

UNIT-I

Introduction to HTML:-

HTML means Hypertext Markup Language. In 1960 Ted Nelson introduced Hypertext. HTML is a scripting language which is used to create web pages. If you are thinking of creating your own web pages, you need to know at least basic HTML. These HTML documents are plain text files, user can create these documents using text editor like Notepad or Edit.

HTML is a hypertext Language because it supports font styled text, pictures, graphics and animations and also it provides hyper links that used to browse the Internet easily. Text becomes hypertext with the addition of links that connects other hypertext documents. Hypertext is a text augmented with links-pointers to other pieces of text, possible else where in the same document (internal linking) or in another document (external linking).

Rules to write HTML Code:-

- ❖ Every HTML document begins with start tag is <HTML> terminates with an ending tag is </HTML>
- ❖ HTML documents should be saved with the extension .html or .htm.
- ❖ A tag is made up of left operator(<), a right operator(>) and a tag name between these two operators.
- ❖ If you forget to mention the right operator(>) or if you give any space between left operator and tag name browser will not consider it as tag.
- ❖ At the same time if browser not understands the tag name it just ignores it, browser won't generate any errors.
- ❖ HTML language is not case sensitive, hence user can write the code in either upper case or lower case. No difference between <HTML> and <html>

Syntax of a tag:

<Tagname [parameters=value]>

Ex: HR is a tag name that displays a horizontal ruler line.

<HR> -----(No parameters, no value)

<HR ALIGN=CENTER>----- (Tag with parameter and value for the parameter)

<HR WIDTH="30%" SIZE=100 ALIGN=RIGHT>----- (Tag with more parameters with their values)

Different types of Tags:

1. Singleton tags
2. Paired tags

Singleton tag does not require an ending tag. (Ex: <HR>

Paired tag required an ending tag, which is similar to opening tag except backslash before the tag name (Ex: <HTML> is opening tag, then ending tag is </HTML>

Comments in HTML:

An HTML comment begins with "<!--" and ends with "<!-->". There should not be a space between angular bracket and exclamation mark.

Creating HTML Page:

The Following steps are needed to create a HTML page

Step 1: Open any text editor like Notepad, Edit, Word etc.

Step 2: Use the file menu to create a new document (File □ New) and type the following code

```
<HTML>
<HEAD>
<TITLE>Example1 </TITLE>
<BODY>
```

Hello III IT ,this is your first web page.- Raju

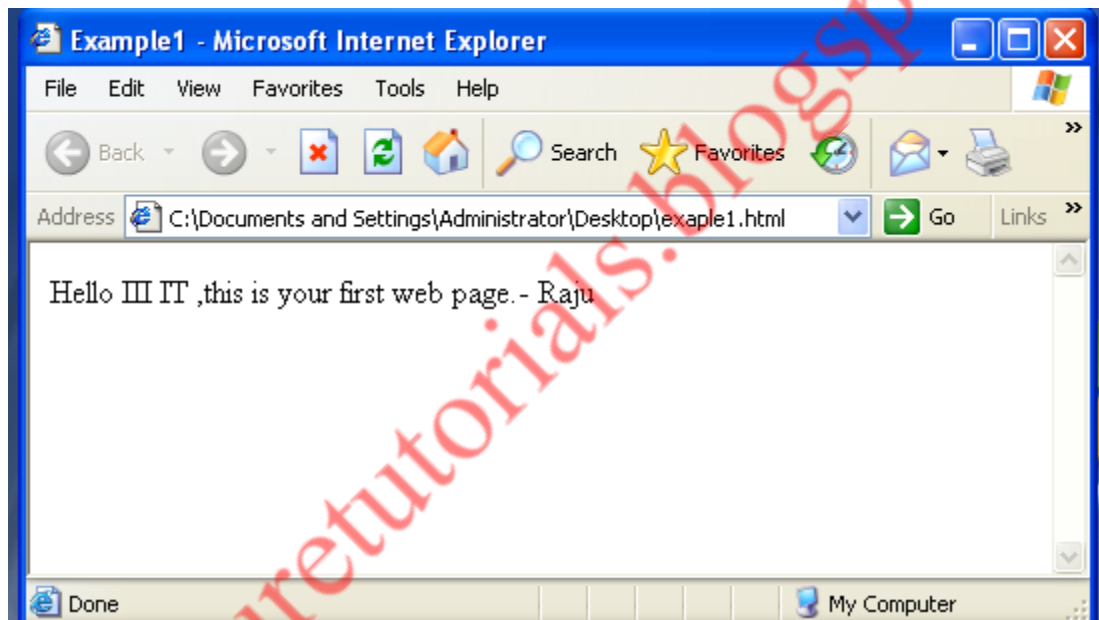
```
</BODY>
</HTML>
```

Step 3: Go to the file menu and choose saveas option (File->saveas) and give the name of the file as “example1.html” under root directory(C:)(or any valid path)

Step 4: After saving, an internet explorer icon will be displayed as shown below



Step 5: Double click to execute it. The output displayed following



Basic HTML tags

1. Body tag:-

Body tag contain some attributes such as bgcolor, background etc. bgcolor is used for background color, which takes background color name or hexadecimal number and #FFFFFF and background attribute will take the path of the image which you can place as the background image in the browser.

```
<body bgcolor="#F2F3F4" background= "c:\amer\imag1.gif">
```

2. Paragraph tag:-

Most text is part of a paragraph of information. Each paragraph is aligned to the left, right or center of the page by using an attribute called as align.

```
<p align="left" | "right" | "center">
```

3. Heading tag:-

HTML is having six levels of heading that are commonly used. The largest heading tag is

<h1>. The different levels of heading tag besides <h1> are <h2>, <h3>, <h4>, <h5> and <h6>. These heading tags also contain attribute called as align.

<h1 align="left" | "right" | "center"><h2>

4. hr tag:-

This tag places a horizontal line across the system. These lines are used to break the page. This tag also contains attribute i.e., width which draws the horizontal line with the screen size of the browser. This tag does not require an end tag.

<hr width="50%">.

5. base font:-

This specify format for the basic text but not the headings.

<basefont size="10">

6. font tag:-

This sets font size, color and relative values for a particular text.

7. bold tag:-

This tag is used for implement bold effect on the text

8. Italic tag:-

This implements italic effects on the text.

<i>... ..</i>

9. strong tag:-

This tag is used to always emphasized the text

.....

10. tt tag:-

This tag is used to give typewriting effect on the text

<tt> </tt>

11. sub and sup tag:-

These tags are used for subscript and superscript effects on the text.

_{.....}

^{.....}

12. Break tag:-

This tag is used to the break the line and start from the next line.

13. & < > ":-

These are character escape sequence which are required if you want to display characters that HTML uses as control sequences.

Example: < can be represented as <.

14. Anchor tag:-

This tag is used to link two HTML pages, this is represented by <a>

 some text

href is an attribute which is used for giving the path of a file which you want to link.

Example 1: HTML code to implement common tags.

mypage.html

```
<html>
<head> <!-- This page implements common html tags -->
<title> My Home page </title>
</head>
<body >
<h1 align="center"> ST ANN"S COLLEGE OF ENGINEERING & TECHNOLOGY</h1>
<h2 align="center"> Chirala</h2>
<basefont size=4>
<p> This college runs under the <tt>management</tt> of <font size=5> <b><i>&quot;
SACET&quot; &amp </i></b></font><br>
  it is affiliated to <strong> JNTUK</strong>
<hr size=5 width=80%>
<h3> <u>&lt; Some common tags &gt</u> </h3><br>
</body>
</html>
```

Text Styles or Cosmetic tags:- HTML provides a numerous range of tags for formatting the text. If you want to format the text with different styles, just you include these tags one by one before text.

.....	Bold Text
<U>.....</U>	Underline Text
<I>.....</I>	Displays as Italics
.....	For Emphasis (New Standard for Italics)
.....	Strong or Bold text (New Standard for Bold)
<S>.....</S> or	Strikes the text
<SAMP>.....</SAMP>	Code sample text
<VAR>.....</VAR>	Small fonts, fixed width
<ADDRESS>.....</ADDRESS>	Like address model (Looks like italics)
<PRE>.....</PRE>	Considers spaces, new lines etc. As it is prints the information

Scrolling Text Tag:-

<marquee> </marquee> Displays scrolling text in a marquee style.

Marquee tag attributes:-

- align: sets the alignment of the text relative to the marquee. Set to top(default), middle or bottom.

- b) behaviour: Sets how the text in the marquee should move, It can be scroll(default), slide(text enters from one side and stops at the other end), or alternative(text seems to bounce from one side to other)
- c) bgcolor: sets the background color for the marquee box
- d) direction: sets the direction of the text for scrolling. It can be left(default), right, down or up.

