HYPER TEXT MARKUP LANGUAGE (HTML)

The language which is used to design the web pages is called as HTML. It is a documentation language to mark the headings, title, tables, frames, etc. It is a universal language to design a static web page. It is machine independent, and all Internet browsers accept the HTML code.

➔ History of HTML

The seed for HTML was sown by IBM in the early 1980s. They wanted to set a documentation system in which one could marks the title, headings, paragraphs and font type selections. They called it General Mark-up Language (GML). In 1986, the International Standardizing Organization (ISO) took up this concept and standardized it as Standard Generalized Mark-up Language (SGML). In 1989, Tim Berners Lee and his team in the European Laboratory for Particle Physics (CERR) designed the present form of the documentation language and called it HTML.

➔ HTML Generations

The oldest version of HTML is called HTML 0. This is read either as HTML version 0 or HTML level 0. HTML 1 is an upgradation of HTML 0. It has new tags for highlighting a text and displaying images. In HTML 2, edit boxes, list boxes, and buttons were introduced. In HTML 3 flexible figure handling procedures were included. It also supports mathematical equations, formulas, a banner area and has several other interesting features. It also makes table formulation easy. The present version in use is HTML 4.0.

➔ HTML Tags

HTML is a set of special codes that can be embedded directly into the text to add formatting and linking information. HTML is specified as TAGS in an HTML document. Tags are instructions that are embedded directly into the text of the document. An HTML tag is a signal to a browser that it should do something other than just how text up on the screen. By convention all HTML tags begin with an open angle bracket (<) and end with a close angle bracket (>).

HTML tags can be of two types :

- **Paired Tags**
- **Singular Tags**

- **Paired Tags**

  A tag is said to be a paired tag if it, along with a companion tag, flanks the text. For example the <I> tag is a paired tag. The <I> tag with its companion tag </I> causes the text contained between them to be rendered in Italic. The effect of other paired tags is applied only to the text they contain.

  In paired tags, the first tag (<I>) is often called the opening tag and the second tag </I> is called the closing tag. The opening tag activates the effect of that particular tag which the closing tag turns that effect off. In paired tag if you forget to put closing tag. The particular effect start from the first tag and goes upto the Last character.

- **Singular Tags**
  The second type of tag is the singular or stand-alone tag. A stand-alone tag does not have a companion tag. For example <BR> tag insert a line break. This tag does not require any type of end tag.

Attributes of tags

Some HTML tags required additional information to be supplied to them. For instance, when a picture is placed on the screen, information like the height and width of the picture can be specified. Additional information supplied to an HTML tag is known as Attributes of a tag. Attributes are written immediately following the tag, separated by a space. Multiple attributes can be associated with a tag, also separated by a space.

➔ The structure of an HTML page

Every HTML page has a rigid structure. The entire web page is enclosed within <HTML> and </HTML> tags. Within these tags two distinct sections are created using the <HEAD> </HEAD> and <BODY> </BODY> tags. These are given as under :

For example,

```
<HTML>
    <HEAD> ……      </HEAD>
    <BODY>
        .
    </BODY>
</HTML>
```

■ **Document Head <HEAD> </HEAD>**

Information placed in this section is essential to the inner working of the document and has nothing to do with the content of the document. With the exception of information contained with the <TITLE> and </TITLE> tags, all information placed within the <HEAD> </HEAD> tags is not displayed in the browser. The HTML tags used to indicate the starts and end of the head section are :

```
<HEAD>
    <TITLE> Hello World </TITLE>
</HEAD>
```

■ **Document Body <BODY> </BODY>**

The tags used to indicate the start and end of the main body of textual information are <BODY> and </BODY> tags. Pages defaults like background color, text color, font size, font weight and so on can be specified as attributes of the <BODY> tag. The attributes that the <BODY> tag takes are :

Attribute	Description
BGCOLOR	Changes the default background color to whatever color is specified with this tag. The user can specify a color by name or its equivalent hexadecimal number.
BACKGROUND	Specifies the name of the Gif file that will be used as the background of the document. This Gif tiles up across the page to give a background.

TEXT	Changes the body text color from its default value to the color specified with this attribute.
LINK	Specifies the color of the specified links which are not visited.
VLINK	Specifies the color of the visited links
ALINK	Specifies the link which we are not visiting.

For example,

```
<BODY BGCOLOR=gray TEXT=blue
BACKGROUND="C:\windows\Setup.bmp"> .....

.....</BODY>
```

■ **Title <TITLE> </TITLE>**

A Web page could have a title that describes what the page is about without being too wordy. This can be done using <TITLE> tag. Text included between <TITLE> and </TITLE> comes as a title of that web page in the browser window.

