3. HYPERTEXT MARKUP LANGUAGE

MARKUP LANGUAGES

- Markup languages are designed for the processing, definition and presentation of text.
- The language specifies code for formatting, both the layout and style, within a text file. The code used to specify the formatting is called **tags**.
- With ML, you simply "mark up" a text document with tags that tell Web browser how to structure it to display.
- Types of Markup Languages:
  - SGML
  - HTML
  - XML
  - XHTML

SGML

- SGML (Standard Generalized Markup Language) is a standard for how to specify a document markup language or tag set.
- SGML is not in itself a document language, but a description of how to specify one. It is metadata.
- SGML is based on the idea that documents have structural and other semantic elements that can be described without reference to how such elements should be displayed.
- The actual display of such a document may vary, depending on the output medium and style preferences.
- Some advantages of documents based on SGML are:
  - They can be created by thinking in terms of document structure rather than appearance characteristics (which may change over time).
  - They will be more portable because an SGML compiler can interpret any document by reference to its document type definition (DTD).
  - Documents originally intended for the print medium can easily be re-adapted for other media, such as the computer display screen.

HTML

- HTML or Hypertext Markup Language is the most widely used language on Web.
- Technically, HTML is not a programming language, but rather a markup language.
- It is the most widely used language to write Web Pages.
- HTML files has extension: filename.html [or] filename.htm

XML

- XML (Extensible Markup Language) is a flexible way to create common information formats and share both the data on the World Wide Web, intranets and anywhere.
- It is similar to HTML and was designed to describe data.
- XML files has extension: filename.xml

XHTML

- Extensible Hypertext Markup Language (XHTML) is a family of XML markup languages that mirror or extend versions of the widely used HTML, the languages in which web pages are formulated.

HTML INTRODUCTION

HTML stands for **Hypertext Markup Language**.

- **Hypertext** refers to the way in which Web pages (HTML documents) are linked together. When you click a link in a Web page, you are using hypertext. It supports font styled text, pictures, graphics and animations and also it provides hyperlinks that used to browse.
- **Markup Language** describes how HTML works. A markup language is a set of **markup tags.**

HTML is a language for web pages and the output of the programs can be seen after using browser. SGML was used as the basis of HTML.

Tags of HTML

HTML is a **tag-based language** as ML's use tags for formatting the text. Each HTML tag describes different document content. HTML supports **WYSIWYG** (stands for What You See is What You Get) displays as it is what information you are observing on the screen will appears in the output. A tag is made up of *left operator (<), a right operator (>) and a tag_name* between the operators.

The simplest tag is nothing more than a name appropriately enclosed in brackets. More complicated tags contain one or more attributes, which specify or modify the behavior of the tag.

If you forget to mention the right operator (>) or give any space between the left operator and tag_name, browser does not consider it as a tag. If the browser not understands the tag_name, it just ignores it, browser won't generate any error.

Syntax:

```
<tag_name[parameter(s)[=value]]>
```

Example:

<hr> is tag that displays a horizontal ruler line

<hr/> Tag with no parameters

<hr align="center"/> Tag with parameter "align" and value for parameter as "center"

<hr width="30%" size=100 align=right/> Tag with more parameters (width, size, align)

HTML Tags are not case sensitive. There's no difference in effect between <hr>, <Hr>, <HR>, or even <hR>; they are all equivalent.

Tags are classified into 2 types:

- **SINGLETON Tags**
  - These tags do not require an ending tag.
  - Example: <hr>,
, <img>, etc
- **PAIRED Tags**
  - These tags require an ending tag which is exactly same as opening tag except backslash (/) before the tag name.
  - Syntax: <tagname>content</tagname>
  - Example:
    - <html> - starting tag or opening tag
    - </html> - terminating tag or ending tag

HTML Versions

Since the early days of the web, there have been many versions of HTML.

Version	Year
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML4.01	1999
XHTML	2000
HTML5	2012

Creating an HTML Page

The following steps are needed to create a HTML Page:

1. Open any text editor program Notepad, Edit etc.
2. Use the option File → New to create a new document.

3. Type the HTML code.

```
<!DOCTYPE html>
<html>
      <head>
            <title>My First Web Page</title> ←  } HEAD SECTION
      </head>
      <body>
            HELLO EVERYBODY.                  ←  } BODY SECTION
            THIS IS MY FIRST WEB PAGE.
      </body>
</html>
```

4. Save the file as<filename>.html (Example: mypage.html)
5. View the HTML page in a Web Browser. (Double-click your saved HTML file)

The HTML page designing contains two sections:

- Head Portion
- Body Portion

Common Tags

TAGS	DESCRIPTION
<html>…</html>	In order to identify that the document is HTML page or not. If the browser finds <html> tag, it displays the content. Describes an HTML Document.
<head>…</head>	It provides information about the document. This tag represents the document's header which can keep HTML tags like <title>, <link> etc.
<title>…</title>	The <title> tag is used inside the <head> tag to mention the document title. Displayed on Title Bar of Browser.
<body>…</body>	This tag represents the document's body which keeps other HTML tags. It describes the visible page content.
<!DOCTYPE…>	This tag defines the document type and HTML version. This declaration helps the browser to display a web page correctly. To display a document correctly, the browser must know both type and version.

<!DOCTYPE> Declaration

The doctype declaration is not case sensitive. All cases are acceptable:

```
<!DOCTYPE html>
<!DOCTYPE HTML>
<!doctype html>
<!Doctype Html>
```

Common Declarations:

HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTDHTML4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

HTML5

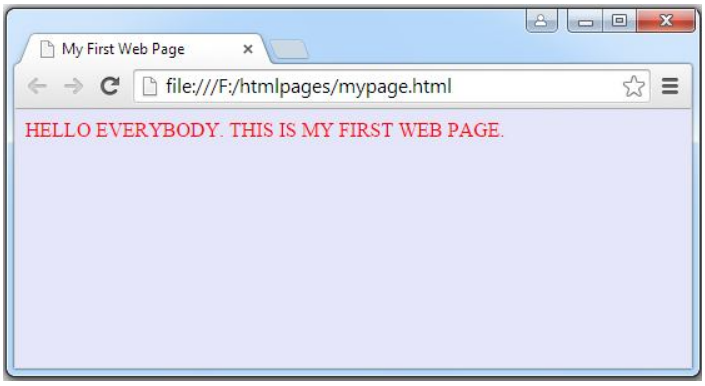
```
<!DOCTYPE html>
```

<BODY> Tag

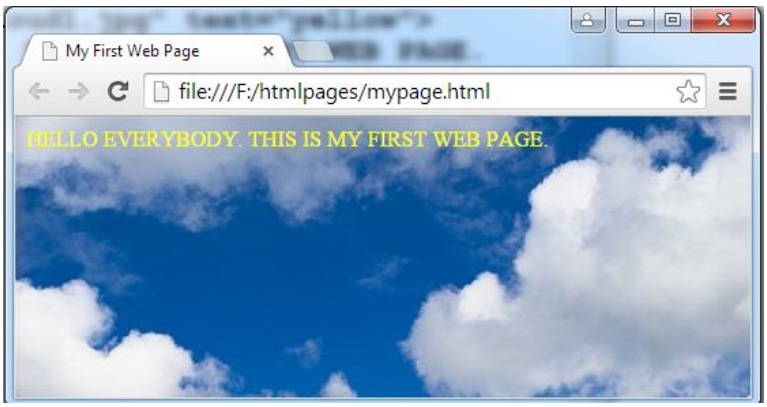
Attributes for <body> tag

Attribute	Value	Description
alink	color	Specifies the color of an active link in a document
background	URL	Specifies a background image for a document
bgcolor	color	Specifies the background color of a document
link	color	Specifies the color of unvisited links in a document
text	color	Specifies the color of the text in a document
vlink	color	Specifies the color of visited links in a document

```
<body bgcolor="blue" text = "green">
```




```
<body background="cloud.jpg" text="yellow">
```



Classification of HTML Tags

Basic Tags

- General Tags, useful for text styles, paragraphs, headers, etc

Cosmetic Tags

- Provides additional look and feel for text i.e., Bold, Italic, Underline, etc

Block Format Tags

- Format the given information, as list of items or tables.

User Interface Tags

- Allows user to enter some information into web page, using forms.

Other Tags

- Break up information like Break line, horizontal line etc

Header Tags

HTML has 6 levels of predefined headings numbered 1 through 6. They are paired tags. Header tags are simple forms of text formatting. The text displayed by Header tags will be Bold by default.

Syntax:

```
<h[n=1to6] [align={left|right|center}]>
TEXT
</hn>
```

Center Tag

<CENTER> tag centers all the information content in between them. It is also a paired tag.

Example: Write a HTML program to demonstrate Header tags

```
<html>
      <head>
            <title>Header Levels</title>
      </head>
      <body>
            <center>
                  <h1>Header Level 1: WEB TECHNOLOGY</h1>
                  <h2>Header Level 2: WEB TECHNOLOGY</h2>
                  <h3>Header Level 3: WEB TECHNOLOGY</h3>
                  <h4>Header Level 4: WEB TECHNOLOGY</h4>
                  <h5>Header Level 5: WEB TECHNOLOGY</h5>
                  <h6>Header Level 6: WEB TECHNOLOGY</h6>
            </center>
      </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 indicate these tags one by one before text.

…	Bold Text
<u> … </u>	Underline Text
…	Italic Text
…	Emphasized Text (Italics)
…	Strong or Bold Text
<s> … </s> (Or) <del> … </del>	Strikes the Text
<samp> … </samp>	Sample Output from a program
<var> … </var>	Defines a variable
<address> … </address>	Like Address Model (Italic)
<pre> … </pre>	Considers space, new lines, etc. Preformatted Text.

HTML allows nesting of tags, but you should be careful about the ending tag sequence. Tag within another tag is called nesting of tags.

Example:

<B><u>HTML</u></B> Output: ***HTML***

Subscripts and Superscripts

HTML tags provide some tags that are useful for making the given text to be as subscript text and superscript text.

Subscript text means the information displayed in the beneath portion of given word. You can observe this kind of text in chemical formulas like water (H_2O), carbon dioxide (CO_2), etc.

Superscript text means the information displayed above the given text which mainly you can observe in mathematical formulas like square (a^2), cubes (c^3) etc.
The tag used for Subscript text is **_…** and for superscript text is **[…]**
Example:

```
H<SUB>2</SUB>SO<SUB>4</SUB>          Output: H2SO4
a<SUP>2</SUP> + b<SUP>3</SUP>        Output: a2 + b3
```

Text Fonts
HTML provides <FONT> tag for facilitating various types of fonts. The fonts can be varied as per the customized needs for given values using the … tag pair.

Syntax:
```
<font [SIZE=+n|-n] [COLOR=color] [FACE=fontname]>
......
</font>
```

The font size is taken basis of Base font size, which is default size of 8. Accordingly –n or +n makes the size increase or decrease respectively. Size ranges from -7 to +7. The font face is used to change the font of the text you are formatting. Font face is nothing but font name for the text i.e., like Courier New, Arial, Times New Roman (default), Verdana, etc.