Example:-

```
<marquee align="middle" behavior="slide" bgcolor="red" direction="down">Raju</marquee>
```

Blinking text Tag:-

`<blink>.....</blank>` displays enclosed text as blinking on and off approximately once a second.

Linking in HTML:-

Text becomes hypertext with the addition of links which connect separate locations with in a collection of hypertext documents.



Hyperlinks can be applied for either text or images. Links may connect several web pages of a web site. Links can connect web pages on the same or different servers. Navigation between pages became easier because of links. Information in the same page also connected through links(internal links).

```
<A> ..... </A>
```

Anchor tag is used for creating links. Minimum it requires a parameter i.e., HREF, which indicates the destination document. Other parameters Name and target can be useful for identification for anchor tag and the location of a frame where target page is to be displayed respectively. Name and target tag are optional.

Syntax: ``

```
.....
..... id=Text that displays as link
</A>
```

HREF Parameter:-

If HREF is included, the text between the opening and closing anchor element that between `<A>` and `` becomes hyper text. If users clicks on this text, they are moved to specified document.

Ex:-

```
<A HREF=www.vaneshdoc.com>Publishers</A>
```

Result displayed as Publishers

When the user click on this text, Publishers web site is displayed on the browser.

Example :

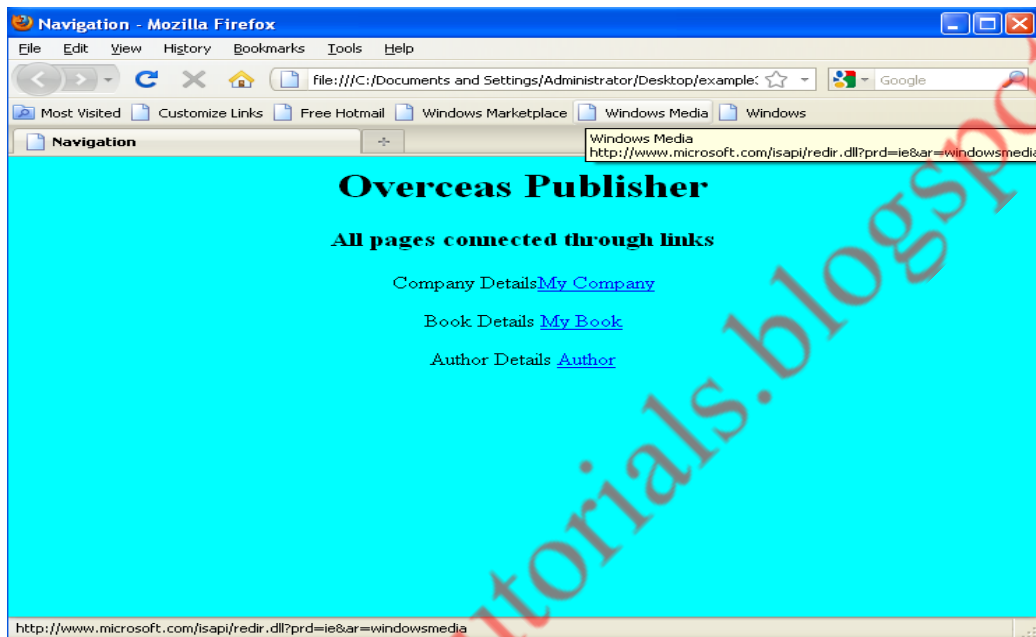
Create a HTML web page that connect web pages created through hyperlinks.

```
<html>
```

```

<head>
<title> Navigation </title>
</head>
<body bgcolor=cyan>
<h1 align=center>Overceas Publisher </h1>
<h3 align=center>All pages connected through links</h3>
<center>
<p>Company Details<A HREF="1.html">My Company</A>
<p>Book Details <A HREF="2.html">My Book</A>
<p> Author Details <A HREF="3.html"> Author</A>
</center>
</body>
</html>

```



Color and Image:

Color can be used for background, elements and links. To change the color of links or of the page background hexadecimal values are placed in the <body> tag.

```

<body bgcolor = "#nnnnnn" text = "#nnnnnn" link= "#nnnnnn" vlink= "#nnnnnn" alink
= "#nnnnnn">

```

The vlink attribute sets the color of links visited recently, alink the color of a currently active link. The six figure hexadecimal values must be enclosed in double quotes and preceded by a hash(#).

Images are one of the aspect of web pages. Loading of images is a slow process, and if too many images are used, then download time becomes intolerable. Browsers display a limited range of image types.

```

<body background = "URL">

```

This tag will set a background image present in the URL.

Another tag that displays the image in the web page, which appears in the body of the text rather than on the whole page is given below

```



```

Example 4: HTML code that implements color and image

```

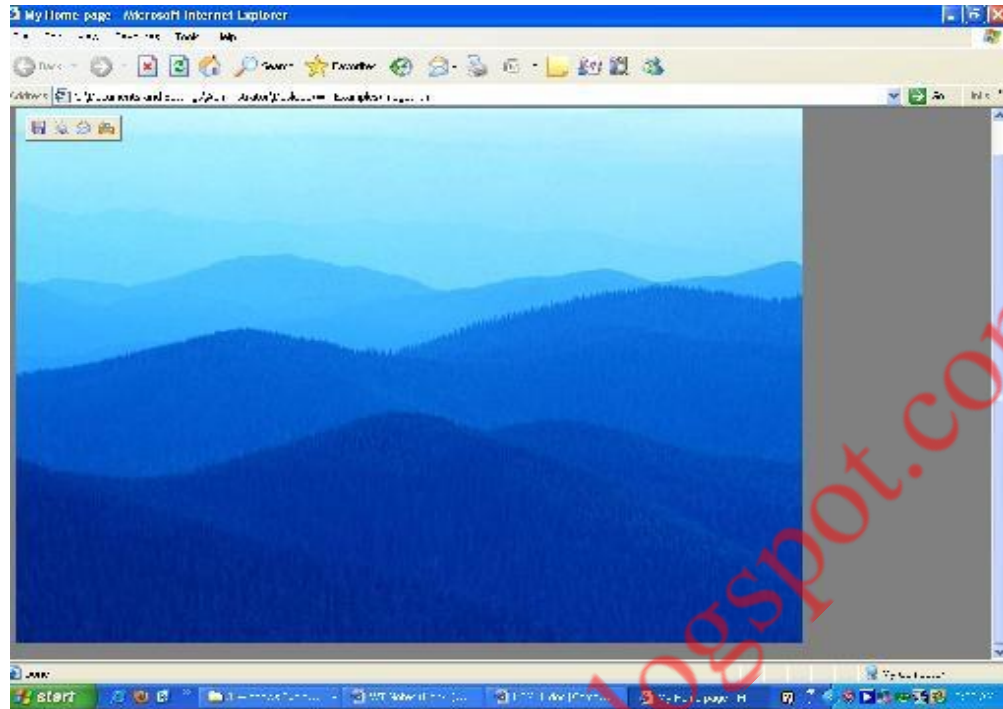
<html>
<head> <!-- This page implements color and image -->
<title> My Home page </title>
</head>
<body bgcolor="gray" text="magenta" vlink="yellow" alink="brown">
<img src= " C:\Documents and Settings\All Users\Documents\My