■ **Footer <ADDRESS> </ADDRESS>**

Just as a title can be placed in the title bar of browser window, certain information is commonly placed at the foot of the web page. Copyright information, contact details of the creator of the Web page and so only are the type of information generally placed at the foot of the web page. The tags specified as <ADDRESS> </ADDRESS>.

This tag should ideally be placed immediately after the last line of the textual material of the web page. However, it can also be placed anywhere in the body of the document.

➔ Text Formatting Tags

Obviously, if we are creating a web page, we might want to format it in different ways. The main thing of designing web pages to format the layout of the web page you are creating. In HTML, there are some tags which are used for the formatting the text in different ways. They are explained as under :

■ **Paragraph Breaks <P> </P>**

In HTML by default word wrapping facility is not given. So we have to make a paragraph break to our line. Otherwise our web page becomes horizontally wide. A blank line always separates paragraphs in textual material. The tag is <P> </P>. It keeps one blank line between two lines where this tag is specified.

■ **Line Breaks
**

This is a singular tag which is used to just move the text to the next line. The only difference between <BR> tag and <P> </P> tag is that <P> </P> tag puts one blank additional line while <BR> just puts text to the next line. It doesn't put a blank line.

■ **Making headings (Heading Styles <H1> </H1> to <H6> </H6>)**

Sometimes in our page we might want to display some information in a form of levels i.e. in sub points means main heading and its sub heading and so on. HTML provides different levels of headings. The highest-level header format is <H1> </H1> and the lowest is <H6> </H6>. All the styles appear in Bold and the size of the heading depends on the level which is supplied to it.

For example,

<H1> H.N.S College </H1> looks like H.N.S College

<H4> H.N.S College </H4> looks like H.N.S College

■ **Drawing Horizontal Lines <HR>**

We can put a Horizontal line. <HR> tag draws lines and horizontal rules. This tag draws a horizontal line across the whole page, wherever specified. The attributes to the <HR> tag are :

Attribute	Description
ALIGN	Aligns the lines on the Browser screen, which is by default, aligned to the center of the screen. LEFT, RIGHT and CENTER will align the line respectively to the Left, Right or to the Center.
SIZE	Changes the size of the rule.
WIDTH	Sets he width of the rule. It can be set to a fixed number of pixels, or to a percentage of the available screen width.
COLOR	Specifies the color of Horizontal Line
NOSHADE	Does not put the shade in Horizontal Line.
SRC	Image with path can be specified to fill in the Horizontal Line

For example,

<HR ALIGH=LEFT WIDTH=50% SIZE=4 COLOR=blue NOSHADE>

- **Bold <B> </B>**

This tag displays the text in Boldface style. It only bolds the specified text within the <B> and </B> tag. For example,

<B> Hello </B> gives output like Hello

- **Italic <I> </I>**

This tag displays the text in Italic style. It only makes the specified text in italic format. The specified text within the <I> and </I> tag. For example,

