WebTV)  
TITLE = advisory Text

40. **<FRAMESET> ...</FRAMESET>** : This is used to create frames on the HTML document. Basically when we want to display more than one windows on the screen, the "Frames" concept is used. The document that needs to use frames must use the "<FRAMESET>" tag instead of the "<BODY>" tag.

Example:

**<FRAMESET COLS="250,\*" >**

**<FRAME SRC="fileone.html" NAME="Controls">**  
**<FRAME SRC="filetwo.html" NAME="Details">**

### </FRAME>

Syntax:

#### <FRAMESET

BORDER = pixels  
BORDERCOLOR=color  
CLASS=name of class  
COLS = column widths ( in pixels or % )  
FRAMEBORDER = NO | YES | 0 | 1  
FRAMESPACING=pixels  
ID = unique identifier  
LANG = ISO language code  
LANGUAGE = JAVASCRIPT | JSCRIPT | VBSCRIPT | VBS  
ROWS = row heights ( in pixels or % )  
TITLE = advisory text  
Event = Script code

<FRAME> ... </FRAME>

</FRAMESET>

42. **<FRAME>...</FRAME>** : The Frame tag is used to represent individual frames in the document.

Example:

```
<FRAME SRC="fileone.html" NAME="Controls">  
<FRAME SRC="filetwo.html" NAME="Details">
```

Syntax:

#### <FRAMES

BORDERCOLOR = color value  
CLASS=name of class  
DATAFLD = column name  
DATASRC = #ID  
FRAMEBORDER = NO | YES | 0 | 1  
ID = unique identifier  
LANG = ISO language code  
LANGUAGE = JAVASCRIPT | JSCRIPT | VBSCRIPT | VBS  
MARGINHEIGHT = pixels  
MARGINWIDTH = pixels  
NAME = frame name  
NORESIZE  
SCROLLING = AUTO | NO | YES  
SRC = URL of frame contents  
TITLE = advisory text  
event = script code >

</FRAMES>

43. **<NOFRAMES>** : When the browser does not support frames, then the information embedded in the "NOFRAMES" tag is used. This tag MUST be used within the <FRAMESET> tag.

Syntax:

<NOFRAMES

CLASS=name of class(s)  
DIR = LTR | RTL ( Right - to - Left ) / ( Left - to - Right )  
ID = unique ALPHANUMERIC identifier  
LANG = ISO language code  
STYLE = inline style information  
TITLE = advisory text  
event = script code  
NON-Framed content >

</NOFRAMES>

44. **Floating Frames <IFRAME>...</IFRAME>** : In order to have floating frames on the browser, the <IFRAME> syntax is used.

Example:

```
<IFRAME NAME="floatframe1" SRC="fileone.html" WIDTH="350" HEIGHT="200"
ALIGN="LEFT" >
```

There would be a floating frame if the browser supported it.

```
</IFRAME>
```

Syntax :

<IFRAME

ALIGN = LEFT | RIGHT | TOP | ABSMIDDLE\* | ABSBOTTOM\* | TEXTTOP\* | MIDDLE |  
BASELINE\* | BOTTOM ( \*\*\* IE only \*\*\* )  
BORDER = pixels ( \*\*\* IE only \*\*\* )  
BORDERCOLOR = color value ( \*\*\* IE only \*\*\* )  
CLASS=name of class ( \*\*\* IE only \*\*\* )  
DATAFLD = column name ( \*\*\* IE only \*\*\* )  
DATASRC = Data source ( \*\*\* IE only \*\*\* )  
FRAMEBORDER = NO | YES | 0 | 1  
HEIGHT = pixels or %.  
HSPACE = pixels ( \*\*\* IE only \*\*\* )  
ID = unique identifier  
LANG = ISO language code ( \*\*\* IE only \*\*\* )  
LANGUAGE = JAVASCRIPT | JSCRIPT | VBSCRIPT | VBS ( \*\*\* IE only

\*\*\* )

MARGINHEIGHT = pixels  
MARGINWIDTH = pixels  
NAME = frame name  
NORESIZE ( \*\*\* IE only \*\*\* )  
SCROLLING = AUTO | NO | YES  
SRC = URL of frame contents  
STYLE=inlined style information ( \*\*\* IE only \*\*\* )  
TITLE = advisory text  
VSPACE = pixels ( \*\*\* IE only \*\*\* )  
WIDTH=pixels or %.  
event = script code >

</IFRAME>

45. **Cascading Style Sheets**: Style Sheets are used to "SEPERATE THE STRUCTURE OF THE DOCUMENT FROM ITS PRESENTATION". The STYLE must be associated with an HTML ELEMENT.