```


Pictures\Sample Pictures\Blue hills.jpg">

</body>

</html>



Example 5: HTML code that implements background image

<html>

<head> <!-- This page implements background image -->

<title> My Home page </title>

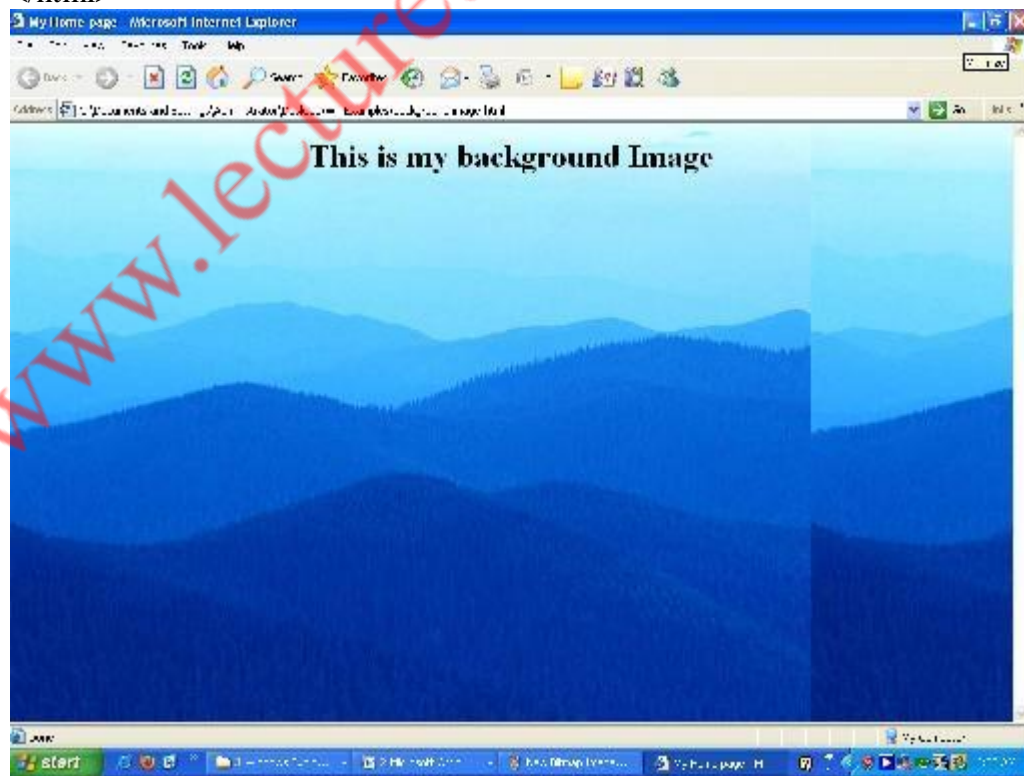
</head>

<body background="C:\Documents and Settings\All Users\Documents\My Pictures\Sample Pictures\Blue hills.jpg">

<h1 align="center"> This is my background Image</h1>

</body>

</html>



Lists:

One of the most effective ways of structuring a web site is to use lists. Lists provides straight forward index in the web site. HTML provides three types of list i.e., bulleted list, numbered list and a definition list. Lists can be easily embedded easily in another list to provide a complex but readable structures. The different tags used in lists are explained below.

Unordered Lists:-

Unordered lists are also called unnumbered .lists. The Unordered list elements are used to represent a list of items, which are typically separated by white space and/or marked by bullets. Using tag does creation of unordered lists in HTML. Which is paired tag, so it requires ending tag that is . The list of items are included in between The TYPE attribute can also be added to the tag that indicates the displayed bullet along with list of item is square, disc or circle. By default it is disc.

Syntax:- <UL [TYPE={square/disc/circle}]>
 item name1
 item name2

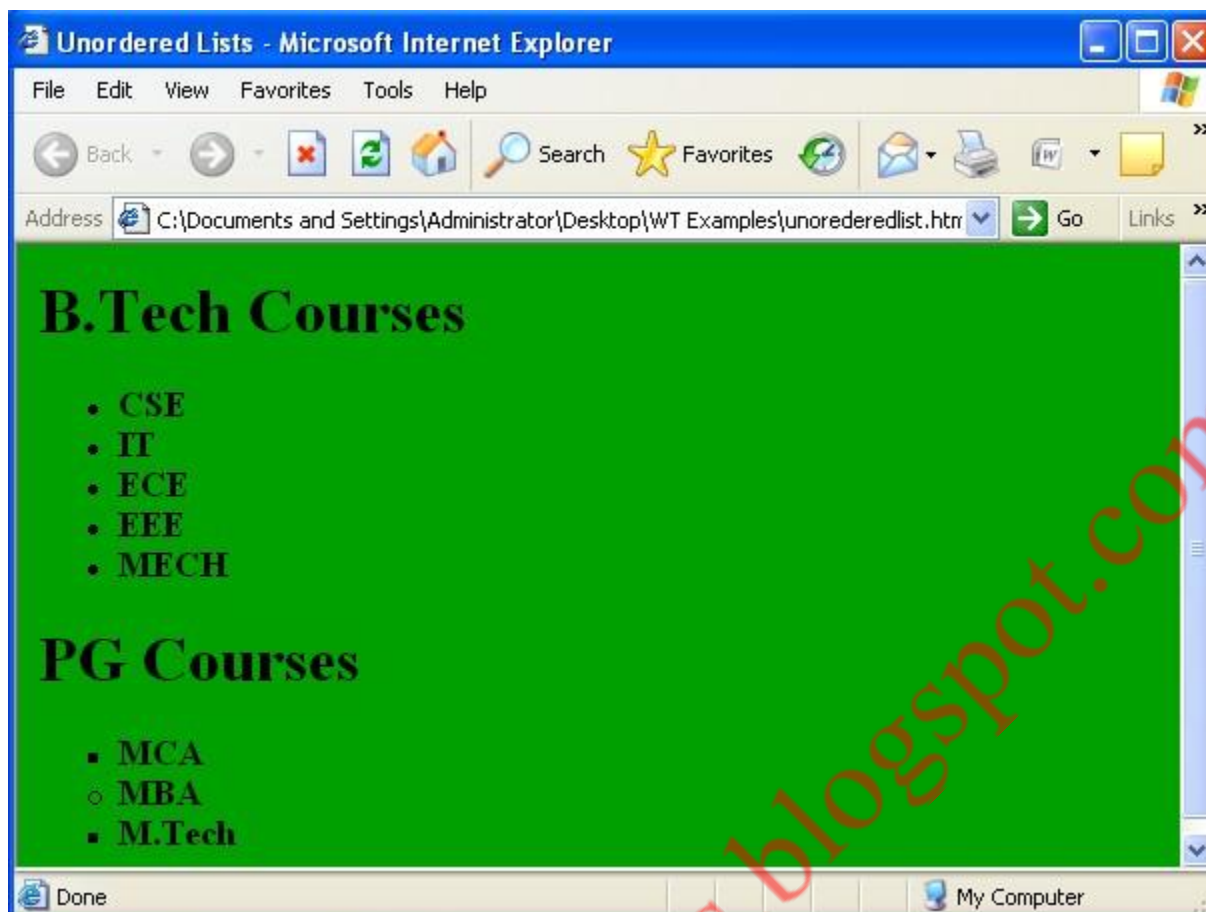
 item namen

Example:

Write a HTML program for displaying names of B.Tech Courses with default bullets and names of PG Courses with square bullets.

```
<html>
  <head>
    <title>Unordered Lists</title>
  </head>
  <body bgcolor="tan">

    <h1>B.Tech Courses
    <h3>
  <ul>
    <li>CSE
    <li>IT
    <li>ECE
    <li>EEE
    <li>MECH
  </ul>
  </h3>
  <h1>PG Courses
  <h3>
    <ul type="square">
      <li>MCA
      <li type="circle">MBA
      <li>M.Tech
    </ul>
  </h3>
</body>
</html>
```

Ordered Lists:-

Ordered lists are also called sequenced or **numbered lists**. In the ordered list the list of item have an order that is signified by numbers, hence it some times called as number lists. Elements used to present a list of items, which are typically separated by white space and/or marked by numbers or alphabets. An **orders** list should start with the element, which is immediately followed by a element which is same as in unordered list. End of ordered lists is specified with ending tag .

Different Ordered list types like roman numeral list, alphabet list etc. can be specified with TYPE tag. Another optional parameter with tag is START attribute, which indicates the starting number or alphabet of the ordered list. For example TYPE="A" and START=5 will give list start with letter E. The TYPE attribute used in , changes the list type for particular item. To give more flexibility to list, we can use VALUE parameter with tag that helps us to change the count for the list item and subsequence items.

Syntax:-

```
<OL [type={"1" or "I" or "a" or "A" or "i"}] START=n>
```

```
<LI>item name1
```

```
<LI>item name2
```

```
-----
```

```
-----
```

```
<LI>item namen
```

```
</OL>
```

Different Ordered list types

Type="1" (default) e.g.1,2,3,4.....