<I> Hello </I> gives *Hello*

- **Underline <U> </U>**

This tag which is used to underline the text is <U> and </U> . For example,

<U> Hello> </U> gives Hello

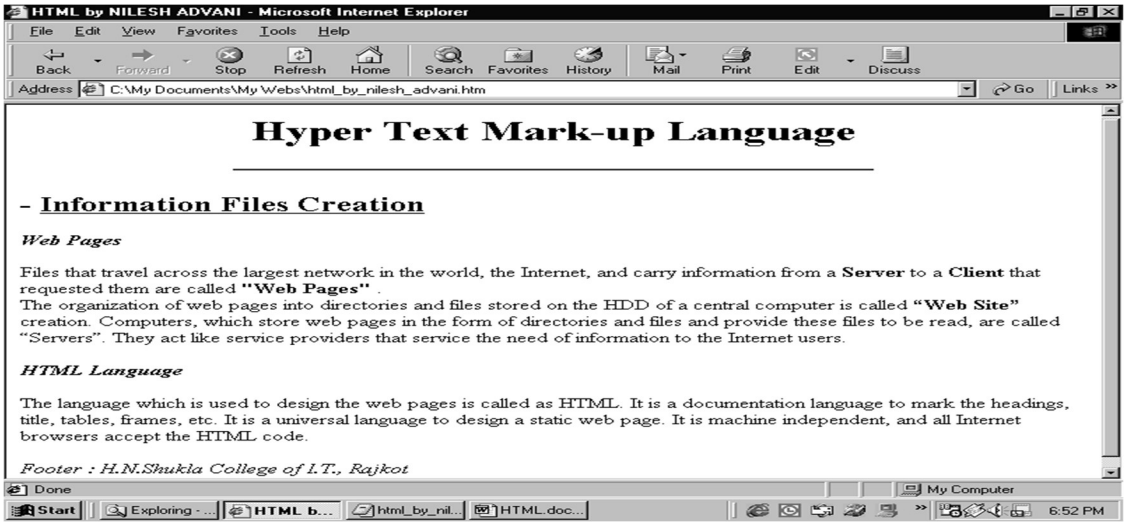
➔ Example using all above Tags

Here is the code which uses all the above mentioned tags with almost all their attributes. Output of this code in browser window is also shown.

```
<html>
<head>
<title>HTML by NILESH ADVANI</title>
</head>
<body TEXT=#000000 bgcolor=white>

<H1 align=center> Hyper Text Mark-up Language </H1>

<HR <hr width="60%" color=red noshade>

<H2> - <U>Information Files Creation </U> </H2>

<H4> <I> Web Pages </I> </H4>

<P> Files that travel across the largest network in the world,
the Internet, and carry information from a <B>Server
</B>to a <B> Client </B> that requested them are called
<B> "Web Pages" </B>.

<BR>The organization of web pages into directories and

files stored on the HDD of a central computer is called
<b>“Web Site”</b> creation. Computers, which store web
pages in the form of directories and files and provide these
files to be read, are called “Servers”. They act like service
providers that service the need of information to the
Internet users. </P>
```

```
<H4> <I> HTML Language </I> </H4>

<P> The language which is used to design the web pages is called as HTML. It is a documentation language to mark the headings, title, tables, frames, etc. It is a universal language to design a static web page. It is machine independent, and all Internet browsers accept the HTML code. </P>

<ADDRESS> Footer : H.N.Shukla College of I.T., Rajkot </ADDRESS>

</body>

</html>
```

Output



■ **Making paragraph Strikethrough <STRIKE> </STRIKE>**

It make the selected portion of our paragraph to Strikethrough by using a paired tag which is <STRIKE> and </STRIKE>

For example, <STRIKE> Hello how are you? </STRIKE> prints ~~Hello how are you?~~

■ **Making characters Superscript **

Characters can be superscripted using these tag. Generally this tag is used to display the Formulas and dates with ordinals

For example, 5 th March, 1975 will show 5th March, 1975

■ **Making characters Subscript **

Characters can be subscripted using these tag. Generally this tag is used to display the Scientific Formulas.

For example, H ₂ SO ₄ will show H_2SO_4

■ **Making characters Larger and Smaller with <BIG> </BIG> and <SMALL> </SMALL>**

We can display the characters in larger font size or smaller font size of the current size. If we specify the text between <BIG> and </BIG> size of font is increased from the current font size which is going. The same happens with <SMALL> </SMALL> also.

For example,

<BIG> Hello </BIG> Hello <SMALL> Hello </SMALL> will show Hello Hello Hello

■ **Centering Text, Images, etc. <CENTER> </CENTER>**

This tags are used to center everything which comes between them. It may be anything. It may be text, lists, images, rules, tables, or any other type of page element. In short, what comes in between of <CENTER> and </CENTER> tag, comes in the center by default. If we are using this we don't have to specify ALIGN attribute of any of the tag.

For example, <CENTER> Hello how are you? </CENTER> puts the specified line in the center of the current line.

- **Spacing – Indenting Text <SPACER>**

This is a singular tag. It is used for inserting blank spaces in an HTML document. The attributes of <SPACER> tag are specified as under :

Attribute	Description
TYPE	To specify whether space has to be Horizontally or Vertically. TYPE = "horizontal" indicates the horizontal space to be left and TYPE = "vertical" indicates the vertical space to be left
SIZE	Indicates the amount of space to be left. Size is specified in terms of an Integer value.

Note : The SPACER command is understood only by the Netscape Browser. So it might happen that in your browser it may not accept or it may not its effect.

For example,

<SPACER TYPE="horizontal" SIZE=90> Hello how are you?