Example:

```
H1      { font-family: Impact;
          font-size: 28pt }

BODY    { font :10 pt;
          font-family: Serif;
          color: black;
          background: white }
```

General Syntax:

```
element { style specification }
```

There are 3 methods of adding style sheets to HTML documents. They are,

### a.) External Style Sheets :

- \*\* Can set style for many documents with one style sheet.
- ## Requires extra download time for style sheet, which may delay page rendering.
- \*\* The style sheet source is just a simple text file with ".css" extension.

Example:

```
<HTML>
<HEAD>
<TITLE> Style Sheet Linking Example </TITLE>
  <LINK REL = "STYLESHEET" HREF="corpstyle.css" MEDIA="screen,print" TYPE =
"text/css" >
</HEAD>
... content affected by style sheet
</HTML>
```

### b.) Document wide Style sheets:

- \*\* Can control style for document in one place.
- \*\* No additional download time for style information
- ## Need to reapply style information for other documents

Example:

```
<HTML>
<HEAD>
  <TITLE> Document-wide Style Sheets </TITLE>
  <STYLE TYPE = "text/css" MEDIA="PRINT">
  <!--

BODY      {      background: white;
              margin-left: 1in;
              margin-right: 1.5in;      }
```

```
H1          {      font-size: 24pt;
                font-family: sans-serif;
                color: red;
                text-align: center      }

P           {      font-size: 12pt;
                font-family: Serif;
                color: black;
                text-indent: 0.5in     }

-->
</STYLE>
</HEAD>
... content affected by style sheet
</HTML>
```

Another method is to use the @import keyword to import it into the document. The idea here is similar to linking or "inline macro expansion".

```
<!DOCTYPE HTML PUBLIC "-//w3c//DTD HTML 4.0 Final //EN">
<HTML>
<HEAD>
  <TITLE> Imported Style Sheets </TITLE>
  <STYLE TYPE = "text/css" MEDIA="PRINT">
  <!--
    @import "@import http://www.ibm.com/styles/corpstyle.css";
    @import "localmachine_docstyle.css";

H1          {      font-size: 14pt;
                font-family: Times Roman;
                color: red;
                text-align: center      }

P           {      font-size: 12pt;
                font-family: Serif;
                color: black;
                text-indent: 0.5in     }
/* LOCAL RULES OVERRIDE IMPORT */

-->
</STYLE>
</HEAD>
... content affected by style sheet...
</HTML>
```

### c.) Inline Style:

- \*\* Can control style to a single character instance.
- \*\* Overrides any external or document styles.
- ## Need to reapply style information throughout the document and outside documents.
- ## Bound too close to HTML ELEMENTS-difficult to update.

Example:

```
<H1 STYLE="font-size: 48pt; font-family: Arial; color: green" CSS1 Test </H1>
```

\*\* Using "CLASSES" & "IDS" also it is possible to provide inline style reference.

\*\* Using the CLASS attribute also, it is possible to provide style sheet reference.

\*\* Using multiple style sheets imports, the reference with the keyword "STYLE" takes precedence over other styles.

46. Form Basics: Typically a form is made up of controls or fields, as well as the markup necessary to structure the form and control its presentation. Hidden form controls are also possible.

Example:

```
<!DOCTYPE HTML PUBLIC "-//w3c//DTD HTML 3.2 Final//En">
<HTML>
  <HEAD>
    <TITLE> Form Stub < TITLE>
  </HEAD>
  <BODY>
    <FORM ACTION="/cgi-bin/post-query" METHOD="POST">
      Form field controls and standard HTML markup.
    </FORM>
  </BODY>
</HTML>
```

Syntax:

```
<FORM
  ACCEPT-CHARSET = list of supported characters sets
  ACTION = URL
  CLASS = class names
  DIR = LTR | RTL
  ENCTYPE=MIME Type
  ID = unique identifier
  LANG = ISO language code          ( *** IE only *** )
  LANGUAGE = JAVASCRIPT | JSCRIPT | VBSCRIPT | VBS          ( *** IE only
*** )
  METHOD = GET | POST
  NAME = ID
  STYLE= style information
  TARGET = frame name
  TITLE = advisory text
  Eventhandler = "script" >
```