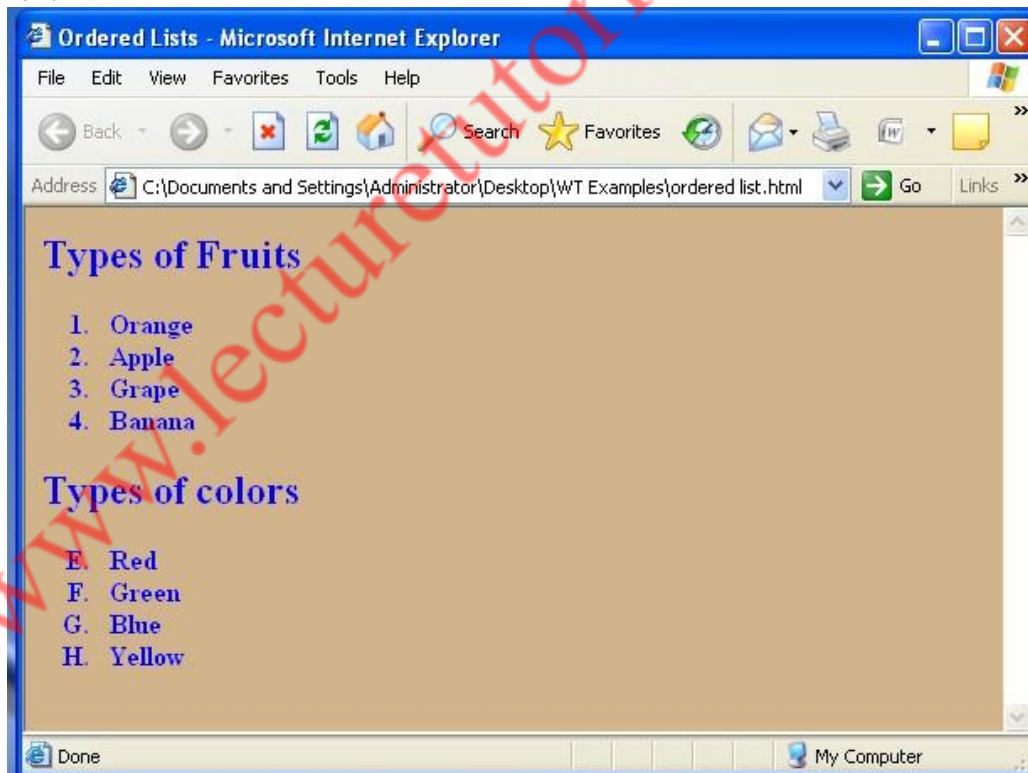
Type="A" Capital letters e.g.A,B,C...

Type="a" Small letters e.g. a,b,c.....

Type="I" Large roman letters e.g. I, II, III,...

Example:-

```
<html>
<head>
<title>Ordered Lists</title>
</head>
<body bgcolor="tan" text="blue">
<h2> Types of Fruits
<h4>
    <OL>
        <LI>Orange
        <LI>Apple
        <LI>Grape
        <LI>Banana
    </OL>
</h4>
<h2>Types of colors
<h4>
    <OL type="A" START=5>
        <LI>Red
        <LI>Green
        <LI>Blue
        <LI>Yellow
    </OL>
</h4>
</body>
</html>
```



Other Lists:-

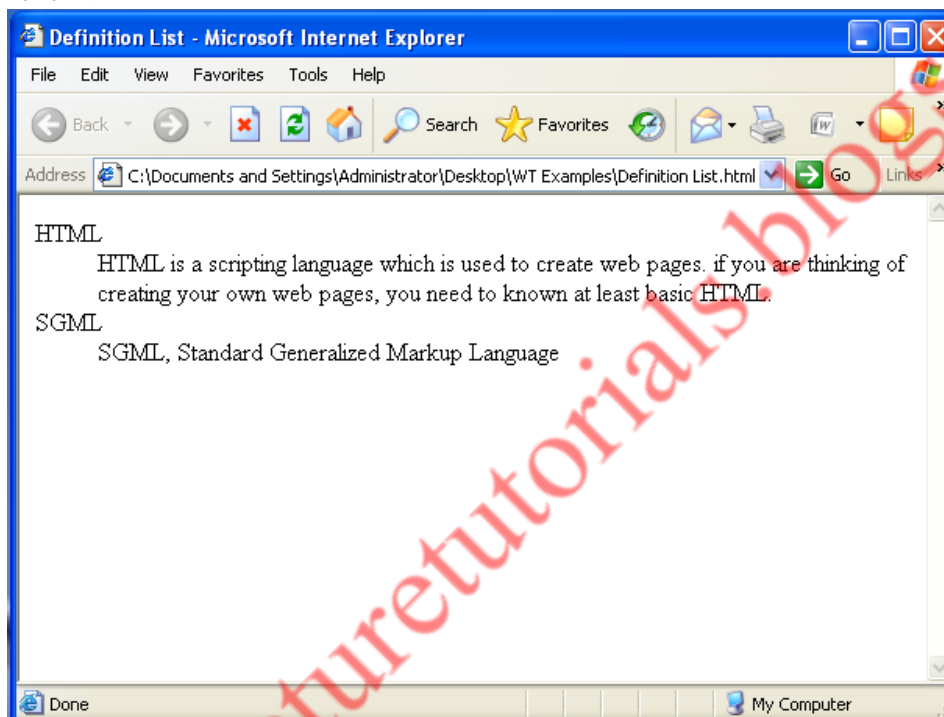
There are several lists in HTML, some of them are definition list and Directory List.

Definition List:- <DL>.....</DL>

A Definition list is a list of definition terms <DT> and corresponding Definition Description<DD> on a new line. To create a definition it must start with <DL> and immediately followed by the first definition term <DT>

Example:-

```
<html>
  <head>
    <title>Definition List</title>
  <body>
    <DL>
      <DT>HTML
      <DD> HTML is a scripting language which is used to create web pages. if you
are thinking of creating your own web pages, you need to know at least basic HTML.
      <DT>SGML
      <DD>SGML, Standard Generalized Markup Language
    </DL>
  </body>
</html>
```



Directory List:-

A Directory list element is used to present a list of items containing up to 20 characters each. Items in a Directory List may be arranged in columns, typically 24 characters wide. A Directory List begins with <DIR> element, which is immediately followed by a element. This tag is a deprecated tag, so it is not preferable to use. Hence, use instead of <DIR>

Other information

```
<DIR>
  <LI>Contacts-2043240
  <LI>Business-4123412
  <LI>Personal-3123122
</DIR>
```

Nested Lists:- Lists can be nested that is Nested Lists is a list within another list.

Tables:

Table is one of the most useful HTML constructs. Tables are found all over the web application. The main use of table is that they are used to structure the pieces of information and to structure the whole web page. Below are some of the tags used in table.

```
<table align="center" | "left" | "right" border="n" width="n%" cellpadding="n"
        cellspacing="n">
.....
</table>
```

Everything that we write between these two tags will be within a table. The attributes of the table will control in formatting of the table. Cell padding determines how much space there is between the contents of a cell and its border, cell spacing sets the amount of white space between cells. Width attribute sets the amount of screen that table will use.

```
<tr> ..... </tr>
```

This is the sub tag of <table> tag, each row of the table has to be delimited by these tags.

```
<th>.....</th>
```

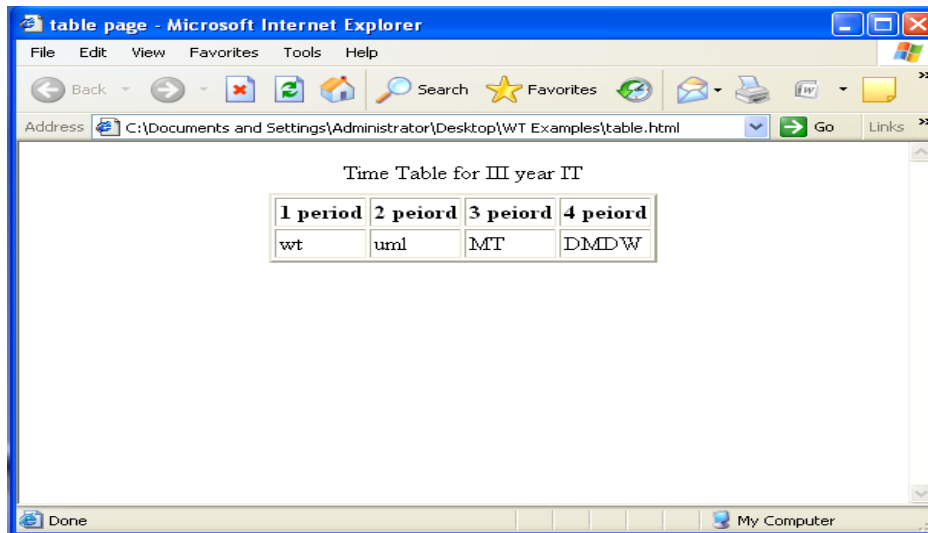
This is again a sub tag of the <tr> tag. This tag is used to show the table heading .

```
<td>.....</td>
```

This tag is used to give the content of the table.