- **Specifying Fonts <FONT> </FONT>**

This is the tag which is used very frequently as it is used to specify the fonts and different styles related to font. The tag is <FONT> and </FONT>. The attributes of this tags are specified as follows:

Attribute	Description
FACE	This is used to specify the name of font. Font name is specified
SIZE	Specifies the size of font. Size can take value between 1 To 7 which is predefined set of size according to the level. It can also be set to relative to the default size. For example, SIZE = +4 specifies add 4 points to the current size of font.
COLOR	Specifies the color of fonts which are used.

The FACE attribute must be any of the fonts which are supported by Windows. If the specified font is not available, the browser uses its

default font. Therefore, it is always a good practice to specify commonly used fonts.

For example,

```
<FONT FACE="impact" SIZE=6 COLOR=red>
    Hello how are you?
</FONT>
```

■ Specifying Different Links <LINK>

It is a singular tag which is used for the following purposes.

- To inform the browser of the previous document.
- To inform the browser of the next document.
- To link the banner.
- To inform the location of the base document location.
- To specify the Style Sheet.

This tag is specified in the <HEAD> </HEAD> tag just next to the <TITLE> tag. Following attributes are used with <LINK>

Attribute	Description
REL	Specifies the type of link. It can be PREVIOUS, NEXT, HOME, BANNER, STYLESHEET, etc.
HREF	Specifies the HTML document file where we want to put the link which is specified. It can be Absolute or Relative path of document.

For example,

```
<LINK REL = home HREF = "d:\nilesh\html\index.html">
```

- **Base Element <BASE>**

It is a singular tag which is used to specify the location of all the base documents which are available. This helps when we have specified the wrong path because it searches then the specified files in the path which is specified in <BASE> tag.

For example,

<BASE HREF = "http://www.xavier.edu">

➔ Understanding Colors

Any color is a combination of three basic colors, which are

- **Red**
- **Green**
- **Blue**

In HTML, each of the above colors has a 00 To FF which is in Hexadecimal form. In numeric it has a range from 0 to 255. If we give FF it means 255 and if we give 00 it means empty. If we want to specify the color code in hexadecimal format, code should begin with # sign in COLOR attribute.

For example, If we want Green color, we can specify in hex code as :

Red = 00 (Empty)

Green = FF (FULL)

Blue = 00 (Empty)

Which means #00FF00

The basic colors are Red, Green and Blue where each have 256 degrees, so we can create different 1,67,77,216 color combinations as 256 x 256 x 256.

We can specify the colors by giving their names which are in a form of constants in HTML, but if we are specifying the color in this way, we can specify only few of the color names, but if we are using the Hexa-Decimal format to specify the colors, we can specify up to 1,67,77,216 colors.

Here is a table of some common color code with their values,

Color	Code
White	#FFFFFF
Red	#FF0000
Green	#00FF00
Blue	#0000FF
Magenta	#FF00FF
Cyan	#00FFFF
Yellow	#FFFF00
Black	#000000
BlueViolet	#9F5F9F
Brass	#B5A642

■ **Specifying TeleType / Monospaced Font <TT> </TT>**

The fonts which each occupies same no. of width and height is called as TeleType or Monospaced Font. The characters which are printed with TypeWriter are called as Teletype fonts because any of those characters occupy same no. of amount in terms of space.

If fonts are Times New Roman. S and I occupies different amount of space. But if it is a Courior Font it occupies same amount of space which is `s` and `i` . We can specify Teletype or Monospaced Fonts by giving <TT> </TT> tag.

For example, if we give

<TT> Hello how are you? </TT>, it will print

```
Hello how are you?
```

■ **Specifying Base Font by <BASEFONT>**

At the beginning of the document, the default font size for the entire page can be selected using the <BASEFONT> singular tag.

For example,

<BASEFONT FACE=arial SIZE=5>. This statement specifies Arial with size of 5 for the entire document.

■ **Preformatting Text <PRE> </PRE>**

The main problem with HTML language is it can not detect any spaces which is typed in the code. So if space is specified it is equivalent to no spaces are specified.

For example, if we specify following code,

```
<P> No.    Name                    Phone </P>
<P> 1.     abcd                    473081 </P>
<P> 2.     hello                   224362 </P>
<P> 3.     Bill Gates              323443 </P>
<P> 4.     Microsoft Corporation    323432 </P>
```

will give output as follows

But, if we specify the following code

```
<PRE>

    No.   Name                    Phone

    1.    abcd                    473081

    2.    hello                   224362

    3.    Bill Gates              323443

    4.    Microsoft Corporation    323432

</PRE>
```