```
<font size="7" color="blue" face="Verdana">
     WELCOME TO ALL
</font>
```

<p> tag also called as paragraph tag, indicates beginning of a paragraph. A paragraph is marked up as follows:

```
<p>This is some text in a paragraph.</p>
```

Browsers automatically add some space (margin) before and after each <p> element.
Attribute of <p> tag: align= left / right / center / justify

**
 tag** also called as line break tag, forces the text to begin on new line. In HTML, the
 tag has no end tag.

<hr> tag
This tag displays horizontal rule in the web page, it includes line break. In HTML, the <hr> tag has no end tag. The <hr> element is used to separate content (or define a change) in an HTML page.

Attribute	Value	Description
align	left center right	Specifies the alignment of a <hr> element
noshade	noshade	Specifies that a <hr> element should render in one solid color (noshaded), instead of a shaded color

size	*pixels*	Specifies the height of a <hr> element
width	*pixels %*	Specifies the width of a <hr> element

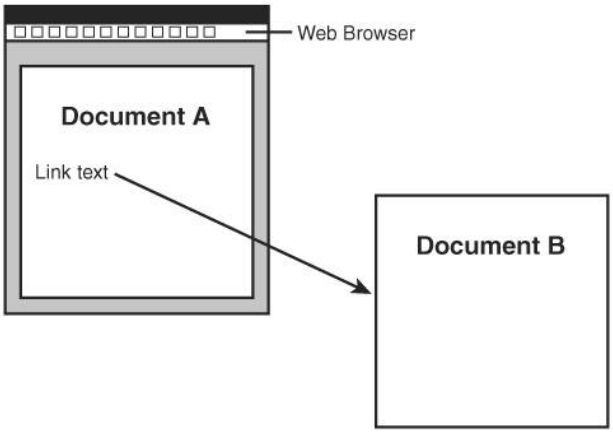
Example: Write a HTML program to demonstrate Text Styles

```
<html>
      <head>
            <title>Text Styles</title>
      </head>
      <body>
            <h1 align="left">TEXT STYLES</h1>
            <p   align="justify"><i>Subscript   text</i>   means   the
information displayed in the beneath portion of given word. You can
observe   this   kind   of   text   in   chemical   formulas   like   water
<b>H<sub>2</sub>O)</b>,   carbon   dioxide   (CO<sub>2</sub>),   etc.
<b><u>Superscript</u></b> text means the information displayed above
the given text which mainly you can observe in mathematical formulas
like square (a<SUP>2</SUP>), cubes (c<SUP>3</SUP> ) etc.</p>
            <hr>
            <em>Emphasized text</em><br><br>
            <strong><font size="4" face=Verdana color="blue">
                  Text Font: Verdana, size: +4, and color="Blue"</font>
            </strong><br><br>
            <s> Strikes the Text </s> <br>
            <del> Strikes the Text </del><br><br>
            <code>
                  for(int i=1;i<10;i++) {
                        printf(i + "\t");
                  }
            </code><br><br>
            <samp>Sample output from a computer program</samp><br><br>
            <var>Variable</var><br><br>
            <address>
                  Anil Neerukonda Institute of Technology & Sciences<br>
                  Sangivalasa,
                  Bheemunipatnam (Mandal),
                  Vishakapatnam (District),
                  Andhra Pradesh, India
            </address>
            <pre>
                  for(int i=1;i<10;i++) {
                        printf(i + "\t");
                  }
            </pre>
      </body>
</html>
```

Output:

Linking in HTML

Text becomes hypertext with the addition of links which connect separate locations within 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 becomes easier because of links. Information in the same page also connected through links (Internal links).

Anchor tag (<a>…) is used for creating links. Minimum it requires a parameter i.e., HREF, which indicates the destination document. Other parameters are name and target can be useful for identification for anchor tag and target specifies where to open the linked document (_blank, _parent, _self, _top, framename). Name and target tag are optional.

By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

Syntax:

```
<a href="document" name="id" target="_blank">
     LINK NAME
</a>
```

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

```
<a href="http://www.google.co.in">GOOGLE</a>
```

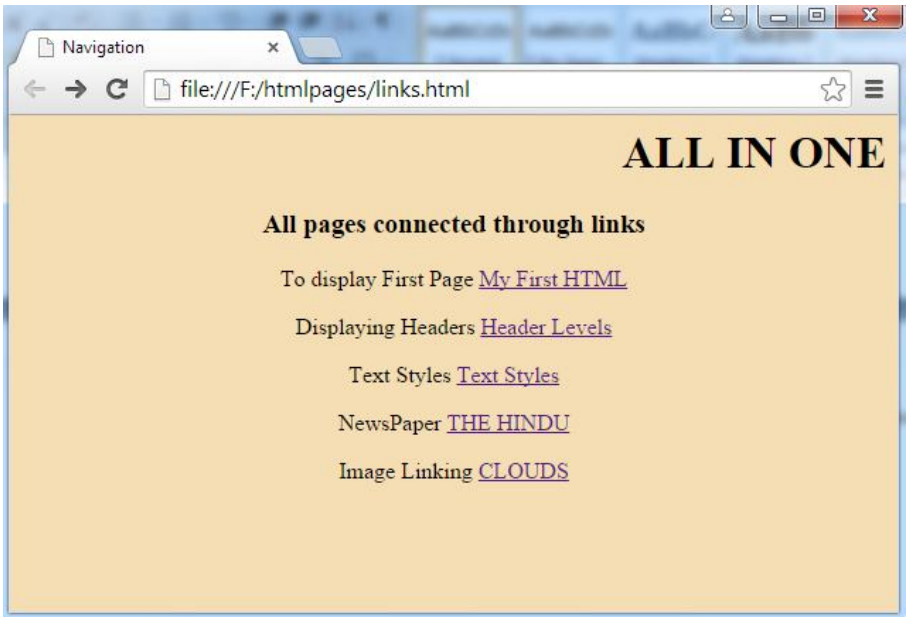
When the user clicks on Google, Google website is displayed on the browser.

Example: Create a HTML web page that connects web pages created through programs and external web sites.

```
<html>
<head>
     <title>Navigation</title>
</head>
<body bgcolor=wheat>
     <h1 align=right>ALL IN ONE</h1>
     <h3 align=center>All pages connected through links</h3>
     <center>
     <p>To display First Page
          <a href="mypage.html">My First HTML</a></p>
     <p>Displaying Headers
          <a href="headers.html" target="_self">Header Levels</a></p>
     <p>Text Styles
     <a href="textstyles.html" target="_blank">Text Styles</a></p>
```

```
            <p>NewsPaper
                  <a href="http://www.thehindu.com">THE HINDU</a></p>
            <p>Image Linking <a href="cloud.jpg">CLOUDS</a></p>
      </center>
</body>
</html>
```

Output:



IMAGES

Images can be embedded inside HTML documents using the <img> tag. Also we can specify "align" parameter for it. Default alignment is center. All browsers support all file formats for images (.gif, .jpg, .png, .bmp, etc) and either of them can be used. Images can be used as links where you click an image to follow a link. Images can include as background for the entire web page by using background parameter in <body> tag.

<img> Tag

By using <img> tag we can include images on the web page. It has several attributes like align, src, width, and height, etc.

Attribute	Value	Description
align	Top, bottom, middle, left, right	Specifies the alignment of an image according to surrounding elements
alt	*text*	Specifies an alternate text for an image
border	*pixels*	Specifies the width of the border around an image
height	*pixels*	Specifies the height of an image
hspace	*pixels*	Specifies the whitespace on left and right side of an image

src	*URL*	Specifies the URL of an image
vspace	*pixels*	Specifies the whitespace on top and bottom of an image
width	*pixels*	Specifies the width of an image

Syntax:

```
<img src="filename/path" alt="Text Message"
                  border=n, height="h" width="w">
```

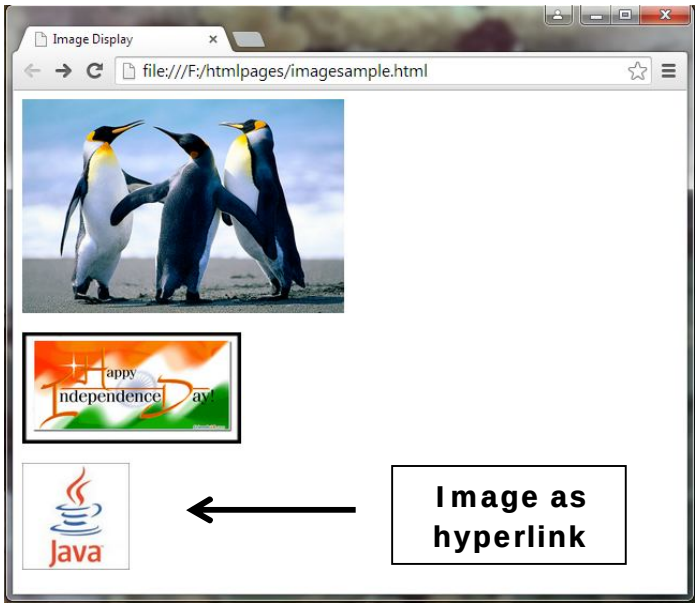
If you specify anchor tag <a> before the <img> tag, then image can be treated as hyperlink.

```
<a href="mypage.html"><img src="cloud.jpg" border=2 height=200></a>
```

Example: Program to illustrate the img tag

```
<html>
<head>
<title>Image Display</title>
</head>
<body>
   <img src="C:\Users\Public\Pictures\Sample Pictures\Penguins.jpg"
          alt="Pengunis" width="300" height="200"><br><br>
   <img src="45.gif" alt="Happy Independence Day" border="2"
          width="200" height="100"><br><br>
   <a href="http://www.oracle.com/technetwork/java/index.html">
     <img src="java.jpg" alt="Link to Java Page" width="100"
          height="100">
   </a>
</body>
</html>
```

Output:



Lists:

List is a collection of items and they may be **ordered** or **unordered**. In Microsoft Word you may be familiar with the term bullets, which are called as unordered lists. **Ordered Lists** are nothing but sequenced lists which are numbered or indicated with alphabets. Other lists included as directory lists, definition lists that are very useful to make the web page attractive.

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 <UL> tag does Creation of unordered lists in HTML which is a paired tag, so it requires in ending tag that is </UL>.

The lists of items are included in between <UL>…</UL>. The TYPE attribute can also be added to the <UL> tag that indicates the displayed bullet along with list of item is square, disc or circle. By ***default it is disc.***

Syntax:

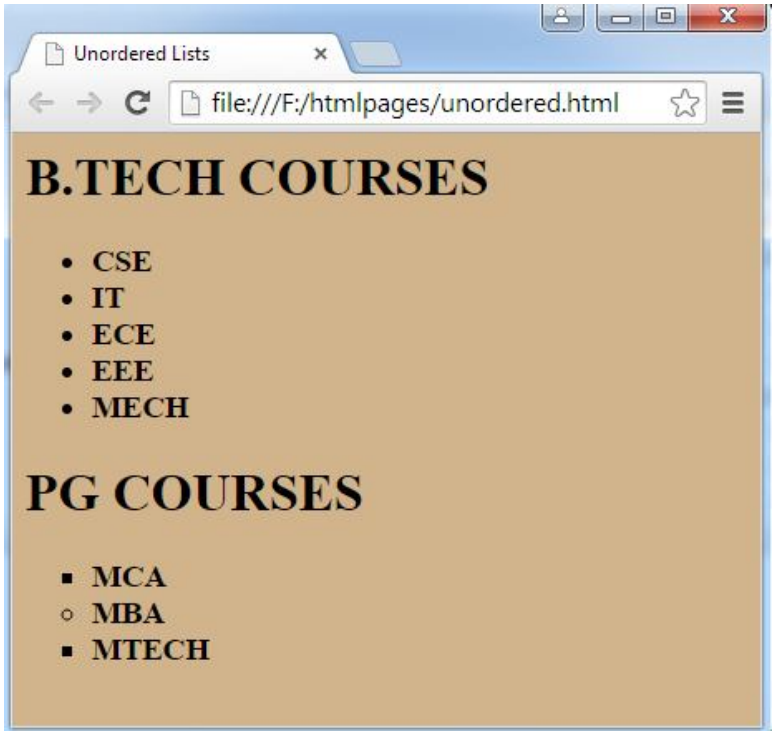
```
<UL> [TYPE={square, disk or circle} ]>
     <LI> item name1
     <LI> item name2
     ----------------------
     ----------------------
     <LI> item namen
</UL>
```

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>MTECH
                  </ul>
            </h3>

      </body>
</html>
```

Output:



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 sometimes 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 ordered list should start with the <OL> element, which is immediately followed by a <LI> element which is same as <LI> in unordered list. End of ordered lists is specified with ending tag </OL>.