Example 3: HTML code showing the use of table tag

```
<html>
<head>
<title> table page</title>
</head>
<body>
<table align="center" cellpadding="2" cellspacing="2" border="2">
<caption> Time Table for III year IT </caption>
<tr><th> 1 period </th>
<th> 2 period </th>
<th> 3 period </th>
<th> 4 period </th>
</tr>
<tr>
<td> wt </td>
<td> unl</td>
<td> MT</td>
<td> DMDW</td>
</tr>
</table>
</body>
</html>
```

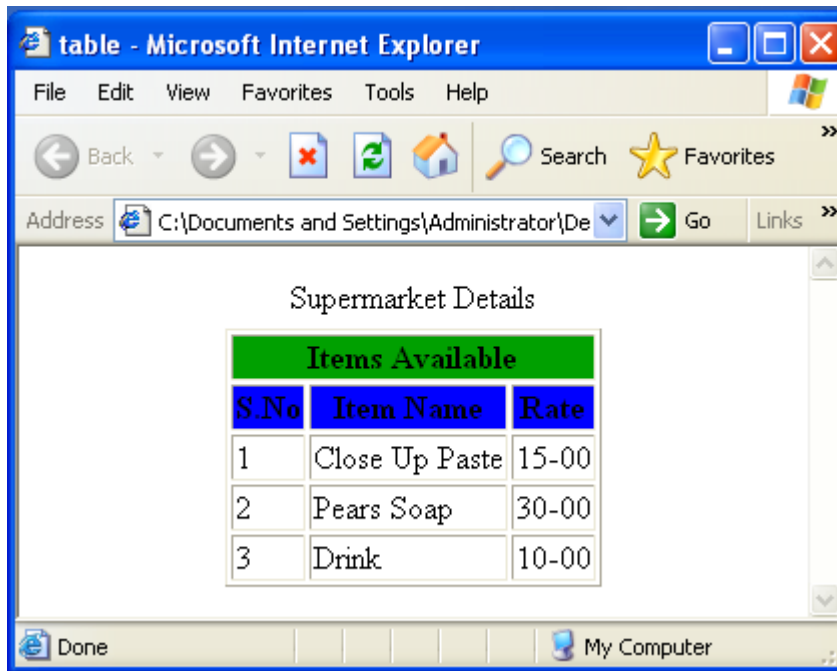


Complex HTML Tables and Formatting:-

You can add background color and background images by using bgcolor and background attributes respectively. Spanning of cells is possible that is you can merge some sequence of rows or columns with the help of ROWSPAN or COLSPAN attributes respectively. For example `<th COLSPAN="2">` widened to span two cells. VALIGN attribute is used for vertical alignment formats and it accepts the values "top", "middle", "bottom" and "baseline".

Example:

```
<html>
  <head>
    <title> table</title>
  </head>
  <body>
    <center>
      <table border="2">
        <caption>Supermarket Details</caption>
        <tr>
          <th colspan=3 bgcolor="tan" align="center">Items
Available</th>
        </tr>
        <tr><th>S.No<th>Item Name<th>Rate</tr>
        <tr><td>1<td>Close Up Paste<td>15-00</tr>
        <tr><td>2<td>Pears Soap<td>30-00</tr>
        <tr><td>3<td>Drink<td>10-00</tr>
      </table>
    </center>
  </body>
</html>
```

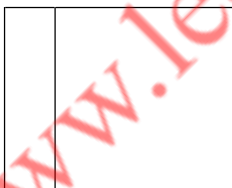



Frames:

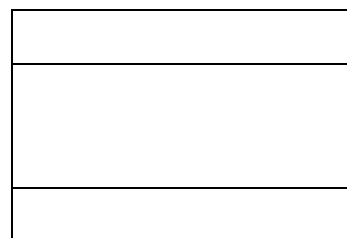
Frames provide a pleasing interface which makes your web site easy to navigate. When we talk about frames actually we are referring to frameset, which is a special type of web page.

Simply frameset is nothing but collection of frames. Web page that contains frame element is called framed page. Framed page begins with `<frameset>` tag and ends with `</frameset>`. Each individual frame is identified through `<frame>` tag. Creation of framed page is very simple. You can nest the framesets. First you decide how you want to divide your webpage and accordingly define frame elements.

Consider the following diagrams, first form divides into two columns and the second form divides into three rows.



Two columns frameset



Three rows frameset

In order to divide into two columns we can use the following syntax

```
<frameset cols="25%,75%">
<frame name="disp" src="1.html">
<frame name="res" src="2.html">
</frameset>
```

In the second diagram we have three rows so by using rows parameter of frameset, we can divide logically the window into three rows.

```
<frameset rows="20%,*,10%">
<frame name="first" src="1.html">
```

```
<frame name="second" src="2.html">
<frame name="third" src="3.html">
</frameset>
```

According to above code ,first row occupies 20% of the window, third row occupies 10% of the window, second row * represents remaining area that is 70% of the window.

Nested Framesets:-

Some times it is required to divide your window into rows and columns, then there is requirement of nested framesets. Frameset with in another frameset is known as nested frameset.

The purpose of NAME parameter in frame tag in the above example is nothing but main importance is if we have some links in left side and you want to display respective pages in the right side frame, then name is essential. Using target parameter of Anchor(A) tag as follows users can specify name of frame.

Example:

First.html

```
<frameset rows="20%,*">
  <frame name="fr1" src="frame1.html">
  <frameset cols="25%,*">
    <frame name="fr2" src="frame2.html">
    <frame name="fr3" src="frame3.html">
  </frameset>
</frameset>
```

Frame1.html

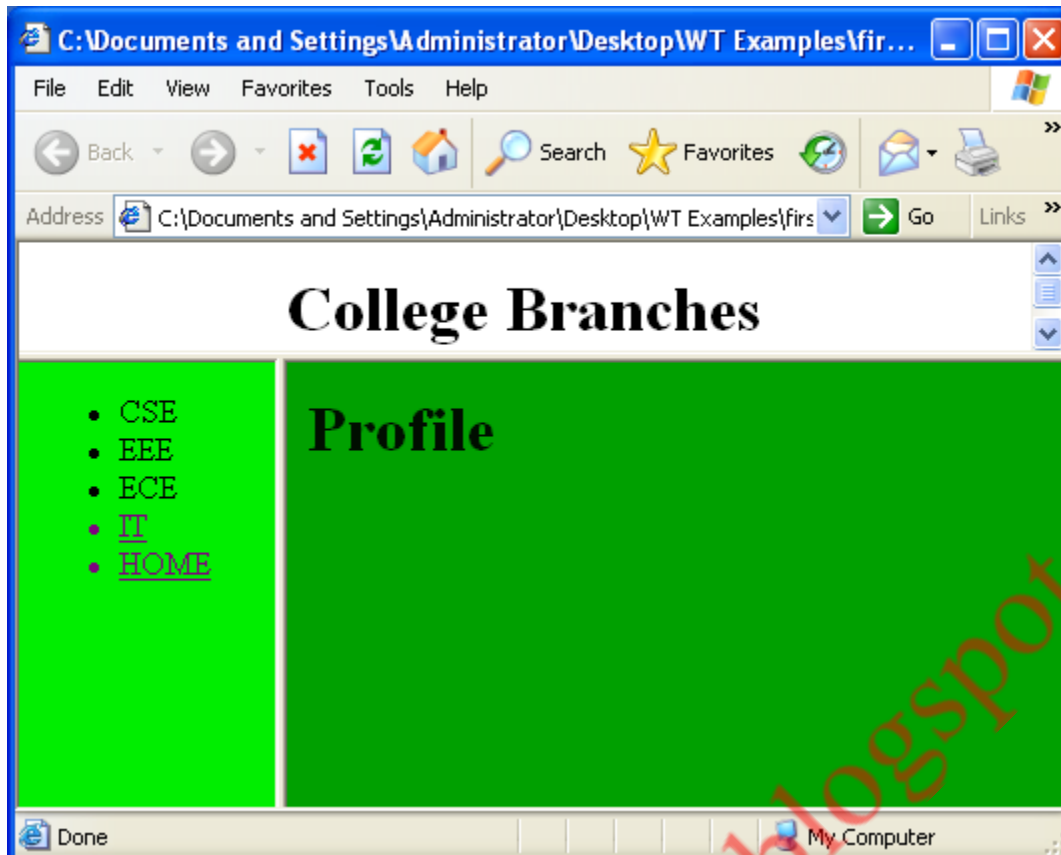
```
<html>
  <body>
    <center><h1> College branches</h1></center>
  </body>
</html>
```

Frame2.html

```
<html>
  <body bgcolor="green">
    <ul>
      <li>CSE
      <li>EEE
      <li>ECE
      <A href="example2.html" target="fr3"><li>IT</A>
    </ul>
  </body>
</html>
```

Frame3.html

```
<html>
  <body text="white" bgcolor="tan">
    <h1>Profile</h1>
  </body>
</html>
```



Forms:

Forms are the best way of adding interactivity of element in a web page. They are usually used to let the user to send information back to the server but can also be used to simplify navigation on complex web sites. The tags that use to implement forms are as follows.

```
<forms action="URL" method = "post" | "get">.....</form>
```

When get is used, the data is included as part of the URL. The post method encodes the data within the body of the message. Post can be used to send large amount of data, and it is more secure than get. The tags used inside the form tag are:

```
<input type = "text" | "password" | "checkbox" | "radio" | "submit" name="string" value="string" size="n">
```

In the above tag, the attribute type is used to implement text, password, checkbox, radio and submit button.