■ **Using Special Characters with &**

Special characters such as <, >, etc. can be included in the web page using escape codes which begin with the ampersand (&) symbol. The ampersand symbol must be followed by the mnemonic keyword for the symbol or it should have the ASCII code of the symbol. The list of special characters with mnemonic and ASCII code is listed as below :

Mnemonic	Symbol	Description	Decimal
Lt	<	Less than	#60
Gt	>	Greater than	#62
Amp	&	Ampresand	#38
AElig	Æ	Capital AE diphthong	#198
O slash	Ø	Capital oh slash	#216
Quote	'	Single quote	#62
	£	Pound sign	#163
	©	Copyright sign	#169
	±	Plus or Minus	#177
	¼	Quarter	#188

	½	Half	#189
	¾	Three-quarters	#190

➔ Using Images and Pictures <IMG> </IMG>

A small picture is equivalent to a thousand words. Pictures immediately attract attention. The play a dominant role in educational technology. In a web page, it is strongly recommended that a number of relevant pictures must be inserted. A picture or an image in the web page can be inserted using the <IMG> tag. The <IMG> tag has several attributes to inform the source, height of the picture, width of the picture alignment, etc. The following are its attributes :

Attribute	Description
SRC	Used to specify the path of the picture. It may be relative or absolute.

HEIGHT	Specifies the height of the image. If height not specified, it is taken relative.
WIDTH	Specifies the width of the image. If width is not specified, it is taken relative.
ALIGN	Used to specify the alignment of the picture.
ALT	When picture is loaded or if it is unable to load the picture, the text written in ALT is shown inplace of picture.
HSPACE	Specify the horizontal space
VSPACE	Specify the vertical space
BORDER	Used to specify the border width of the image. Default it is none (0).
LOWSRC	Used to specify the path of the Low Resolution Image

Here some different types of alignment are shown.

Alignment	Effect
Left	Placed at the left edge of page
Right	Placed at the right edge of page
Middle	Placed on the middle of the page.
Top	Aligned with the top of the tallest item on the line.
Text Top	Aligned with the top of the tallest item on the line. The text top has usually same effect as top
Abs middles	Aligns the middle of the current line with the middle of the image
Baseline	Aligns the bottom of the image with baseline of the current line.

■ Low Resolution Images

Whenever the image file given in the SRC tag is of high resolution, the browser takes some time to load the image file. It is possible to show an image of low resolution until the high resolution image is loaded. We can inform browser to first load low resolution image until high resolution image is not loaded.

For example,

<IMG SRC="c:\firstlogo.jpg" LOWSRC="c:\secondlogo_low.jpg"> will first show firstlogo_low.jpg image until secondlogo.jpg image is not loaded in the browser.

➔ Different Image Formats

Images can be stored in any one of the following popular formats :

- **BIT Map (bmp) Format**
- **Graphics Interchange (gif) Format**
- **Joing Photographic Expert Group (jpeg) Format**

The Bit Map format is most common. The GIF format was created by CompuServe to provide a means of exchanging graphic images quickly. Files of gif have a color depth of 8 bits per pixel for a total of 256 colors.

The fig file format uses compression technique called LZW compression – LZW comes from the names of Lampel, Niv and Welch. Lampel and Ziv were two mathematicians who were the originators of several compression techniques. Welch designed the LZW compression scheme with the help of the theory developed by Lampel and Ziv.

The jpeg format supports 24-bit color, which gives upto 16.7 million colors. In fact, jpeg is a compression scheme. Files that use the jpeg compression schemes are said to be in jpeg format.

There is yet another image format called Portable Network Graphics (PNG) format, which is supported by the Internet Explorer. It uses a compression technology called deflation. The png format has full=color non-glossy image compression, which makes files smaller and quicker to load. Whenever we want to display a png format image in a web page, we must use the <EMBED> tag instead of the <IMG> tag.

For example,

```
<EMBED SRC="c:\firstlogo.png" height=300>
```

➔ Different types of Lists

When we want to mention a list of items, there are two methods of doing so. We can number them as 1, 2, 3, … etc. or we can list them one below the other without numbers using some symbols. When we list them without numbers, it is called an Unordered List and when we list the items with number, it is called as an Ordered List. One another type of list is there, which is called Definition List. So, there three types of lists which are :

- **Unordered List**
- **Ordered List**
- **Definition List**

■ **Unordered Lists <UL> </UL>**

An unordered list is represented by the <UL> and </UL> tags. The <UL> tag is given at the beginning and the </UL> tag is given at the end. Each list item is given an <LI> (List Item) tag, which is singular tag.