Different Ordered list types like Roman numeral list, alphabet list etc can be specified with TYPE tag. Another optional parameter with <OL> 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<LI>, changes the list type for particular item. To give more flexibility to list, we can use VALUE

parameter with <LI> 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

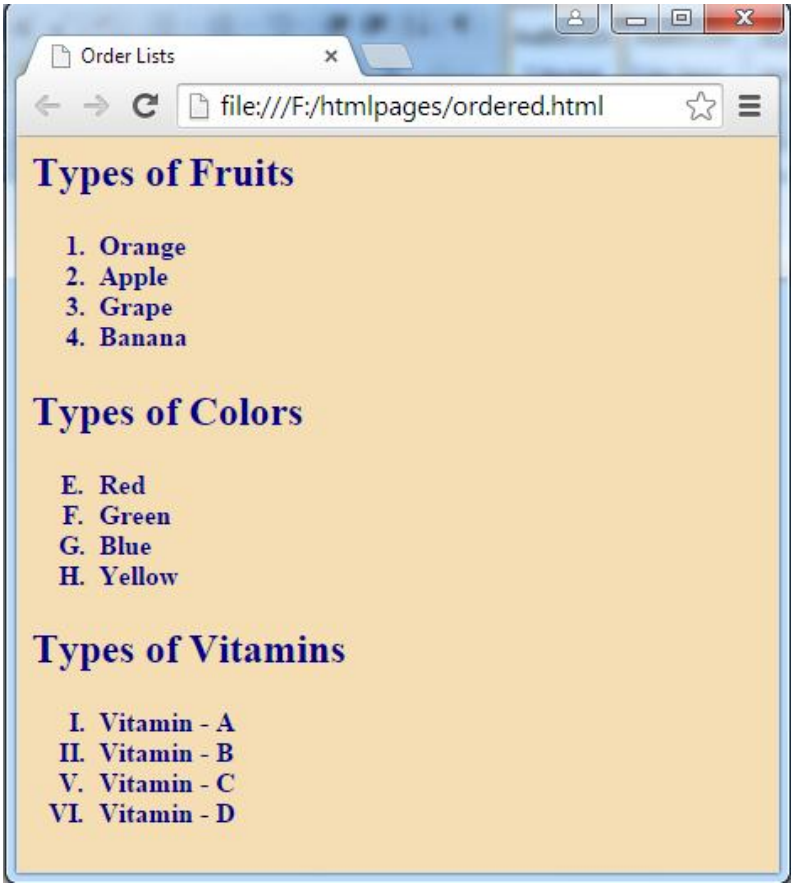
Type=”1” (default) e.g. 1, 2, 3, 4 ……
Type=”A” Capital letters e.g. A, B, C, ….
Type=”a” Small letters e.g., a, b, c …….
Type=”I” Large roman letters e.g. I, II, III, ……

Example: Write a program for displaying database application packages as numbered list and Microsoft packages as alphabet lists starts with E. Display types of formatting tags by use VALUE parameter.

```
<html>
     <head>
          <title> Order Lists </title>
     </head>
     <body bgcolor="wheat" text="darkblue">
     <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>
     <h2>Types of Vitamins
```

```
      <h4>
            <OL Type="I">
                  <LI>Vitamin - A
                  <LI>Vitamin - B
                  <LI value=5>Vitamin - C
                  <LI>Vitamin - D
            </OL>
      </h4>
      </body>
</html>
```

Output:



Other Lists: -

There are several other 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.

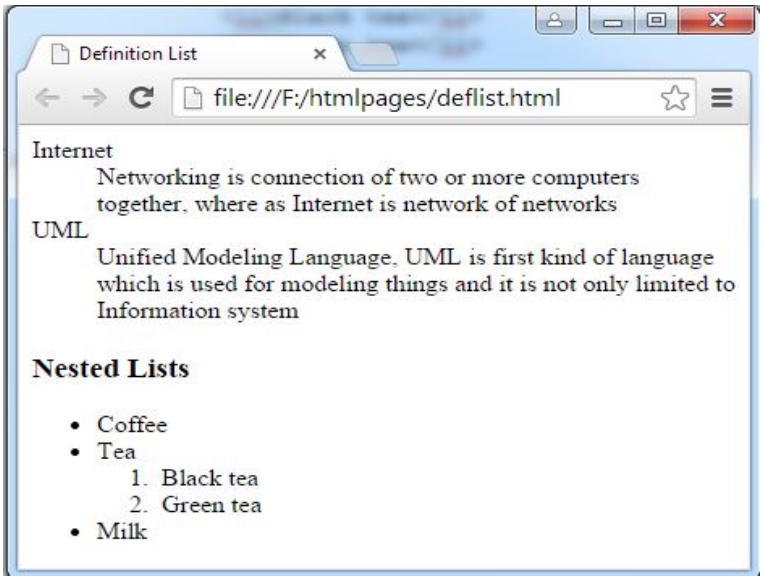
To create a definition it must start with <DL> and immediately followed by the first definition term <DT>.

Nested Lists: Lists can be nested that is Nested Lists is list with in another list.

Example: Write a HTML program to demonstrate Definition List and Nested List

```
<html>
      <head>
            <title> Definition List </title>
      <body>
            <DL>
            <DT>Internet
                  <DD>Networking is connection of two or more computers
                  together, where as Internet is network of networks
            <DT>UML
                  <DD>Unified Modeling Language, UML is first kind of
                  language which is used for modeling things and it is
                  not only limited to Information system
            </DL>
            <h3>Nested Lists</h3>
            <ul>
                  <li>Coffee</li>
                  <li>Tea
                        <ol>
                              <li>Black tea</li>
                              <li>Green tea</li>
                        </ol>
                  </li>
                  <li>Milk</li>
            </ul>
      </body>
</html>
```

Output:



HTML Tables: -

Tables are one of the important concepts while displaying information on web page. This is another way of formatting text. The biggest advantage of using tables on the web pages is that the information gets arranged systematically. There are several types of tables come across out day-to-day life like Railway Time Table, Calendar, Shopping list etc.

Tag Representation:-

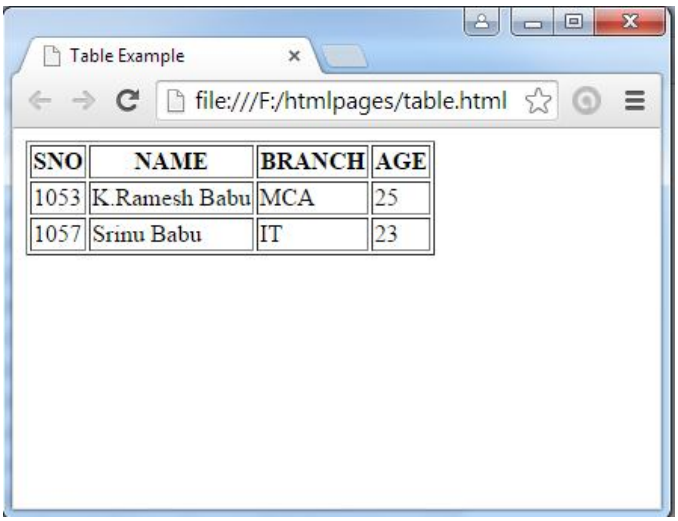
Tag	Description
<table></table>	Indicates start of a table Attributes : Align, background, bgcolor, border, bordercolor, cols, height, width
<thead></thead>	Starts the header part of table, and it should followed by <TH> Attributes : Align, bgcolor, valign,
<tbody></tbody>	Groups the body content in a table. Starts the main area of table, says actual data is started Attributes: align, bgcolor, valign.
<TR></TR>	Indicates starting of a table row Attributes: Align, bgcolor, bordercolor, valign.
<TH></TH>	Used to give table heading (column headers) Attributes: Align, background, bgcolor, bordercolor, height, valign, width.
<TD></TD>	Actual table data for a cell, for each cell we should give <TD> Attributes: Align, background, bgcolor, bordercolor, height, scope, valign, width.
<tfoot></tfoot>	Used to define foot information of the table. Attributes: Align, bgcolor, valign.
<caption></caption>	To insert caption in to table, inserts caption data directly above the table, caption text is used to help text-based browsers interpret the table data. Attributes: Align, Valign.

With the help of <Table>,<TH>,<TR>,<TD> we can create tables, but to give more clear picture for which is header and which is footer it is better to use <THEAD> and <TFOOT> tags. Header and Footer tags useful when the table may be placed over a number of pages, but these tags are not as yet widely implemented.

Example: HTML Program to demonstrate Table Example

```
<html>
     <head>
          <title> Table Example </title>
     </head>
```

```
    <body>
        <table Border=1>
        <THEAD>
            <TR><TH>SNO<TH>NAME<TH>BRANCH<TH>AGE</TR>
        </THEAD>
        <TBODY>
            <tr>
                <Td>1053<TD>K.Ramesh Babu<TD>MCA<TD>25
            </tr>
            <tr>
                <td>1057<td>Srinu Babu<td>IT<td>23
            </tr>
        </TBODY>
        </table>
    </body>
</html>
```



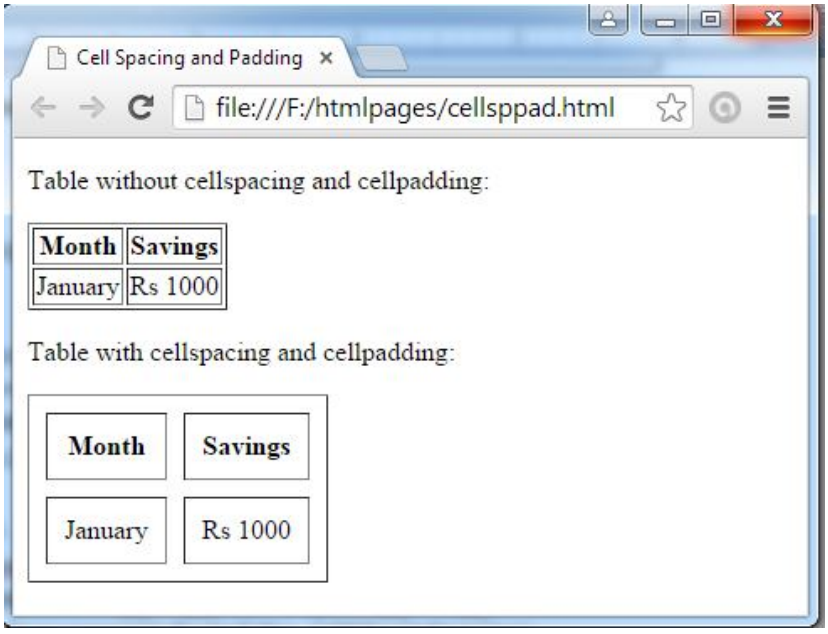
Attributes

Attribute	Value	Description
align	left center right	Specifies the alignment of a table according to surrounding text
bgcolor	*rgb(x,x,x)* *#xxxxxx* *colorname*	Specifies the background color for a table
border	1 0	Specifies whether the table cells should have borders or not
cellpadding	*pixels*	Specifies the space between the cell wall and the cell content
cellspacing	*pixels*	Specifies the space between cells
frame	Void, above, below, hsides lhs, rhs, vsides box, border	Specifies which parts of the outside borders that should be visible

rules	None, groups rows, cols, all	Specifies which parts of the inside borders that should be visible
summary	*text*	Specifies a summary of the content of a table
width	*pixels* *%*	Specifies the width of a table

Example: HTML Program to demonstrate Cell Spacing and Cell Padding attributes

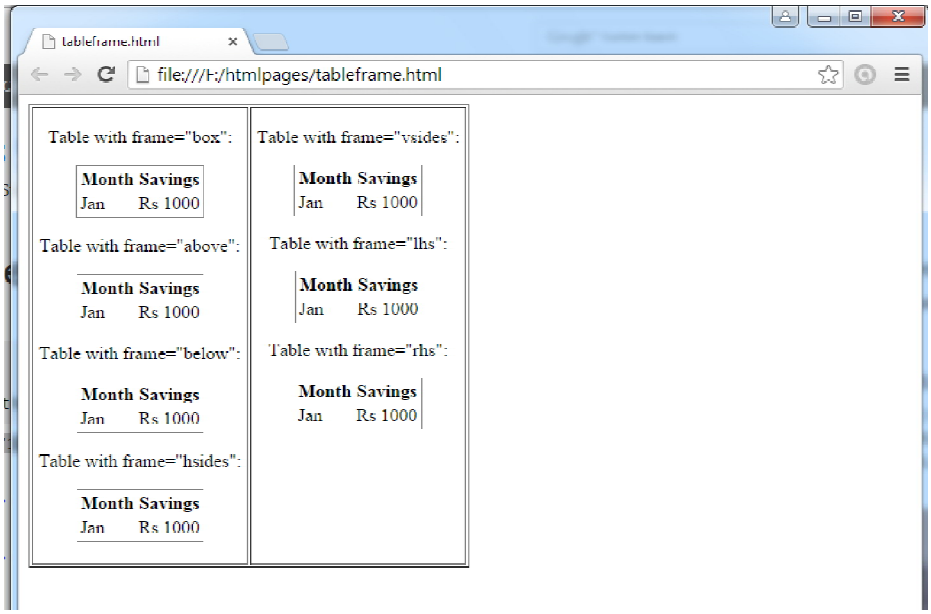
```
<!DOCTYPE html>
<html>
<head>
<title>Cell Spacing and Padding</title>
</head>
<body>
     <p>Table without cellspacing and cellpadding:</p>
     <table border="1">
          <tr><th>Month</th><th>Savings</th></tr>
          <tr><td>January</td><td>Rs 1000</td></tr>
     </table>
     <p>Table with cellspacing and cellpadding:</p>
     <table cellspacing="10" cellpadding="10" border="1">
          <tr><th>Month</th><th>Savings</th></tr>
          <tr><td>January</td><td>Rs 1000</td></tr>

     </table>
</body>
</html>
```