Text: It is used to input the characters of the size n and if the value is given than it is used as a default value. It uses single line of text. Each component can be given a separate name using the name attribute.

Password: It works exactly as text, but the content is not displayed to the screen, instead an * is used.

Radio: This creates a radio button. They are always grouped together with a same name but different values.

Checkbox: It provides a simple checkbox, where all the values can be selected unlike radio button.

Submit: This creates a button which displays the value attribute as its text. It is used to send the data to the server.

```
<select name="string">.....</select>
```

This tag helps to have a list of item from which a user can choose. The name of the particular select tag and the name of the chosen option are returned.

```
<option value="string" selected>.....</option>
```

The select statement will have several options from which the user can choose. The values will be displayed as the user moves through the list and the chosen one returned to the server.

```
<textarea name="string" rows="n" cols="n">.....</textarea>
```

This creates a free format of plain text into which the user can enter anything they like. The area will be sized at rows by cols but supports automatic scrolling.

Example 6: HTML code that implements forms

```
<html>
<head>
<title>form</title>
</head>
<body>
<p align="left">Name:<input type="text" maxlength=30 size=15>
<p align="left">Password:<input type="password" maxlength=10 size=15>
<p align="left">Qualification: <br><input type="checkbox" name="q"
value="be">B.E
<input type="checkbox" name="q" value="me">M.E
<p align="left">Gender:<br> <input type="radio" name="g" value="m">Male
<input type="radio" name="g" value="f">Female
<p align="left">course:<select name="course" size=1>
<option value=cse selected>CSE
<option value=it>CSIT
</select>
<p align="left">Address:<br><textarea name="addr" rows=4 cols=5
scrolling=yes></textarea>
<p align="center"><input type="submit" name="s" value="Accept">
<p align="center"><input type="reset" name="c" value="Ignore">
</body>
</html>
```

Cascading Style Sheet:-

CSS stands for Cascading Style Sheets. It not only extends its features in controlling colors and sizes of fonts, but also controls spaces between various elements, the color and width of a given line etc.

Syntax: p {font-size:30pt;}
 p =selection
 Font-size =property
 30pt =value

Types of Style Sheets:-

1. Inline Style sheet
2. Embedded Style Sheet
3. External Style Sheet

An inline sheet applies style to a particular element in a webpage

Embedding a style sheet is for defining styles to a single webpage. Linking and importing are two ways to attach external style sheet to html documents.

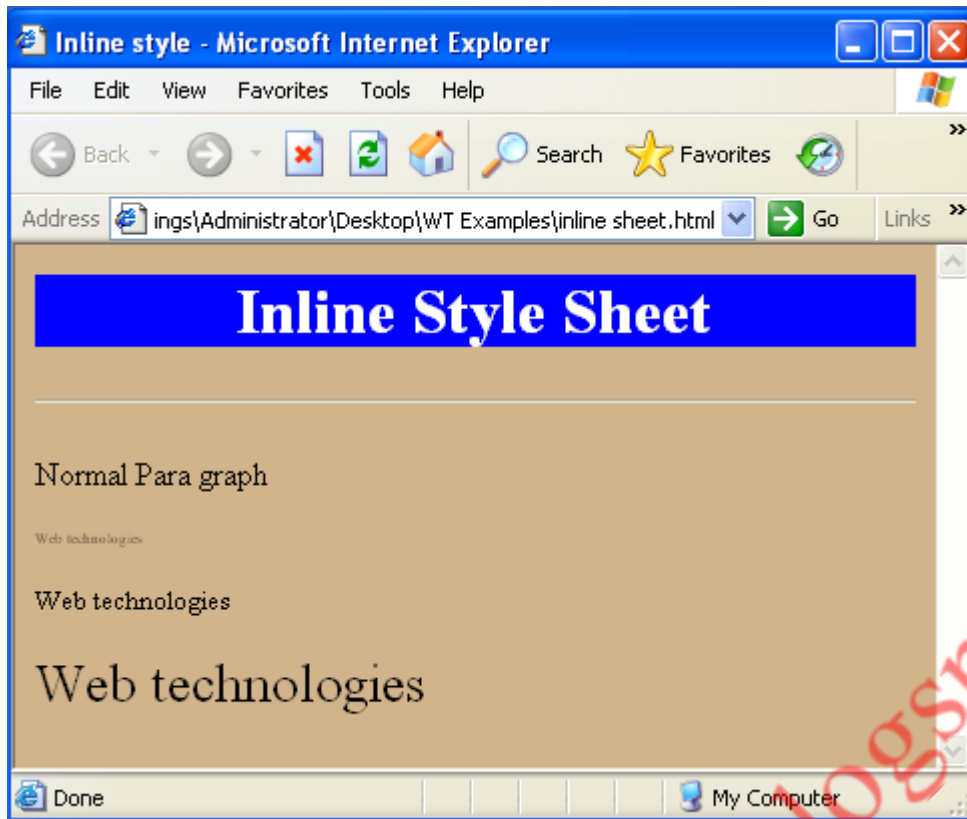
An external style sheet is a file created separately, saved with the .css extension and attached to an HTML document by means of the link element. Linking an external style sheet can apply styles to multiple web pages.

Inline Style Sheet:-

Inclusion of style in a tag is called inline styles. Operator colon(:) Is followed by style property. To separate multiple properties we have to use operator semicolon(;

Example:-

```
<html>
  <head>
    <title>Inline style</title>
  </head>
  <body bgcolor="tan">
    <h1 align="center" style="color:white;background-color:blue">
      Inline Style Sheet
    </h1>
    <hr>
    <p>Normal Para graph</p>
    <p style="font-size:5pt">Web technologies</p>
    <p style="font-size:10pt">Web technologies</p>
    <p style="font-size:20pt">Web technologies</p>
  </body>
</html>
```

Embedded Style Sheet:-

If Style is used as tag, in Header Section then that style sheet is known as internal style sheet. If you include all the formatting parameters in between `<style>` and `</style>`, then this is called as internal style sheets or embedded style sheets. Advantage of Internal Style Sheet comparing with inline styles, at a time several tags can be formatted with internal style sheets, where as in inline styles only one tag at a time formatted.

`<style>`

```
P      { color:red;font-family:arial}
.s5    { font-size:5;}
.s10   { font-size:10;}
.s15   { font-size:20;}
H1     { color:white;background-color:blue }
```

`</style>`

Example:-

`<html>`

`<head>`

`<title>Embedded Style sheet</title>`

`<style>`

```
P      { color:red;font-family:arial}
.s5     { font-size:5;}
.s10    { font-size:10;}
```