<UL> tag has one attribute through which we can specify the Type of Unordered List. That is TYPE attribute. It has following options

- **DISC (will give a solid round black circle.) <UL TYPE="DISC">**
- **SQUARE (will give a solid square black bullet.) <UL TYPE="SQUARE">**
- **CIRCLE (will give a black bordered circle.) <UL TYPE="CIRCLE">**

■ **Ordered Lists <OL> </OL>**

Lists which appear with numbers are called ordered lists. The HTML code for as ordered list is similar to that of an unordered list, except that we use an <OL> tag instead of a <UL> tag. <OL> tag has more attributes than <UL> tag, because when we are using numbers there are many options that we may apply.

Attributes of <OL> are

Attribute	Description
TYPE	Controls the numering scheme to be used. TYPE="1" will give counting number (1, 2, 3, …) TYPE="A" will give Uppercase Letters (A, B, C, ….) TYPE="a" will give Lowercase letters (a, b, c ….) TYPE="I" will give Uppercase Roman Numerals (I, II, III, …) TYPE="i" will give Lowercase Roman Numerals (i, ii, iii,….)
START	As we are using numbers, it indicates the starting of number.
VALUE	Changes the numbering sequence in the middle of an ordered list. It is to be specified with the <LI> tag.

■ **Definition Lists <DL> </DL>**

Definition list values appear within tags <DL> and </DL>. Definition lists consists of two parts. There is no difference between Ordered/Unordered Lists and Definition lists. Only Definition list shows lists without having any number or symbol. Simply it puts indentation wherever specified. Following are two types of tags in Definition List :

<DT> Definition Term/Title.

<DD> Definition Description

```
<DL>
    <DT> Keyboard
    <DD> Input Device
    <DT> Printer
    <DD> Output Device
</DL>
```

will give output as

Keyboard

Input Device

Printer

Output Device

➔ Defining LINKS <A> </A>

HTML allows linking to other HTML documents as well as images. Clicking on a section of text or an image in one web page will open an entire web page or an image. The text or an image that provides such linkages is called HyperText, a Hyperlink or a Hotspot.

The browser distinguishes Hyperlinks from normal text, Every hyperlink has following characteristics :

- **Appears blue in color which is default. We can change this color.**
- **The hyper linked text or image is underlined.**
- **When the mouse cursor is placed over it, the standard arrow shaped mouse cursor changes to the shape of a hand.**

Links are of three types the color of which are specified in <BODY> tag:

- **LINK (Color of hypertext)**
- **ALINK (Link which is being clicked)**
- **VLINK (Already visited link)**

Links are created in a web page by using the <A> and </A> tags. Anything written between these tags becomes a hyperlink/hotspot.

By clicking on the hyperlink navigation to a different web page or image takes place.

The document to be navigated to needs to be specified. By using HREF attribute of the <A> tag the next navigable web page or image can be specified.

For example,

<A HREF=”nilesh.html”> My Site </A>

puts a hyperlink on My Site and when clicked moves to NILESH.HTML web page.

Hyperlinks can be of two types :

- **Links to an external document**
- **Links to the same document at a specific place.**

■ Link to an external document

When in a link, name of other HTML document is given it is called as Link to an external document. This is done simply the same way as it is shown above in the example. We have to write the path and document name which is to be linked between <A> and </A> tag.

For example,

<A HREF=”nilesh.html> My Site </A>. This will put a link on word My Site like shown and by clicking on the specified document is open.

By default, a hyperlink takes a user to the beginning of the new web page. Some times it might e necessary to jump to a particular location within the other web page. To do this we have to specify a Named Anchor (Bookmark) of that external web page. We can do this by using NAME attribute of Anchor tag.

This process can be summarized into two steps :

- **Specify the Named Anchor (Bookmark)**
- **Specify the link to that web page and named anchor.**

For example,

1) **<A NAME=”chapter-1”> Chapter-1 </A> in the external document.**
2) **In page from link is to be specified. It should be as follows :**
   <A HREF=”chapters.htm#chapter-1”> Chapter-1 </A>

■ **Link to an internal document**

Sometimes, a link is required to the same page but in different location when our web page is too large to show the information.

As the jump has to be done at a specific location, the procedure is same as linking to the named anchor tag of another page. The only difference is that in this name of external web page is not specified. Directly the named anchor is specified which links directly to the same location of the same web page.