Example: HTML Program to demonstrate Frame Attribute

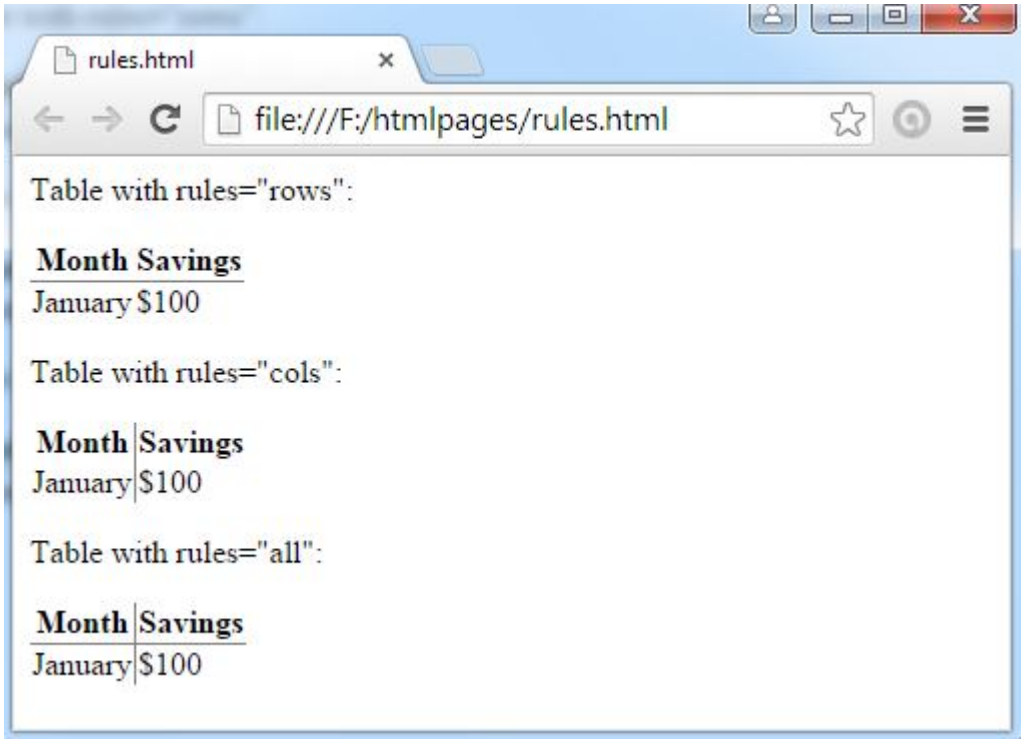
```
<!DOCTYPE html>
<html>
     <body>
          <table width="50%" border=1>
          <tr>
               <td align=center>
                    <p>Table with frame="box":</p>
                    <table frame="box">
                         <tr><th>Month</th><th>Savings</th></tr>
                         <tr><td>Jan</td><td>Rs 1000</td></tr>
                    </table>
                    <p>Table with frame="above":</p>
                    <table frame="above">
                         <tr><th>Month</th><th>Savings</th></tr>
                         <tr><td>Jan</td><td>Rs 1000</td></tr>
                    </table>
                    <p>Table with frame="below":</p>
                    <table frame="below">
                         <tr><th>Month</th><th>Savings</th></tr>
                         <tr><td>Jan</td><td>Rs 1000</td></tr>
                    </table>
                    <p>Table with frame="hsides":</p>
                    <table frame="hsides">
                         <tr><th>Month</th><th>Savings</th></tr>
                         <tr><td>Jan</td><td>Rs 1000</td></tr>
                    </table><br/>
               </td>
               <td align=center valign=top>
                    <p>Table with frame="vsides":</p>
                    <table frame="vsides">
                         <tr><th>Month</th><th>Savings</th></tr>
                         <tr><td>Jan</td><td>Rs 1000</td></tr>
                    </table>
                    <p>Table with frame="lhs":</p>
                    <table frame="lhs">
                         <tr><th>Month</th><th>Savings</th></tr>
                         <tr><td>Jan</td><td>Rs 1000</td></tr>
                    </table>
                    <p>Table with frame="rhs":</p>
                    <table frame="rhs">
                         <tr><th>Month</th><th>Savings</th></tr>
```

```
                    <tr><td>Jan</td><td>Rs 1000</td></tr>
                </table>
            </td>
        </tr>
        </table>
    </body>
</html>
```



Example: HTML Program to demonstrate Rules Attribute

```
<html>
    <body>
        <p>Table with rules="rows":</p>
        <table rules="rows">
            <tr><th>Month</th><th>Savings</th></tr>
            <tr><td>Jan</td><td>Rs 1000</td></tr>
        </table>
        <p>Table with rules="cols":</p>
        <table rules="cols">
            <tr><th>Month</th><th>Savings</th></tr>
            <tr><td>Jan</td><td>Rs 1000</td></tr>
        </table>
        <p>Table with rules="all":</p>
        <table rules="all">
            <tr><th>Month</th><th>Savings</th></tr>
            <tr><td>Jan</td><td>Rs 1000</td></tr>
        </table>
    </body>
</html>
```



Colgroup: (Complex HTML Tables and Formatting)

HTML provides many additional features for tables. The <COLGROUP> element is used to group and format columns. Each COL element in the <COLGROUP>…..</COLGROUP> tags can format any number of columns, which specified with SPAN attribute.

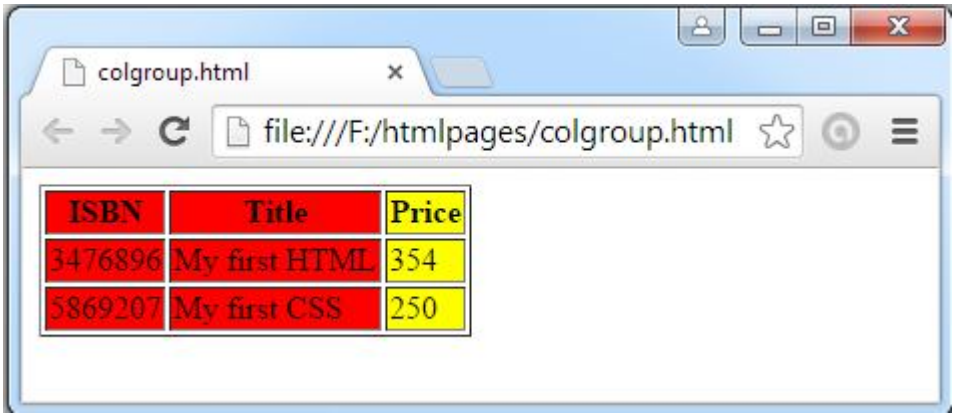
The <colgroup> tag specifies a group of one or more columns in a table for formatting. The <colgroup> tag is useful for applying styles to entire columns, instead of repeating the styles for each cell, for each row. The <colgroup> tag must be a child of a <table> element, after any <caption> elements and before any <thead>, <tbody>, <tfoot>, and <tr> elements.

```
<COLGROUP>
      <COL ALIGN=”center”>
      <COL SPAN=”2” ALIGN=”left”>
</COLGROUP>
```

Example: HTML Program to demonstrate Colgroup Formatting

```
<html>
<body>
<table border="1">
  <colgroup>
    <col span="2" style="background-color:red">
    <col style="background-color:yellow">
  </colgroup>
  <tr>
```

```
    <th>ISBN</th>
    <th>Title</th>
    <th>Price</th>
  </tr>
  <tr>
    <td>3476896</td>
    <td>My first HTML</td>
    <td>354</td>
  </tr>
  <tr>
    <td>5869207</td>
    <td>My first CSS</td>
    <td>250</td>
  </tr>
</table>
</body>
</html>
```

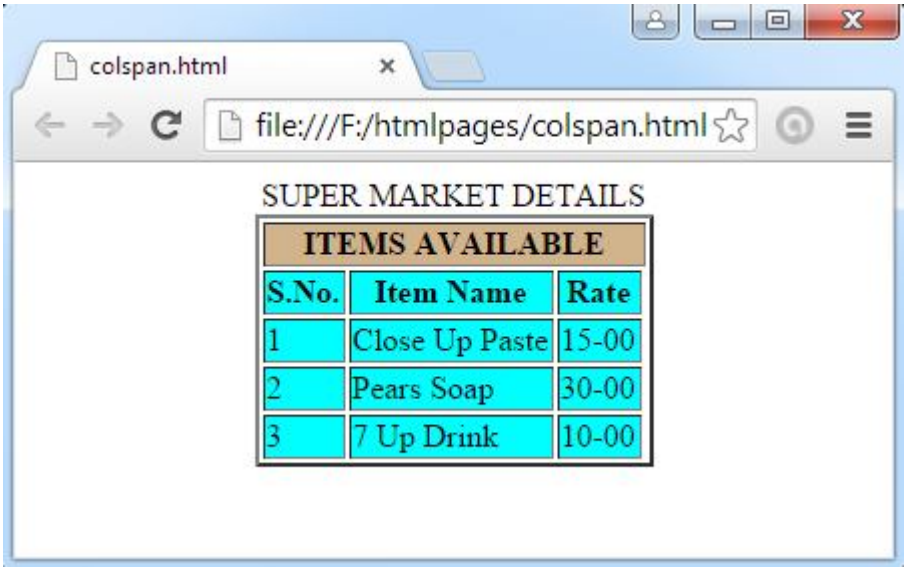


Spanning of cells is possible that is you can merge some sequence of rows or columns with the help of ROWSPAN or COLSPAN attributes respectively.
The colspan attribute defines the number of columns a cell should span.

Example: HTML Program to illustrate COLSPAN

```
<html>
     <body>
          <center>
               <table border="2">
               <Caption>SUPER MARKET DETAILS</caption>
               <colgroup>
                    <col span=4 align="center" bgcolor="cyan">
               </colgroup>
               <tr>
                    <th colspan=4 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>
```

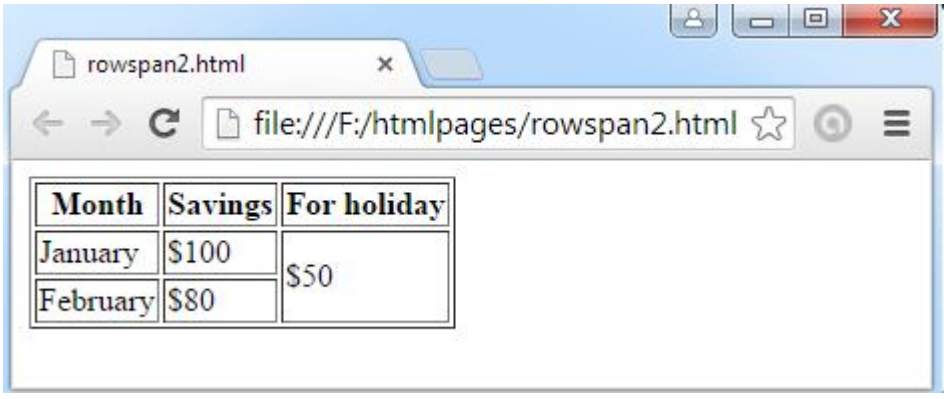
```
                <tr><td>3<td>7 Up Drink<td>10-00</tr>
                </table>
            </center>
        </body>
</html>
```



The rowspan attribute specifies the number of rows a cell should span.