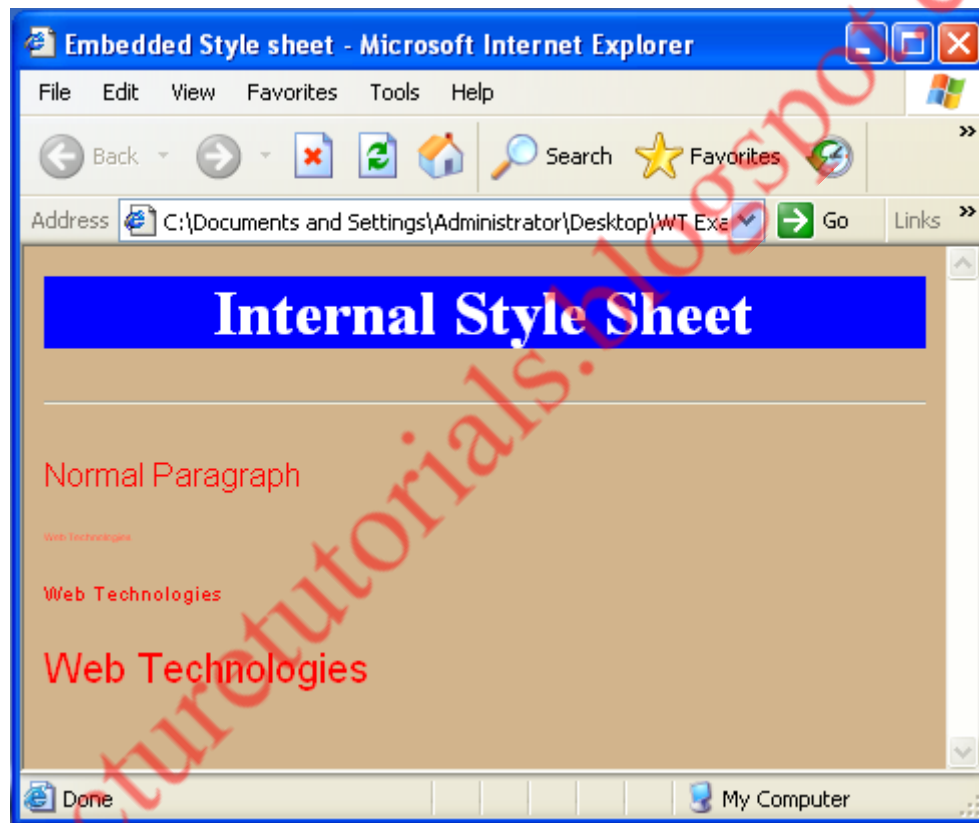
```

        .s15    { font-size:20;}
        H1      { color:white;background-color:blue}
    </style>

</head>
<body bgcolor=tan>
    <h1 align="center"> Internal Style Sheet</h1>
    <hr>
    <p>Normal Paragraph</p>
    <p class="s5">Web Technologies</p>
    <p class="s10">Web Technologies</p>
    <p class="s5">Web Technologies</p>

</body>
</html>

```



External Style Sheets:-

Third kind of style is external style sheet. In this total style elements are defined in a separate document, and this document is added to required web page. By using this we can use this style sheets in different web pages. As style sheet is separated from the document it gained the name External Style sheet.

Example:

Write one html program that calls a style sheet file.

Ourstyles.css

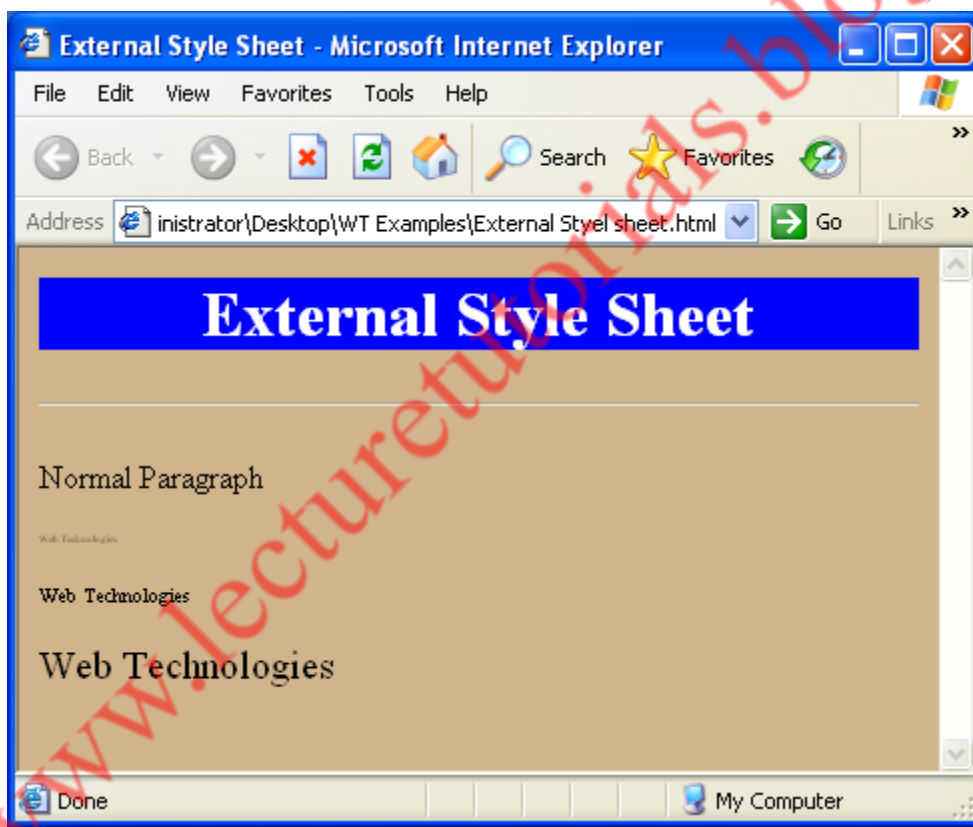
```

<link rel="stylesheet" type="text/css" href="ourstyles.css">
    P      { color:red;font-family:arial}
    .s5    { font-size:5;}
    .s10   { font-size:10;}
    .s15   { font-size:20;}
    H1     { color:white;background-color:blue}

```

External style sheet.html

```
<html>
<head>
<title>External Style Sheet</title>
<link rel="stylesheet" type="text/css" href="ourstyles.css">
</head>
<body bgcolor=tan>
    <h1 align="center">External Style Sheet</h1>
    <hr>
    <p>Normal Paragraph</p>
    <p class="s5">Web Technologies</p>
    <p class="s10">Web Technologies</p>
    <p class="s5">Web Technologies</p>
</body>
</html>
```



CSS Links

DEFINING STYLES FOR LINKS

As mentioned in the above table, there are four different selectors with respect to links. You can specify whatever style you'd like to each of these selectors, just like you'd do with normal text.

The four selectors are:

```
A:link
Defines the style for normal unvisited links.

A:visited
Defines the style for visited links.

A:active
Defines the style for active links.
A link becomes active once you click on it.

A:hover
Defines the style for hovered links.
A link is hovered when the mouse moves over it.
Note: Not supported by Netscape browsers prior to version 6.
```

PRACTICAL EXAMPLES

Here you can see a few examples on how CSS can be used to replace the traditional image based mouseover effects for links.

The hover style is not supported by Netscape browsers prior to version 6, but since it does no harm, you can still use it for the benefit of the +90% of visitors that arrive using MSIE).

One of the most common uses of CSS with links is to remove the underline. Typically it's done so that the underline appears only when a hover occurs. In the example below, we did just that. In addition we added a red color for hovered links.

Example: Hover

```
<style type="text/css">
A:link {text-decoration: none}
A:visited {text-decoration: none}
A:active {text-decoration: none}
A:hover {text-decoration: underline; color: red;}
</style>
```

Another example would be to create links that are both underlined and overlined.

Example: Underline/Overline

```
<style type="text/css">
A:link {text-decoration: none}
A:visited {text-decoration: none}
A:active {text-decoration: none}
A:hover {text-decoration: underline overline; color: red;}
</style>
```

A third example would be to create links that change in size when hovered.

Example: Size changing links

```
<style type="text/css">
A:link {text-decoration: none}
A:visited {text-decoration: none}
A:active {text-decoration: none}
A:hover {font-size:24; font-weight:bold; color: red;}
</style>
```

A final example would be to create links that have a permanent background color, obviously standing out from the rest.

Example: Background colored links

```
<style type="text/css">
A:link {background: #FFCC00; text-decoration: none}
A:visited {background: #FFCC00; text-decoration: none}
A:active {background: #FFCC00; text-decoration: none}
A:hover {background: #FFCC00; font-weight:bold; color: red;}
</style>
```

MULTIPLE LINKSTYLES ON SAME PAGE

The final topic deals with how to add multiple link styles that can be used on the same page.

In the above examples we addressed the HTML selector - A:link etc - and thus redefined the overall link style.

How do we define a link style that is only active in a certain area of the page?

The answer is: context dependent selectors.

Rather than addressing the A:link selector we will address it while being dependant on a certain outer class that surrounds the area where we'd like our link style to be effective.

For example:

```
<html>
<head>
<style type="text/css">
.class1 A:link {text-decoration: none}
.class1 A:visited {text-decoration: none}
.class1 A:active {text-decoration: none}
.class1 A:hover {text-decoration: underline; color: red;}

.class2 A:link {text-decoration: underline overline}
.class2 A:visited {text-decoration: underline overline}
.class2 A:active {text-decoration: underline overline}
.class2 A:hover {text-decoration: underline; color: green;}
</style>
</head>
</html>
```



```

</style>
</head>

<body>
ONE TYPE OF LINKS
<br>
<span class="class1">
<a href="http://www.yahoo.com">YAHOO</a>
<br>
<a href="http://www.google.com">GOOGLE</a>
</span>
<br>
<br>
ANOTHER TYPE OF LINKS
<br>
<span class="class2">
<a href="http://www.yahoo.com">YAHOO</a>
<br>
<a href="http://www.google.com">GOOGLE</a>
</span>
</body>
</html>

```

Note how we use the to define the context.