... Form control and HTML markup

```
</FORM>
```

**Text Controls:** Text entry type control is used as an input box to enter text.

Example1:

```
<INPUT TYPE="TEXT" NAME="CUSTOMERNAME">
```

Example2:

```
<!DOCTYPE HTML PUBLIC "THIS IS AN EXAMPLE">
<HTML>
<HEAD>
<TITLE> THIS IS A HELLO WORLD HTML !!!
</TITLE>
</HEAD>

<BODY>
<FORM>

    <INPUT TYPE="password" NAME="custname" SIZE="10" MAXLENGTH="20" >
    </INPUT>

</FORM>
</BODY>

</HTML>
```

Syntax: Commonly used Attributes,

TYPE = TEXT | PASSWORD  
NAME = name of the control  
SIZE = size of font to use.  
MAXLENGTH = max. length of characters.  
VALUE = The text you want displayed in the control.

**Multi-Text Controls:** Text entry type control is used as for multi-line text input.

Example:

```
<!DOCTYPE HTML PUBLIC "THIS IS AN EXAMPLE">
<HTML>
<HEAD>
<TITLE> THIS IS A HELLO WORLD HTML !!!
</TITLE>
</HEAD>

<BODY>
<FORM>
<HR>
    <TEXTAREA ROWS="5" COLS="80" NAME="COMMENTS">
        Please enter your comments here.
    </TEXTAREA>
</FORM>
</BODY>

</HTML>
```

Syntax: Commonly used Attributes,

ROWS = no. of rows.  
COLS = no. of columns.  
NAME = name of the control  
Pull-Down Menus: These are similar to combo boxes.

Example:

```
<SELECT NAME ="CarType">
    <OPTION>Toyota
    <OPTION>Honda
```

```
<OPTION>Nissan
<OPTION>Mitsubishi
<OPTION SELECTED>Mazda
</SELECT>
```

Syntax: Commonly used Attributes,

NAME = name of the control

<OPTION> = used to indicate the item in the combo box.

<OPTION SELECTED> = Used to indicate the default item to be selected

<OPTION VALUE="TOYO">Toyota = This is used when the corresponding code type for the description selected.

### SCROLLING LISTS or LISTBOX:

MULTIPLE ROWS= "n" = This is used when you want a SCROLL LIST or LISTBOX instead of a combo box.

Ex.

```
<SELECT NAME ="CarType" MULTIPLE SIZE="3">
  <OPTION>Toyota
  <OPTION>Honda
  <OPTION>Nissan
  <OPTION>Mitsubishi
  <OPTION SELECTED>Mazda
</SELECT>
```

Check Boxes: These are used to represent options to the user.

Example:

```
<INPUT TYPE="CHECKBOX" NAME="CHEESE" VALUE="CHEESE" CHECKED>
<INPUT TYPE="CHECKBOX" NAME="HAM">
```

Syntax: Commonly used Attributes,

NAME = name of the control

CHECKED = whether the control must be checked by default.

VALUE= the value to be sent to server on select.

Radio Buttons: These are used to represent mutually exclusive options to the user.

Example:

COLOR:

Red <INPUT TYPE="RADIO" NAME="CHEESE" VALUE="RED" >

Green <INPUT TYPE="RADIO" NAME="CHEESE" VALUE="GREEN" CHECKED>

Blue <INPUT TYPE="RADIO" NAME="CHEESE" VALUE="BLUE" >

Syntax: Commonly used Attributes,

NAME = name of the control, [to have a GROUP of radio buttons, use the same NAME for all]

CHECKED = whether the control must be checked by default.

VALUE= the value to be sent to server on select.

Reset & Submit Buttons:

The <INPUT> element has 2 values namely "RESET" & "SUBMIT" for the TYPE attribute. The RESET button option is used to reset the controls to their default values, or clear the changes on the controls.

The SUBMIT creates a button that triggers the browser to send the contents of the form to the address specified in the ACTION attribute of the <FORM> tag.