Example: HTML Program to illustrate ROWSPAN

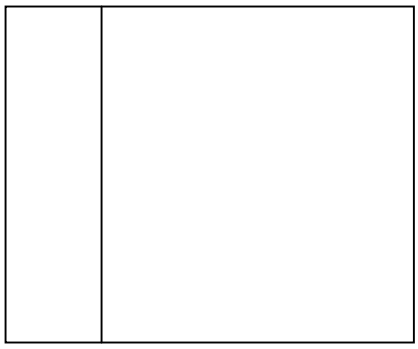
```
<html>
     <body>
          <table border="1">
               <tr><th>Month</th><th>Savings</th><th>For
holiday</th></tr>
               <tr><td>January</td><td>$100</td>
                    <td rowspan="2">$50</td></tr>
               <tr><td>February</td><td>$80</td></tr>
          </table>
     </body>
</html>
```



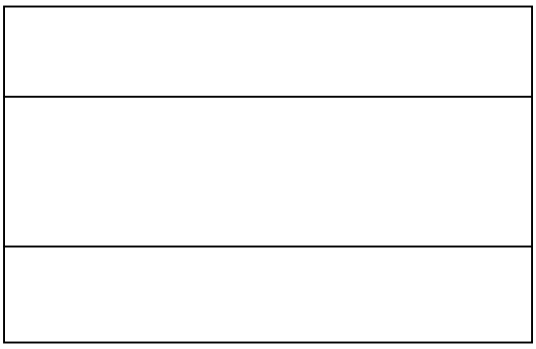
Framesets

Simply frame set 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 web page 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="sec" 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:

Sometimes 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 frame set.

Ex: In the above case, there are three frames with two frame sets. First total web page divided into two row-wise frames; first frame contains home page log. Second frame is again divided into two column-wise frames, in which first contains names that supported companies, other one displays history of company.

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.

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

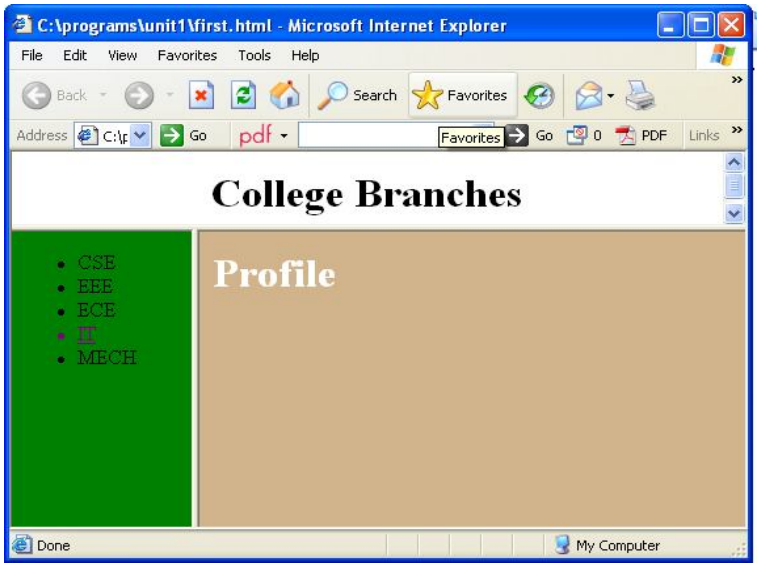
```
<html>
    <body>
        <center><h1>College Branches</h1></center>
    </body>
</html>
```

frame2.html

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

frame3.html

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



The above program creates the frames which can be resized but if we want to lock these frames at their respective sizes we can perform this by noresize parameter. Modify the above program as follows:

```
<frameset rows="20%,*">
    <frame name="fr1" src="frame1.html" noresize>
    <frameset cols="25%,*">
```

```
            <frame name="fr2" src="frame2.html" noresize>
            <frame name="fr3" src="frame3.html" noresize>
        </frameset>
    </frameset>
```

If we don't want to have the borders to the frames then we can do this by following

```
<frameset cols="150,*" frameborder="0" framespacing="0">
```

If we don't want to have the scrolling to the frames then we cando this by following

```
<frame name="xyz" src="xyz1.html" scrolling="No">
```

<iframe> tag

The <iframe> tag specifies an inline frame. An inline frame is used to embed another document within the current HTML document. <iframe> …</iframe>

Attributes:

align = left | right | top | middle | bottom

frameborder = 1 | 0

width = pixels

height = pixels

marginheight = pixels

Specifies the top and bottom margins of the content of an <iframe>

marginwidth = pixels

Specifies the left and right margins of the content of an <iframe>

name = text

Specifies the name of an <iframe>

scrolling = yes|no|auto

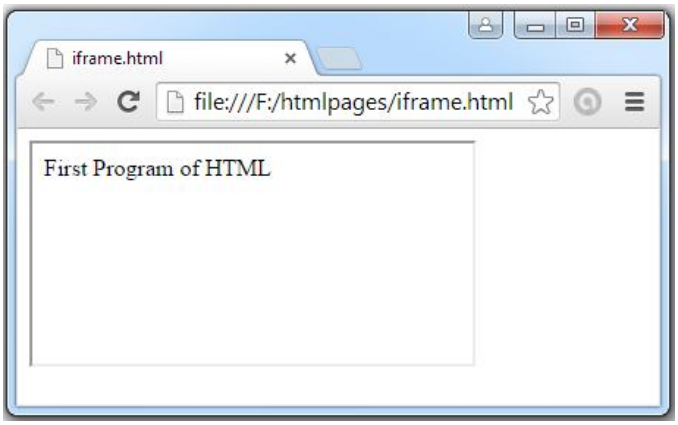
Specifies whether or not to display scrollbars in an <iframe>

seamless = seamless

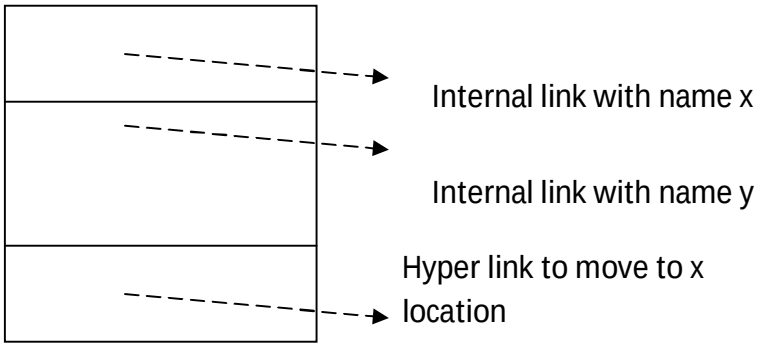
Specifies <iframe> should look like it is a part of the containing document

src = URL

Specifies the address of the document to embed in the <iframe>



Internal Linking: Linking lets you assign a location name to any other page or within the same page. If it happens with in the same page then it is called *internal linking*. You can create individual points on a web page and those individual points on a web page can be connected together with the help of anchor tag, and this total process is known as internal linking.



Using **Anchor tag** with the help of Name parameter creates a location on a page. Name is used to create individual link on the web page.

You can also create internal links within a page - for example a table of contents at the top with links to each chapter below. All you need to use is a very useful attribute called id (identification) and the symbol "#".

Use the id attribute to mark the element to which you want to link. For example:

```
<h1 id="heading1">heading 1</h1>
```

You can now create a link to that element by using "#" in the link attribute. The "#" must be followed by the id of the tag you want to link to. For example:

```
<a href="#heading1">Link to heading 1</a>
```

All will become clear with an example:

```
<html>
       <body>
          <p><a href="#heading1">Link to heading 1</a></p>
          <p><a href="#heading2">Link to heading 2</a></p>
     <br/><br/><br/><br/><br/><br/><br/><br/><br/><br/>
     <br/><br/><br/><br/><br/><br/><br/><br/><br/><br/>
     <br/><br/><br/><br/><br/><br/>
          <h1 id="heading1">heading 1</h1>
          <p>Text text text text</p>
          <h1 id="heading2">heading 2</h1>
          <p>Text text text text</p>
       </body>
</html>
```

HTML FORMS

HTML form is used to create user interface screens.

The <form> Element

HTML forms are used to collect user input. The <form> element defines an HTML form:

<form>
    form elements
</form>

<form> tag has several attributes and important among them is METHOD and ACTION.

- **METHOD Attribute:** It specifies how the data entered in form is sent to the destination, which may be web server. There are two methods of sending of information; one is POST, which sends the information along with the body of the HTML page, other on GET, sends the data along with web page address that is appended to the URL. There are several differences between POST method and GET method.
- **ACTION Attribute:** It indicates path to this script; when the user submits the form where it should go is given in ACTION parameter.

Post Method Vs Get Method

S.No.	POST method	GET method
1.	Information sent along with body	Information sent along with URL
2.	Data is not visible while sending	Data visible is site address
3.	Provides security	No security, because password also sent like this
4.	Can sent any number of characters	Only limited number of characters
5.	Special characters are also possible	Special characters are not possible to sent, only for standard characters

HTML forms contain form elements. Form elements are different types of input elements, checkboxes, radio buttons, submit buttons, and more.

User Interface Elements:

Label: HTML is not provide any separate tag for creating Label, just simple you can type the data which preceded by user interface element like text field that is taken label.

Text Field: You can create text fields in HTML by using <INPUT> tag. This INPUT element is powerful HTML tag, which is used for creating several window tools. In order to create a text field you can give <INPUT type=text>.

- Along with if you want to specify size simply can add `size` parameter to it.
- To identify each text field separately, use `name` parameter for every Input tag.
- You can also set a maximum number of characters that the text input will accept by inserting the `maxlength`=”length” attribute.
- `readonly` parameter indicates the text field information not modified
- `value` indicates the default value of the text field

Eg: Employee number : `<input type="text" size=20 name="eno">`

Password Field: When the user typed the information in a text field it is displayed as it is. But if you want display encoded characters like * for every type character, you should go for Password Fields. Creation of Password field is very simple, change value of Type parameter to "password", it creates password text field instead of simple text field.

Eg: Enter Password: `<input type="password" size=10 name="pwd">`

Buttons: - There are three types' buttons that you can create in HTML. They are simple user defined buttons, submit buttons and reset buttons. To create simple button you can give <Input type="button">, to display text on it, use Value parameter.

Eg: `<input type=button value="OK">`

Submit button is special kind of button, when the user selects that button, form action should be invoked. Other button is **Reset button**, when it clicked, clears all entries the user entered into the form. It helps the user correct mistakes or simply starts over. To create these buttons just you specify "submit" or "reset" in type parameter of Input tag.

```
Eg:  <input type="submit" value= "CREATE ACCOUNT">
     <input type="reset" value= "CLEAR">
```

Radio Buttons: Using input type="radio" we can place radio button on the web page. This component allows us to make only one selection at a time.

The parameters are name, value, checked

```
Eg:  <input type="radio" name="sex" value="m">
     <input type="radio" name="sex" value="f" checked>
```

Checkboxes: Used when multiple values are required simultaneously. It is the simplest component which is used particularly when we want to make some selection from several options. The parameters are name, value, checked, size

```
<input type="checkbox" name="c1" value="CSE">
<input type="checkbox" name="c1" value="IT">
```

Combo boxes and List boxes*:* List box is displaying n elements at a time, and provides facilities for user to select more than one. Where as a combo box displays single element at a time, and by clicking on combo arrow, displays other options available in it. It differs from List Box as it is not provide facility for selecting more than one item.

The procedure for creating list box and combo box is same. By using <Select> tag you can create this list of items instead of Input tag. To provide option in these, you should you <Option> tag. If you give Size parameter along with <select> tag it creates a list box, otherwise it creates combo box.