This is smart for two reasons:

- 1) The obvious, that it allows us to use different link styles on the same page, rather than being limited to using a single overall link style.
- 2) We can define entire areas where a certain link style works for all links within that area. Thus, we don't have to add a style definition to each and every link in that area.

CSS Layers

With CSS, it is possible to work with layers: pieces of HTML that are placed on top of the regular page with pixel precision.

The advantages of this are obvious - but once again Netscape has very limited support of CSS layers - and to top it off: the limited support it offers is quite often executed with failures.

So the real challenge when working with layers is to make them work on Netscape browsers as well.

LAYER BASICS

look at the code:

LAYER 1 ON TOP:

```
<div style="position:relative; font-size:50px; z-index:2;">LAYER  
1</div>  
<div style="position:relative; top:-50; left:5; color:red; font-size:80px;  
z-index:1">LAYER 2</div>
```

LAYER 2 ON TOP:

```
<div style="position:relative; font-size:50px; z-index:3;">LAYER  
1</div>  
<div style="position:relative; top:-50; left:5; color:red; font-size:80px;  
z-index:4">LAYER 2</div>
```

To create a layer all you need to do is assign the position attribute to your style. The position can be either absolute or relative.

The position itself is defined with the top and left properties.

Finally, which layer is on top is defined with the z-index attribute.

RELATIVE VERSUS ABSOLUTE POSITIONING

You can either position your layer calculated from the upper left corner(absolute) or calculated from the position where the layer itself is inserted (relative).

position:absolute

If you define the position to be absolute it will be calculated from the upper left corner of the page - unless the layer is defined inside another layer, in which case it will be calculated from the upper left corner of the parent layer.

position:relative

If you define the position to be relative it will be relative to the position of the tag that carries the style.

That is, if you add a relatively positioned layer in the middle of the page, then the position will be calculated from that exact spot in the middle of your page where it was added.

DEFINING THE POSITION

While the position property indicates the out spring of our coordinate system, the left and top properties defines the exact position of our layer.

You can enter both positive and negative values for these properties - thus it is possible to place content higher up and further to the left on the page than the logical position in the HTML code where the layer itself is defined.

In other words: at the bottom of your HTML code you can enter the code for a layer that is positioned at the top of the resulting page.

Both left and top properties can be dynamically changed with JavaScript.

This means that it is possible to move things around on the screen even after the page has finished loading.

In fact this technique can be (and has been) used to create entire games. Other uses might be menus that pop out when a mouse-over is detected on a link. The possibilities are endless - but in order to keep things simple, we will not dig into details about these dynamic HTML effects here.

POSITION IN THE STACK - THE Z-INDEX

Picture a game of 52 cards. If the ace of spades was at the bottom we'd say it had z-index:1;. If the queen of hearts was at the top we'd say she had z-index:52;.

Try looking at the code example at the top of this page again, and see how we used the z-index to put LAYER 1 on top in the first example, while we had LAYER 2 on top in the second example.

Very interesting possibilities arise from the fact that the z-index can be dynamically changed with JavaScript.

You could create several "pages" on top of each other - all on the same page. When the user clicks a link it will simply move the layer with the desired info on top rather than load a new page. The techniques to create effects like that goes beyond the scope of pure CSS however, so for now we will just refer to DHTML (Dynamic HTML - a mix between JavaScript and CSS) for further explorations into that area.

VISIBLE VERSUS HIDDEN LAYERS

A final property is the visibility property that will allow you to create invisible layers.

Why would anyone want to create an invisible layer? Well, imagine the possibilities it gives for adding pop-up menus and other cool effects on your pages.

With dynamic HTML it is possible to change the visibility of a layer according to certain

events. The most common use of this is to create menus that pop out (like the sub menus in the START menu on Windows). The trick behind these menus is to create all submenus as invisible layers. Then, when a mouse-over is detected on a link the according layer becomes visible. (Sounds pretty easy - actually is pretty easy - except when tried on Netscape browsers that seem to have only a vague idea of the logic behind CSS layers).

Valid values for the visibility property are: visible and hidden.

This example shows how to create an invisible layer:

```
<div style="position:relative; visibility:hidden;">HELLO!!!</div>
```

PRACTICAL USE OF LAYERS

It's obvious that layers offer certain possibilities for precise positioning of static elements on your pages.

In reality layers are often used in more dynamic ways:

- Flying elements/banners on the page
- Games where you move an object around
- Menus that pop out when triggered
- Menus that become visible when triggered

While all of these effects might seem pretty cool and useful - the fact is that the web is filled with dynamic effects that are much more cool than the average visitor really likes.

The more you create a unique interface for your site the more you force the visitor to forget about what she is used to. Do not underestimate the power of sticking to the elements that the average visitor is accustomed to.

What's cool about creating an effect that makes 90% of all web designers clap their hands while leaving 90% of non-web designers confused or disappointed?

In any case, judge for yourself if a certain effect is really needed - and if so: do not hesitate to use it.


CSS CURSORS


Microsoft Internet Explorer 4+ and Netscape 6+ supports customized cursors defined with CSS.


Although the cursors will not have the customized look in other browsers it usually doesn't ruin anything. These browsers will simply show the normal arrow-cursor which would be same case as if you refrained from customizing cursors at all.


So unless the page really doesn't work without the customized cursor there shouldn't be technical reasons for choosing not to.


However there might be other reasons for thinking twice before adding custom cursor to your pages. Many users are easily confused or irritated when a site breaks the standard user interface.


 default=cursor:default


 crosshair=cursor:crosshair


 hand=cursor:hand


 pointer=cursor:pointer


 Cross browser=cursor:pointer;cursor:hand


 move=cursor:move


 text=cursor:text


 wait=cursor:wait


 help=cursor:help


 n-resize=cursor:n-resize


 ne-resize=cursor:ne-resize


 e-resize=cursor:e-resize


 se-resize=cursor:se-resize


 s-resize=cursor:s-resize


 sw-resize=cursor:sw-resize


 w-resize=cursor:w-resize


 nw-resize=cursor:nw-resize


 progress=cursor:progress


 not-allowed=cursor:not-allowed


 no-drop=cursor:no-drop

 vertical-text=cursor:vertical-text

 all-scroll=cursor:all-scroll

 col-resize=cursor:col-resize

 row-resize=cursor:row-resize

 cursor:url(uri)=cursor:url(uri
)

ADDING A CUSTOMIZED CURSOR

The syntax for a customized cursor is this:
(Position the mouse over each link to see the effect)

Selector {cursor:value}

For example:

```
<html>
<head>
<style type="text/css">
.xlink {cursor:crosshair}
.hlink{cursor:help}
</style>
</head>

<body>
<b>
<a href="mypage.htm" class="xlink">CROSS LINK</a>
<br>
<a href="mypage.htm" class="hlink">HELP LINK</a>
</b>
</body>
</html>
```

REDEFINING THE CURSOR FOR ENTIRE PAGES

If you want to redefine the cursor so that it's not only showing up when moved over a link, you simply specify the desired cursor using the body-selector.

For example:

```
<html>
<head>
<style type="text/css">
body { cursor:crosshair}
</style>
</head>

<body>
<b>
SOME TEXT
<br>
<a href="mypage.htm">ONE LINK</a>
<br>
<a href="mypage.htm">ANOTHER LINK</a>
</b>
</body>
</html>
```

REDEFINING THE CURSOR FOR AREAS ON A PAGE

If you want one look of the cursor in one area of the page and another look of the cursor in another area you can do it with context dependant selectors.

This way, you create different styles for links, that are dependant on the context. Now if the context is set with a dummy tag, such as you don't have to specify the desired style each and every time there is a link in the section.

For example:

```
<html>
<head>
<style type="text/css">
.xlink A{ cursor:crosshair}
.hlink A{ cursor:help}
</style>
</head>

<body>
<b>
<span class="xlink">
<a href="mypage.htm">CROSS LINK 1</a><br>
<a href="mypage.htm">CROSS LINK 2</a><br>
<a href="mypage.htm">CROSS LINK 3</a><br>
</span>
<br>
<span class="hlink">
<a href="mypage.htm">HELP LINK 1</a><br>
```

```
<a href="mypage.htm">HELP LINK 2</a><br>  
<a href="mypage.htm">HELP LINK 3</a><br>  
</span>  
</b>  
</body>  
</html>
```

www.lecturetutorials.blogspot.com