For example,

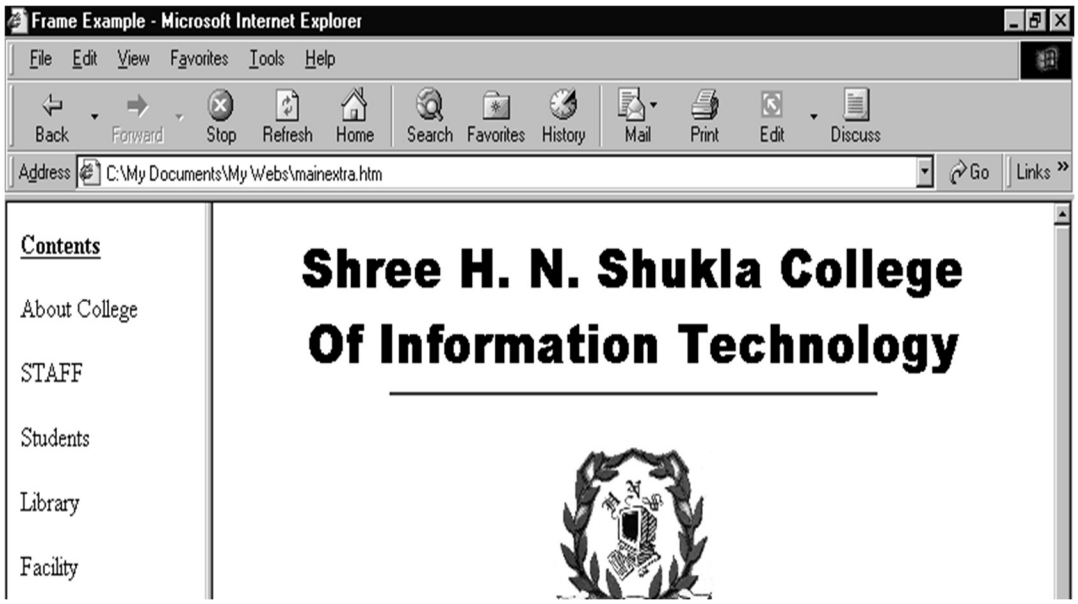
1. **<A NAME="chapter-1"> Chapter-1 </A> in the external document.**
2. **In page from link is to be specified. It should be as follows :**
   <A HREF="#chapter-1"> Chapter-1 </A>

- **Link anywhere but in Different Target Frame**

Sometimes, if we are using Web pages which are created using Frames, we are specifying the link to any particular web page which should be open in another target frame. In this situation, target frame has to be specified, which indicates that if you are clicking in a link the target web page will be opened in another frame. Target attribute is used with <A> tag.

For example,

In our web page there are two frames



No if we want to put a link that if we click on STAFF hyperlink, it should open the target HTML document in the near frame. Frame-1 is "contenxt" and Frame-2 is "details". So following HTML command will be written for that :

<A HREF="c:\Nilesh\HNS\staff.htm" TARGET="details"> STAFF </A>

(FRAMES WILL BE DISCUSSED IN LATER CHAPTERS)

➔ Tables <TABLE> </TABLE>

A table is a two dimensional matrix, consisting of rows and columns. Tables are intended for displaying data in columns on a web page. All table related tags are included between the <TABLE> and </TABLE> tags. Each row of a table is described between the <TR> and </TR> tags. Each columns of a table is described between the <TD> and </TD> tags.

Table rows can be of two types :

- **Header Rows**
  A table header row is defined using <TH> and </TH> tags. The content of a table header row is automatically centered and appears in boldface.

- **Data Rows**
  There could be a single data cell or multiple data cells.

The attributes that can be included in the <TABLE> tag are :

Attribute	Description
ALIGN	Horizontal alignment. It can be LEFT,CENTERor RIGHT.
VALIGN	Vertical alignment. It can be TOP, MIDDLE or BOTTOM.
WIDTH	Sets the WIDTH to a specific number of pixels or to a percentage of the available screen width. If not specified, the data cell is adjusted based on the cell data value.
BORDER	Controls the border to be placed around the table. The thickness is specified in pixels.
BORDERCOLOR	Used to specify the color of border.
BORDERCOLORLIGHT	Used to specify Top and Left Border Color of table. (Shaded)
BORDERCOLORDARK	Used to specify Bottom and Right Border Color of table.(Shaded)
BGCOLOR	Used to specify the background color of table.
BACKGROUND	Used to specify the Background image path of table.
CELLPADDING	This attribute controls the distance between the data in a cell and the boundaries of the cell.