Ex: Courses

```
<select name="branch">
     <Option value="CSE">CSE</option>
     <Option value="IT">IT</option>
     <Option value="ECE">ECE</option>
     <Option value="EEE">EEE</option>
     <Option value="MECH">MECH</option>
</select>
```

Ex: Languages Known
```
<select size=3 multiple>
    <option value="English">English</option>
    <option value="Telugu">Telugu</option>
    <option value="Hindi">Hindi</option>
    <option value="Tamil">Tamil</option>
    <option value="Oriya">Oriya</option>
</select>
```

File: used to include a file with a form in a web page.
```
<input type=file name="f1">
```

TextArea: It is used to input multiple-line information from the end user like Address field.
```
<textarea name="t1" cols="25" rows="4"></textarea>
```

Example: Create a web page for displaying Student Admission Form. Display SNo, Sname, and Password, sex (radio, male or female), require Hostel Facility (Checkbox), Address (textarea), and Course Name (combo box), submit and reset buttons.

```
<html>
<head>
    <title> Student Admissions </title>
</head>
<body bgcolor="tan">
<form>
    <center>
    <h3>Admission Form</h3>
    <table border="1">
        <tr><td>sno<td><input type="text" size=5></tr>
        <tr><td>sname<td><input type="text" size=25></tr>
        <tr><td>password<td><input type="password"></tr>
        <tr>
            <td>Gender
            <td>
            <input type="radio" name="m1" value="m" checked>male
                <input type="radio" name="m1" value="f">female
        </tr>
        <tr><td>...
        <td><input type="checkbox">Required Hostel Facility</tr>
        <tr><td>address
            <td><textarea rows=3 cols=20></textarea></tr>
        <tr>
            <td>course
            <td><select>
                <option>MTech
                <option>MCA
                <option>Btech
                <option>BCA
                <option>Others
            </select></tr>
    </table>
    <input type="submit" value="submit">
```

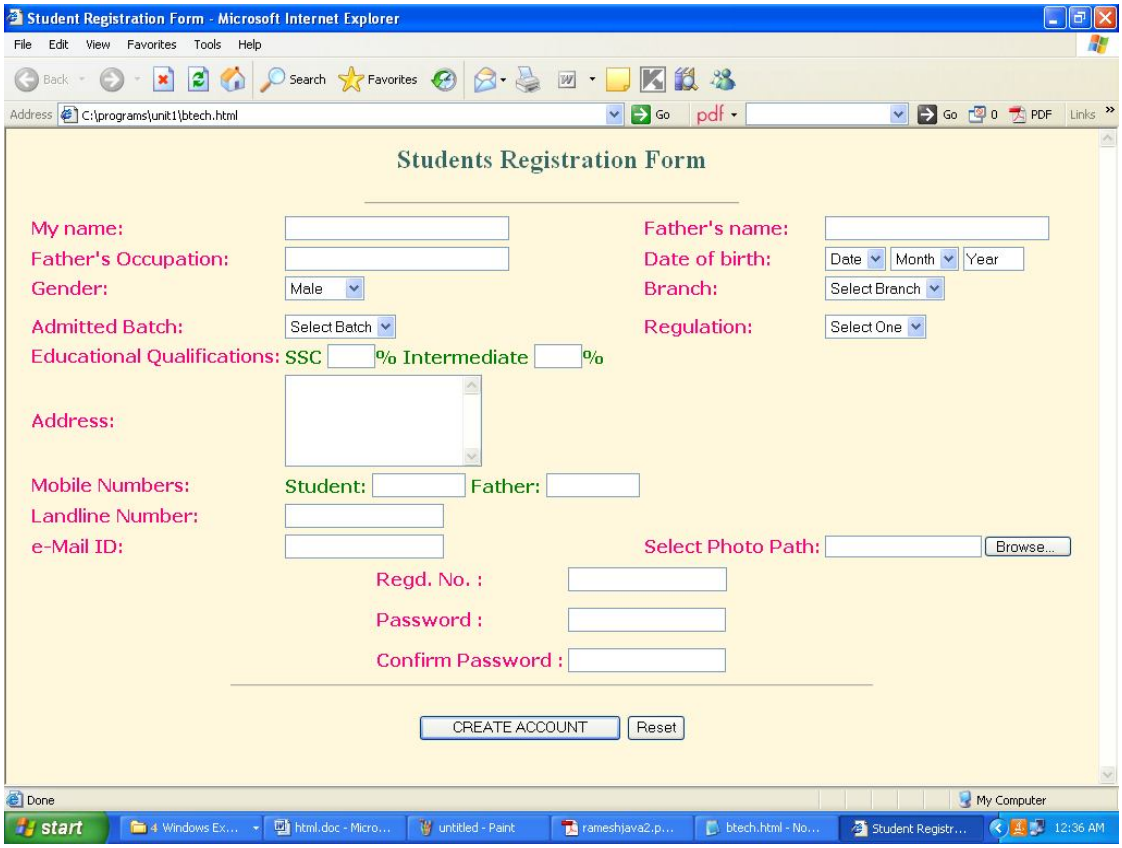
```
      <input type="reset">
      </center>
</form>
</body>
</html>
```

Example: Create a web page for displaying Student Registration Form.

```
<html>
<head><title>Student Registration Form </title></head>
<body bgcolor = "#FFF8DC"  text = "336666">
<center><h2>Students Registration Form</h2></font></center><hr width =
"35%"></center>
<center>
<table border = 0>
<tbody>
<tr align = left><th><font face="Verdana" size="3" color="#FF0080">My
name: <td><INPUT TYPE="text" name = "name" size = "30">
<th><font face="Verdana" size="3" color="#FF0080">Father's name:
<td><INPUT TYPE="text" name = "fname" size = "30"></tr>
<tr align = left><th><font face="Verdana" size="3"
color="#FF0080">Father's Occupation:<td><INPUT TYPE="text" name =
"occp" size = "30">
<th><font face="Verdana" size="3" color="#FF0080">Date of birth:<td>
<SELECT NAME="dd"
SINGLE><OPTION>Date<OPTION>01<OPTION>02<OPTION>03<OPTION>04
```

```
<OPTION>05<OPTION>06<OPTION>07<OPTION>08<OPTION>09<OPTION>10<OPTION>11
<OPTION>12<OPTION>13<OPTION>14<OPTION>15<OPTION>16<OPTION>17<OPTION>18
<OPTION>19<OPTION>20<OPTION>21<OPTION>22<OPTION>23<OPTION>24<OPTION>25
<OPTION>26<OPTION>27<OPTION>28<OPTION>29<OPTION>30<OPTION>31</select>
<SELECT NAME="mm"
SINGLE><OPTION>Month<OPTION>Jan<OPTION>Feb<OPTION>Mar
<OPTION>Apr<OPTION>May<OPTION>Jun<OPTION>Jul<OPTION>Aug<OPTION>Sep<OPT
ION>Oct<OPTION>Nov<OPTION>Dec</select>
<INPUT TYPE="text" name = "yy" size = "5" value = "Year"
maxlength="4"></tr>
<tr align = left><th><font face="Verdana" size="3"
color="#FF0080">Gender: <td><SELECT NAME="gend" SINGLE>
<OPTION>Male <OPTION>Female</select>
<th><font face="Verdana" size="3" color="#FF0080">Branch: <td><SELECT
NAME="branch" SINGLE><OPTION>Select Branch<OPTION>CSE-A<OPTION>CSE-
B<OPTION>IT<OPTION>ECE<OPTION>EEE<OPTION>MECH</SELECT>
<tr><th><td></tr><tr><th><td></tr><tr><th><td></tr>
<tr align = left><th><font face="Verdana" size="3"
color="#FF0080">Admitted Batch:
<td><SELECT NAME="adm" SINGLE><OPTION>Select Batch<OPTION>2005-2009
<OPTION>2006-2010<OPTION>2007-2011<OPTION>2008-2012<OPTION>2009-
2013<OPTION>2010-2014<OPTION>2011-2015<OPTION>2012-2016<OPTION>2013-
2017
</SELECT>
<th><font face="Verdana" size="3" color="#FF0080">Regulation:
<td><SELECT NAME="regu" SINGLE><OPTION>Select
One<OPTION>R05<OPTION>R07<OPTION>R08</SELECT>
<tr align = left><th><font face="Verdana" size="3"
color="#FF0080">Educational
Qualifications: <td><font face="Verdana" size="3" color="green"><b>SSC
<INPUT TYPE="text" SIZE="3" NAME="ssc" maxlength="5">%    Intermediate
<INPUT TYPE="text" SIZE="3" NAME="inter" maxlength="5">%</b></tr>
<tr align = left><th><font face="Verdana" size="3"
color="#FF0080">Address: <td><textarea  rows = "5" name =
"add"></textarea></tr>
<tr align = left><th><font face="Verdana" size="3"
color="#FF0080">Mobile Numbers:<td><font face="Verdana" size="3"
color="green"><b>Student: <INPUT TYPE="text" size = "10" NAME="stu"
maxlength="10">  Father: </b><INPUT TYPE="text" size = "10"
NAME="fath" maxlength="10"></tr>
<tr align = left><th><font face="Verdana" size="3"
color="#FF0080">Landline
Number:<td><INPUT TYPE="text"  size = "20" NAME="ll"
maxlength="12"></tr>
<tr align = left><th><font face="Verdana" size="3" color="#FF0080">e-
Mail ID: <td><INPUT TYPE="text" NAME="eid" size = "20">
<th><font face="Verdana" size="3" color="#FF0080">Select Photo Path:
<td><INPUT TYPE="file" NAME="pho" size = "20">
</tr>
</font>
</tbody>
</table>
```

```
</center>
<center>
<table border = 0>
<tbody>
<tr align = left><th><font face="Verdana" size="3" color="#FF0080">
Regd. No. : <td><INPUT TYPE="text" SIZE="20" NAME="user"
maxlength="10"> </tr>
<tr><th><td></tr><tr><th><td></tr><tr><th><td></tr>
<tr align = left><th><font face="Verdana" size="3" color="#FF0080">
Password : <td><INPUT TYPE="password" SIZE="20" NAME="pass">  </tr>
<tr><th><td></tr><tr><th><td></tr><tr><th><td></tr>
<tr align = left><th><font face="Verdana" size="3" color="#FF0080">
Confirm Password :<td><INPUT TYPE="password" SIZE="20" NAME="cpass">
</tr>
</font>
</tbody>
</table>
</center>
<hr width = "60%">
<h3 align = "center"><INPUT TYPE="submit" value= "CREATE ACCOUNT">
<INPUT TYPE="RESET" NAME="RESET"></h3>
</body>
</html>
```



HTML 5

Tag	Info	V	Attributes*
<!-- -->	comment	4 / 5	none
<!DOCTYPE>	document type	4 / 5	none
<a>	hyperlink	4 / 5	href | hreflang | media | ping | rel | target | type
<abbr>	abbreviation	4 / 5	*global attributes***
<acronym>	acronym	4	-
<address>	address element	4 / 5	*global attributes***
<applet>	applet	4	-
<area>	area inside an image map	4 / 5	alt | coords | href | hreflang | media | ping | rel | shape | target | type
<article>	article	5	*global attributes***
<aside>	outside the main flow of the narrative	5	*global attributes***
<audio>	sound content	5	autobuffer | autoplay | controls | loop | src
<b>	bold text	4 / 5	*global attributes***
<base>	base URL for all the page links	4 / 5	href | target
<basefont>	Base font for the document	4	-
<bb>	invoked user agent com-	5	type
<bdo>	direction of text display	4 / 5	dir
<big>	big text	4	-
<blockquote>	long quotation	4 / 5	cite
<body>	body element	4 / 5	*global attributes***
 	inserts a single line break	4 / 5	*global attributes***
<button>	push button	4 / 5	autofocus | disabled | form | formaction | formenctype | formmethod | formnovalidate | formtarget | name | type | value
<canvas>	Graphic area	5	height | width
<caption>	table caption	4 / 5	*global attributes***
	centered text	4	-
<cite>	citation	4 / 5	*global attributes***
<code>	computer code text	4 / 5	*global attributes***
<col>	attributes for table columns	4 / 5	span
<colgroup>	groups of table columns	4 / 5	span
<command>	command button	5	checked | default | disabled | hidden | icon | label | radiogroup | type
<datagrid>	data in a tree, list or tabular	5	disabled
<datalist>	dropdown list	5	*global attributes***
<dd>	definition description	4 / 5	*global attributes***
<del>	deleted text	4 / 5	cite | datetime
<details>	details of an element	5	open
<dialog>	dialog (conversation)	5	*global attributes***
<dir>	directory list	4	-
	section in a document	4 / 5	*global attributes***
<dfn>	definition term	4 / 5	title
<dl>	definition list	4 / 5	*global attributes***
<dt>	definition term	4 / 5	*global attributes***
	emphasized text	4 / 5	*global attributes***