Example:

```
<!DOCTYPE HTML PUBLIC "THIS IS AN EXAMPLE">
<HTML>
<HEAD>
<TITLE> HTML FORM !!!
</TITLE>
</HEAD>

<BODY>
<FORM ACTION="mailto:webmaster@ibm.com" METHOD="POST"
ENCTYPE="text/plain">

<INPUT TYPE="password" NAME="custname" SIZE="10" MAXLENGTH="20" >
<HR>
<TEXTAREA ROWS="5" COLS="80" NAME="COMMENTS">
Please enter your comments here.
</TEXTAREA>
<HR>

    <SELECT NAME ="CarType">
        <OPTION>Toyota
        <OPTION>Honda
        <OPTION>Nissan
        <OPTION>Mitsubishi
        <OPTION SELECTED>Mazda
    </SELECT>
<HR>
    <SELECT NAME ="CarType" MULTIPLE SIZE="3">
        <OPTION>Toyota
        <OPTION>Honda
        <OPTION>Nissan
        <OPTION>Mitsubishi
        <OPTION SELECTED>Mazda
    </SELECT>
<HR>
<INPUT TYPE="CHECKBOX" NAME="CHEESE" VALUE="CHEESE" CHECKED>
<INPUT TYPE="CHECKBOX" NAME="HAM">
<INPUT TYPE="RADIO" NAME="CHEESE"CHECKED>
<INPUT TYPE="RADIO" NAME="HAM">
<HR>
COLOR :
<BR>
Red <INPUT TYPE="RADIO" NAME="CHEESE" VALUE="RED" >
Green <INPUT TYPE="RADIO" NAME="CHEESE" VALUE="GREEN" CHECKED>
Blue <INPUT TYPE="RADIO" NAME="CHEESE" VALUE="BLUE" >
<HR>

<INPUT TYPE="SUBMIT" VALUE="Submit Form" NAME="SubmitButton">
```

```
<INPUT TYPE="RESET" VALUE="Reset Form" NAME="ResetButton">
```

```
</FORM>
```

```
</BODY>
```

```
</HTML>
```

Syntax: Commonly used Attributes,

VALUE = The caption to be displayed on the BUTTON.

NAME = name of the control.

Note: When multiple SUBMIT buttons are used, say with "Insert", "Update" and "Delete" functionality, then the value of the button is sent to the address specified by the <FORM>'s "ACTION" attribute.

Example:

```
<INPUT TYPE="SUBMIT" VALUE="INSERT" NAME="Insert">
<INPUT TYPE="SUBMIT" VALUE="UPDATE" NAME="Update">
<INPUT TYPE="SUBMIT" VALUE="DELETE" NAME="Delete">
<INPUT TYPE="RESET" VALUE="Reset Form" NAME="ResetButton">
```

When the form is submitted, the value of the button is sent to the form handling program , which will decide what to do with the submitted data based upon its contents.

**Hidden Controls:** Usage of hidden controls is to preserve the state. For example when the user enters his USERID and PASSWORD on the first form, then on he is shown say, "Order Entry" form. But whenever data for subsequent transactions is sent, the hidden controls namely, USERID & PASSWORD are used to hold the user information to be sent to the server there on.

Example:

```
<INPUT TYPE ="hidden" NAME ="SubmittingForm" VALUE="Form1">
```

Syntax: Commonly used Attributes,

TYPE = "HIDDEN"

VALUE = The value to be sent.

NAME = name of the control.

Image Form Control/INPUT: This used in conjunction with a IMAGE source, and the X,Y coordinates are sent to the server when clicked on this kind of input control.

Example: <INPUT TYPE="IMAGE" SRC="ibmlogo.gif" NAME="SALES" ALT="Sales Product">

All the attributes that apply for "IMAGE" tag applies to this input control.

File Form Control/INPUT: This is used when a FILE dialog is to be used for setting some data and transmitting that file, which could be an image or a text file etc.

Example:

```
<INPUT TYPE="FILE" NAME="FileName">
```

This will create a text box, and a "BROWSE..." button next to it. It is possible to select a particular file of choice and sent it.

Generalized Buttons: It is also possible to create "GENERALIZED" buttons, and later on tie some "script" code to execute when action is performed on the button.

```
<INPUT TYPE="BUTTON" VALUE="Press Me!" NAME="ABUTTON">
```