Tag	Info	V	Attributes*
<embed>	external interactive content or plugin	5	height | src | type | width
<fieldset>	fieldset	4 / 5	disabled | form | name
<figure>	group of media content, and their caption	5	*global attributes***
<font>	text font, size, and color	4	-
<footer>	footer for a section or page	5	*global attributes***
<form>	form	4 / 5	action | data | replace | accept | accept-charset | enctype | method | target
<frame>	sub window	4	-
<frameset>	set of frames	4	-
<h1> to <h6>	header 1 to header 6	4 / 5	*global attributes***
<head>	information about the document	4 / 5	none
<header>	header for a section or page	5	*global attributes***
<hgroup>	heading section	5	*global attributes***
<hr>	horizontal rule	4 / 5	*global attributes***
<html>	html document	4 / 5	manifest
<i>	italic text	4 / 5	*global attributes***
<iframe>	inline sub window (frame)	4 / 5	src | name | sandbox | seamless | width | height
<img>	image	4 / 5	alt | src | height | ismap | usemap | width
<input>	input field	4 / 5	accept | alt | autocomplete | autofocus | checked | disabled | form | formaction | formenctype | formmethod | formnovalidate | formtarget | height | list | max | maxlength | min | multiple | name | pattern | placeholder | readonly | required | size | src | step | type | value | width
<ins>	inserted text	4 / 5	cite | datetime
<isindex>	single-line input field	4	-
<kbd>	keyboard text	4 / 5	*global attributes***
<label>	label for a form control	4 / 5	for
<legend>	fieldset title	4 / 5	*global attributes***
	list item	4 / 5	value
<link>	resource reference	4 / 5	href | rel | media | hreflang | type | sizes
<mark>	marked text	5	*global attributes***
<map>	image map	4 / 5	id
<menu>	menu list	4 / 5	label | type
<meta>	meta information	4 / 5	charset | content | http-equiv | name
<meter>	measurement within a predefined range	5	high | low | max | min | optimum | value
<nav>	navigation links	5	*global attributes***
<noframes>	noframe section	4	-
<noscript>	noscript section	4 / 5	none
<object>	embedded object	4 / 5	data | height | type | usemap | width | object

Tag	Info	V	Attributes*
	ordered list	4 / 5	start | reversed
<optgroup>	option group	4 / 5	disabled | label
<option>	option in a drop-down list	4 / 5	disabled | label | selected | value
<output>	some types of output	5	form
<p>	paragraph	4 / 5	*global attributes***
<param>	parameter for an object	4 / 5	name | value
<pre>	preformatted text	4 / 5	*global attributes***
<progress>	progress of a task of any kind	5	max | value
<q>	short quotation	4 / 5	cite
<ruby>	ruby annotations	5	*global attributes***
<rp>	provide parentheses around a ruby text	5	*global attributes***
<rt>	ruby text component	5	*global attributes***
<s>	strikethrough text	4	-
<samp>	sample computer code	4 / 5	*global attributes***
<script>	script	4 / 5	async | type | defer | src | charset
<section>	section	5	cite
<select>	selectable list	4 / 5	autofocus | data | disabled | form | multiple | name
<small>	small text	4 / 5	*global attributes***
<source>	media resources	5	media | src | type
	inline section	4 / 5	*global attributes***
<strike>	strikethrough text	4	-
	strong text	4 / 5	*global attributes***
<style>	style definition	4 / 5	media | type | scoped
<sub>	subscripted text	4 / 5	*global attributes***
<sup>	superscripted text	4 / 5	*global attributes***
<table>	table	4 / 5	*global attributes***
<tbody>	table body	4 / 5	*global attributes***
<td>	table cell	4 / 5	colspan | rowspan | headers
<textarea>	text area	4 / 5	autofocus | cols | disabled | form | name | readonly | required | rows | maxlength | placeholder | wrap
<tfoot>	table footer	4 / 5	*global attributes***
<th>	table header	4 / 5	colspan | rowspan | scope
<thead>	table header	4 / 5	*global attributes***
<time>	date/time	5	datetime
<title>	document title	4 / 5	none
<tr>	table row	4 / 5	*global attributes***
<tt>	teletype text	4	-
<u>	underlined text	4	-
	unordered list	4 / 5	*global attributes***
<var>	variable	4 / 5	*global attributes***
<video>	video	5	src | poster | autobuffer | autoplay | loop | controls | width | height
<xmp>	preformatted text	4	-

* **Attributes:** Lists attributes specific to that tag. Deprecated (html4 only) attributes are not listed

** **Global Attributes:** class | contenteditable | contextmenu | dir | draggable | id | irrelevant | lang | ref | registrationmark | tabindex | template | title

V = Which version of HTML is this tag valid for

HTML 5 - extended

Tag	Info	Attributes
<!-- -->	**comment**: comments are displayed in code only. Tag contents are not rendered in the browser	none
<!DOCTYPE>	**document type**: defines which specification the document follows	none
<a>	**anchor**: used to provide a link to another web resource	**href**: destination resource of the hyperlink **hreflang**: gives the language of the linked resource **media**: describes for which media the target document was designed **ping**: gives the URLs of the resources that are interested in being notified if the user follows the hyperlink **rel**: relationship between the document containing the hyperlink and the destination resource [*alternate* | *archives* | *author* | *bookmark* | *contact* | *external* | *feed* | *first* | *help* | *icon* | *index* | *last* | *license* | *next* | *nofollow* | *noreferrer* | *pingback* | *prefetch* | *prev* | *search* | *stylesheet* | *sidebar* | *tag* | *up*] **target**: gives the name of the browsing context that will be used [*_blank* | *_parent* | *_self* | *_top*] **type**: gives the MIME type of the linked resource
<abbr>	**abbreviation**: an abbreviation or acronym, optionally with its expansion	*global attributes***
<address>	**address element**: represents the contact information for its nearest article or body element ancestor	*global attributes***
<area>	**area**: either a hyperlink with some text and a corresponding area on an image map, or a dead area on an image map	**alt**: alternate text for the area **cords**: coordinates for the clickable area **href**: destination resource of the hyperlink **hreflang**: gives the language of the linked resource **media**: describes for which media the target document was designed **ping**: gives the URLs of the resources that are interested in being notified if the user follows the hyperlink **rel**: relationship between the document containing the hyperlink and the destination resource [*alternate* | *archives* | *author* | *bookmark* | *contact* | *external* | *feed* | *first* | *help* | *icon* | *index* | *last* | *license* | *next* | *nofollow* | *noreferrer* | *pingback* | *prefetch* | *prev* | *search* | *stylesheet* | *sidebar* | *tag* | *up*] **shape**: defines the shape of the area [*default* | *rect* | *rectangle* | *circ* | *circle* | *poly* | *polygon*] **target**: gives the name of the browsing context that will be used [*_blank* | *_parent* | *_self* | *_top*] **type**: gives the MIME type of the linked resource
<article>	**article element**: a section of a page that consists of a composition that forms an independent part of a document, page, or site	*global attributes***
<aside>	**aside element**: a section of a page that consists of content that is tangentially related to the content around the aside element, and which could be considered separate from that content	*global attributes***
<audio>	**sound content**: represents a sound or audio stream	**autobuffer**: determines if the audio will be buffered [*autobuffer*] **autoplay**: determine if the audio will automatically play [*autoplay*] **controls**: indicates that the author has not provided a scripted controller and would like the user agent to provide its own set of controls [*controls*] **loop**: sets whether the audio will start once the end is reached [*loop*] **src**: URL of the audio to play
<b>	**bold text**: creates text that will be made bold	*global attributes***
<base>	**base element**: base URL for all the page links	**href**: URL to use as the base URL for links in the page **target**: sets the base target for links in the page [*_blank* | *_parent* | *_self* | *_top*]
<bb>	**browser button**: a user agent command that the user can invoke	**type**: indicates the kind of command [*makeapp*]
<bdo>	**bdo element**: represents explicit text directionality formatting control for its children	**dir**: direction override [*ltr* | *rtl*]
<blockquote>	**block quote element**: a section that is quoted from another source	**cite**: URL of the origin of the quote
<body>	**body element**: main content of the document	*global attributes***
 	break: inserts a single line break	*global attributes***

Tag	Info	Attributes
<button>	**button**: a button page element	**autofocus**: indicate that a control is to be focused as soon as the page is loaded [*autofocus*] **disabled**: prevents the button from being pressed [*disabled*] **form**: used to explicitly associate the button element with its form owner **formaction**: URL that specifies a form processing agent **formenctype**: specifies the content type used to submit the form to the server [*application/x-www-form-urlencoded* | *multipart/form-data* | *text/plain*] **formmethod**: which HTTP method will be used to submit the forms data [*get* | *post* | *put* | *delete*] **formnovalidate**: indicate whether the form is to be validated during submission [*formnovalidate*] **formtarget**: gives the target when the form is submitted [*_blank* | *_parent* | *_self* | *_top*] **name**: elements name **type**: controls the behavior of the button when it is activated [*submit* | *reset* | *button*] **value**: gives the element's value for the purposes of form submission
<canvas>	**canvas element**: a resolution-dependent bitmap canvas, which can be used for rendering graphs, game graphics, or other visual images on the fly	**height**: height of the canvas in pixels - default is 150 **width**: width of the canvas in pixels - default is 300
<caption>	**table caption**: the title of the table that is its parent, if it has a parent and that is a table element.	*global attributes***
<cite>	**citation**: represents the title of a work	*global attributes***
<code>	**computer code text**: represents a fragment of computer code. This could be an XML element name, a filename, a computer program, or any other string that a computer would recognize.	*global attributes***
<col>	**column**: defines the attribute values for one or more columns in a table. Used inside of a table or colgroup	**span**: number of columns the tag should span
<colgroup>	**column group**: a group of one or more columns in the table that is its parent, if it has a parent and that is a table element	**span**: number of columns the tag should span
<command>	**command button**: a command that the user can invoke (like radio button or checkbox)	**type**: Specifies the type of command [*checkbox* | *command* | *radio*] **label**: gives the name of the command, as shown to the user **icon**: a URL to a picture that represents the command **disabled**: prevents the command from being executed [*disabled*] **checked**: Determines if the command is checked by default [*checked*] **radiogroup**: gives the name of the group of commands that will be toggled when the command itself is toggled **title**: gives a hint describing the command, which might be shown to the user to help them
<datagrid>	**datagrid element**: an interactive representation of tree, list,	**disabled**: defines whether the list is selectable [*disabled*]
<datalist>	**dropdown list**: a set of option elements that represent predefined options for other controls	*global attributes***
<dd>	**definition description**: description, definition, or value, part of a term-description group in a description list (dl element), and the discourse, or quote, part in a conversation (dialog element)	*global attributes***
<del>	**deleted text**: represents a removal from the document	**cite**: a URL used to specify the address of a document that explains the change **datetime**: used to specify the time and date of the change
<details>	**details element**: represents additional information or controls which the user can obtain on demand	**open**: indicates whether the details are to be shown to the user [*open*]
<dialog>	**dialog element**: represents a conversation, meeting minutes, a chat transcript, a dialog in a screenplay, an instant message log, or some other construct in which different players take turns	*global attributes***
	document block: creates a block level element with no special meaning	*global attributes***

attributes - values in [] are the accepted values

HTML 5 - extended

Tag	Info	Attributes
<dfn>	**definition term**: the defining instance of a term	**title**: the exact value of the term being defined
<dl>	**definition list**: an association list consisting of zero or more name-value groups (a description list). Each group must consist of one or more names (dt elements) followed by one or more values (dd elements)	*global attributes***
<dt>	**definition term**: the term, or name, part of a term-description group in a description list (dl element), and the talker, or speaker, part of a talker-discourse pair in a conversation (dialog element)	*global attributes***
****	**emphasized text**: represents stress emphasis of its contents.	*global attributes***
<embed>	**embed element**: an external (typically non-HTML) application or interactive content	**src**: URL of the resource being embedded **type**: gives the MIME type of the plugin to instantiate **height**: height of the embedded content in pixels **width**: width of the embedded content in pixels
<fieldset>	**fieldset element**: a set of form controls grouped under a common name	**disabled**: controls whether all the form control descendants are disabled [*disabled*] **form**: used to explicitly associate the fieldset element with its form owner **name**: gives the name of the form control
<figure>	**figure element**: some flow content, optionally with a caption, that is self-contained and is typically referenced as a single unit from the main flow of the document	*global attributes***
<footer>	**footer element**: represents a footer for the section it applies to	*global attributes***
<form>	**form element**: represents a collection of form-associated elements, some of which can represent editable values that can be submitted to a server for processing	**accept-charset**: gives the character encodings that are to be used for the submission **action**: URL that specifies a form processing agent **autocomplete**: determines if form elements will have their autocomplete turned on or off by default [*on* | *off*] **enctype**: specifies the content type used to submit the form to the server [*application/x-www-form-urlencoded* | *multipart/form-data* | *text/plain*] **method**: which HTTP method will be used to submit the forms data [*get* | *post* | *put* | *delete*] **name**: elements name **novalidate**: indicate whether the form is to be validated during submission [*novalidate*] **target**: gives the target when the form is submitted [*_blank* | *_parent* | *_self* | *_top*]
<h1> to <h6>	**headers (1-6)**: represent headings for their sections. elements have a rank given by the number in their name	*global attributes***
<head>	**head element**: contains information about the document	none
<header>	**header element**: represents a group of introductory or navigational aids	*global attributes***
<hgroup>	**heading group**: used to group a set of h1–h6 elements when the heading has multiple levels, such as subheadings, alternative titles, or taglines	*global attributes***
<hr>	**horizontal rule**: creates a horizontal rule (line)	*global attributes***
<html>	**html document**: root of an HTML document.	**manifest**: a URL to the address of the document's application cache manifest
<i>	**italic text**: indicates the text is to be rendered with emphasis	*global attributes***
<iframe>	**inline frame**: represents a nested browsing window	**src**: URL of a page that the nested browsing context is to contain **name**: elements name **sandbox**: enables a set of extra restrictions on any content hosted by the iframe [*allow-same-origin* | *allow-forms* | *allow-scripts*] **seamless**: indicates whether the iframe element's browsing context is to be rendered in a manner that makes it appear to be part of the containing document [*seamless*] **height**: height of the frame in pixels **width**: width of the frame in pixels
<img>	**image**: represents an image	**alt**: text to display if the image can not **src**: a URL to the image file **usemap**: name of the map to use for the image **ismap**: provides access to a server-side image map **height**: height of the image in pixels **width**: width of the image in pixels

Tag	Info	Attributes
<input>	**input field**: a typed data field, usually with a form control to allow the user to edit the data	***Attributes are dependant upon input type*** **accept**: specified to provide user agents with a hint of what file types the server will be able to accept **alt**: provides the textual label for the alternative button for users and user agents who cannot use the image **autocomplete**: determines if the data is considered sensitive and if autocomplete will be used [*on* | *off* | *default*] **autofocus**: determines if the input will get focus when a page loads [*autofocus*] **checked**: determines if the input will be checked by default [*checked*] **disabled**: prevents the input from being pressed [*disabled*] **form**: used to explicitly associate the button element with its form owner **formaction**: URL that specifies a form processing agent **formenctype**: specifies the content type used to submit the form to the server [*application/x-www-form-urlencoded* | *multipart/form-data* | *text/plain*] **formmethod**: which HTTP method will be used to submit the forms data [*get* | *post* | *put* | *delete*] **formnovalidate**: indicate whether the form is to be validated during submission [*formnovalidate*] **formtarget**: gives the target when the form is submitted [*_blank* | *_parent* | *_self* | *_top*] **height**: height of the input in pixels **list**: used to identify an element that lists predefined options suggested to the user **max** and **max**: indicate the allowed range of values for the element **maxlength**: controls the maxlength of the input to a control **multiple**: indicates whether the user is to be allowed to specify more than one value [*multiple*] **name**: elements name **pattern**: specifies a regular expression against which the control's value is to be checked **placeholder**: a short hint intended to aid the user with data entry **readonly**: determines if the control is readonly [*readonly*] **required**: determines if the input is required before the form submits [*required*] **size**: gives the number of characters that, in a visual rendering, the user agent is to allow the user to see while editing **src**: URL to an image (image button) **step**: indicates the granularity that is expected (and required) of the value **type**: controls the data type (and associated control) of the element [*hidden* | *text* | *search* | *tel* | *url* | *email* | *password* | *datetime* | *date* | *month* | *week* | *time* | *datetime-local* | *number* | *range* | *color* | *checkbox* | *radio* | *file* | *submit* | *image* | *reset* | *button*] **value**: sets the element's value **width**: width of the input in pixels
<ins>	**inserted text**: an addition to the document	**cite**: a URL used to specify the address of a document that explains the change **datetime**: used to specify the time and date of the change
<kbd>	**keyboard text**: user input (typically keyboard input, although it may also be used to represent other input, such as voice commands)	*global attributes***
<label>	**label**: caption in a user interface	**for**: specified to indicate a form control with which the caption is to be associated
<legend>	**fieldset title**: sets the title of a fieldset element	*global attributes***
** **	**list item**: represents a list item of an Ordered (OL) or Unordered list (UL)	**value**: used in an Ordered List (OL) to set the display value
<link>	**resource link**: allows authors to link their document to other resources	**href**: destination resource of the hyperlink **rel**: relationship between the document containing the hyperlink and the destination resource [*alternate* | *archives* | *author* | *bookmark* | *contact* | *external* | *feed* | *first* | *help* | *icon* | *index* | *last* | *license* | *next* | *nofollow* | *noreferrer* | *pingback* | *prefetch* | *prev* | *search* | *stylesheet* | *sidebar* | *tag* | *up*] **media**: describes for which media the target document was designed **hreflang**: gives the language of the linked resource **type**: gives the MIME type of the linked resource **sizes**: gives the sizes of icons for visual media.
<mark>	**marked text**: a run of text in one document marked or highlighted for reference purposes, due to its relevance in another context.	*global attributes***
<map>	**image map**: in conjunction with any area element descendants, defines an image map	**name**: gives the map a name so that it can be referenced
<menu>	**menu list**: a list of commands	**label**: sets a visible label for the menu **type**: indicates the kind of menu being declared [*context* | *toolbar* | *list*]

attributes - values in [] are the accepted values

HTML 5 - extended

Tag	Info	Attributes
<meta>	**meta information**: sets meta information for the page (like title, description)	**charset**: specifies the character encoding used by the document **content**: sets the value of the document metadata **http-equiv**: sets a pragma directive [*content-language* | *content-type* | *default-style* | *refresh*] **name**: set the name of the meta information
<meter>	**meter element**: scalar measurement within a known range, or a fractional value	**high**: specifies the range that is considered to be the "high" part **low**: specifies the range that is considered to be the "low" part **min:** specifies the lower boundary **max**: specifies the upper boundary **optimum**: specifies the range that is considered to be the "optimum" part **value**: current location within the range
<nav>	**navigation element**: section of a page that links to other pages or to parts within the page: a section with navigation links	*global attributes***
<noscript>	**noscript section**: represents nothing if scripting is enabled, and represents its children if scripting is disabled	*global attributes***
<object>	**embedded object**: an external resource, which, depending on the type of the resource, will either be treated as an image, as a nested browsing context, or as an external resource to be processed by a plugin	**data**: specifies the address of the resource **name**: valid browsing context name **usemap**: name of the map to use for the image **form**: form to associate the object with **type**: gives the MIME type of the plugin to instantiate **height**: height of the embedded content in pixels **width**: width of the embedded content in pixels
** **	**ordered list**: list of items, where the items have been intentionally ordered	**start**: the ordinal value of the first list item **reversed**: indicates that the list is a descending list [*reversed*]
<optgroup>	**option group:** a group of option elements with a common label	**disabled**: disables all options in the group [*disabled*] **label**: gives the name of the group, as shown to the user
<option>	option element: an option in a select element or as part of a list of suggestions in a datalist element	**disabled**: prevent any clicks on an option item [*disabled*] **label**: provides a label for element **selected**: determines if the option is selected by default [*selected*] **value**: provides a value for element
<output>	**output element**: the result of a calculation	**form**: used to explicitly associate the output element with its form owner **for**: allows an explicit relationship to be made between the result of a calculation and the elements that represent the values that went into the calculation or that influenced the calculation
<p>	**paragraph**: creates a paragraph	*global attributes***
<param>	**parameter element**: defines parameters for plugins invoked by object elements. It does not represent anything on its own	**name**: gives the name of the parameter. **value**: gives the value of the parameter.
<pre>	**preformatted text**: represents a block of preformatted text	*global attributes***
<progress>	**progress element**: represents the completion progress of a task.	**max**: specifies how much work the task requires in total **value**: specifies how much of the task has been completed
<q>	**short quotation**: phrasing content quoted from another source	**cite**: a URL of a page where the quote was taken from
<ruby>	**ruby annotations**: allows one or more spans of phrasing content to be marked with ruby annotations	*global attributes***
<rp>	**ruby text parentheses**: can be used to provide parentheses around a ruby text component of a ruby annotation	*global attributes***
<rt>	**ruby text component**: marks the ruby text component of a ruby annotation	*global attributes***
<samp>	**sample**: sample output from a program or computing system.	*global attributes***
<script>	**script element**: allows authors to include dynamic script and data blocks in their documents	**async**: the script will be executed asynchronously, as soon as it is available [*async*] **type**: gives the MIME type of the script or format of the data **defer**: the script is executed when the page has finished parsing [*defer*] **src**: gives the address of the external script resource to use **charset**: specifies the character encoding of the external script resource
<section>	**section element**: represents a generic document or application section	**cite**: a URL of a page where the section was taken from

Tag	Info	Attributes
<select>	**selectable list**: a control for selecting amongst a set of options	**autofocus**: determines if the controls gets focus when the page loads [*autofocus*] **disabled**: prevent the selection of an item [*disabled*] **form**: form to associate the select with **multiple**: allows the selection of multiple items [*multiple*] **size**: gives the number of options to show to the user
<small>	**small text**: small print or other side comments	*global attributes***
<source>	**source element**: allows authors to specify multiple media resources for media elements.	**media**: gives the intended media type of the media resource **src**: URL of the media resource **type**: gives the MIME type of the source
****	**span**: used for an inline element	*global attributes***
****	**strong**: represents strong importance for its contents	*global attributes***
<style>	**style definition**: allows authors to embed style information in their documents	**media**: says which media the styles apply to **type**: gives the MIME type (default: text/css) **scoped**: indicates that the styles are intended just for the subtree rooted at the style element's parent element [*scoped*]
<sub>	**subscript**: subscript text	*global attributes***
<sup>	**superscript**: superscript text	*global attributes***
<table>	**table element**: represents data with more than one dimension, in the form of a table	*global attributes***
<tbody>	**table body**: represents a block of rows that consist of a body of data for a table	*global attributes***
<td>	**table cell**: represents a data cell in a table	**colspan**: sets how many columns a cell will span **rowspan**: sets how many rows a cell will span **headers**: space separated list of ids corresponding to the th ids and give header information for the cell
<textarea>	**text area**: a multiline plain text edit control for the element's raw value	**autofocus**: determines if the textarea gets focus when the page loads [*autofocus*] **cols**: specifies the expected maximum number of characters per line **disabled**: prevents entry of text [*disabled*] **form**: form to associate the textarea with **readonly**: control whether the text can be edited by the user or not [*readonly*] **required**: will be required to enter a value before submitting the form [required] **rows**: specifies the number of lines to show **maxlength**: controls the maximum amount of characters which can be entered **placeholder**: a hint intended to aid the user with data entry **wrap**: defines how text is wrapped [*soft* | *hard*]
<tfoot>	**table footer**: the block of rows that consist of the column summaries (footers) for a table	*global attributes***
<th>	**table header**: represents a header cell in a table	**colspan**: determines how many columns a cell will span **rowspan**: determines how many rows a cell will span **headers**: space separated list of ids corresponding to the th ids and give header information for the cell scope: determines where the cell provides its header information [*col* | *colgroup* | *row* | *rowgroup*]
<thead>	**table header**: the block of rows that consist of the column labels (headers) for a table	*global attributes***
<time>	**date/time**: a precise date and/or a time in the Gregorian calendar	**datetime**: date/time using the Gregorian calendar
<title>	**title element**: sets the title of the document	none
<tr>	**table row**: a row of cells in a table	*global attributes***
** **	**unordered list**: a list of items, where the order of the items is not important	*global attributes***
<var>	**variable**: this could be an actual variable in a mathematical expression or programming context	*global attributes***
<video>	**video element**: a video or movie	**poster**: URL of an image file that the user agent can show while no video data is available **autobuffer**: determines if the audio will be buffered [*autobuffer*] **autoplay**: determine if the audio will automatically play [*autoplay*] **controls**: indicates that the author has not provided a scripted controller and would like the user agent to provide its own set of controls [*controls*] **loop**: sets whether the audio will start once the end is reached [*loop*] **src**: URL of the audio to play **width**: width of the video in pixels **height**: height of the video in pixels

attributes - values in [] are the accepted values