CELLSPACING	Controls the spacing between adjacent cells.
COLSPAN	This attribute inside a <TH> or <TD> tag instructs the browser to make the cell defined by the tag to take up more than one column. This can be sset equal to number of columns the cell is to occupy which is called as Merging cells.
ROWSPAN	This works in the same way as the COLSPAN works except that it allows a cell to take up more than one row. The attribute can be set by giving a numeric value.

Attributes of <TH> and <TD> tags are,

Attribute	Description
BGCOLOR	Used to specify the background color of cell
BORDERCOLOR	Used to specify the border color of specific cell.
BORDERCOLORLIGHT	Same as above.
BORDERCOLORDARK	Same as above.
BACKGROUND	Used to specify the Background image path of particular cell.

■ **The Caption Tag <CAPTION> </CAPTION>**

This tag is used to show the header or footer of the table. Header or Footer is specified between <CAPTION> and </CAPTION> tags. The table caption can be made to appear above or below the table structure with the help of ALIGN attributed. By default, it puts caption on the header. If we use <CAPTION ALIGN=”top”>, caption comes in top and if we use <CAPTION ALIGN=”bottom”, caption comes at bottom in table.

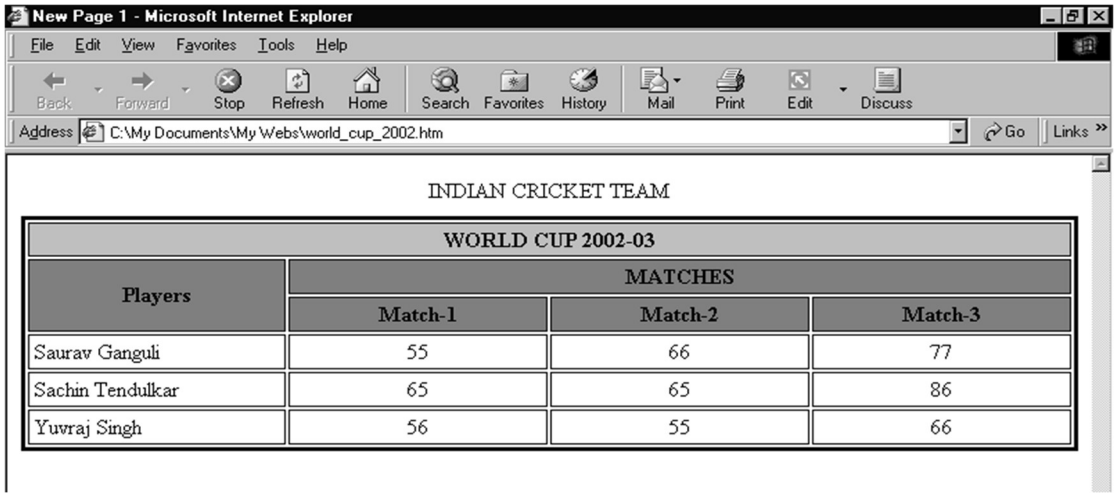
Consider following code and output

```
<table border="3" cellpadding="3" width="100%" bordercolor="#000080">
<CAPTION> INDIAN CRICKET TEAM </CAPTION>
 <tr>
  <td width="100%" colspan="4" bgcolor="#C0C0C0">
   <p align="center"><b>WORLD CUP 2002-03</b></td>
 </tr>
 <tr>
  <td width="25%" rowspan="2" bgcolor="#808080" align="center">
<b>Players</b></td>
```

```
  <td width="75%" colspan="3" bgcolor="#808080" align="center">
<b>MATCHES</b></td>
 </tr>
 <tr>
  <td width="25%" bgcolor="#808080" align="center"><b>Match-1</b></td>
  <td width="25%" bgcolor="#808080" align="center"><b>Match-2</b></td>
  <td width="25%" bgcolor="#808080" align="center"><b>Match-3</b></td>
 </tr>
 <tr>
  <td width="25%">Saurav Ganguli</td>
  <td width="25%" align="center">55</td>
  <td width="25%" align="center">66</td>
  <td width="25%" align="center">77</td>
 </tr>
 <tr>
  <td width="25%">Sachin Tendulkar</td>
  <td width="25%" align="center">65</td>
```

```
  <td width="25%" align="center">65</td>
  <td width="25%" align="center">86</td>
 </tr>
 <tr>
  <td width="25%">Yuvraj Singh</td>
  <td width="25%" align="center">56</td>
  <td width="25%" align="center">55</td>
  <td width="25%" align="center">66</td>
 </tr>
</table>
```

Output in browser :



INDIAN CRICKET TEAM

WORLD CUP 2002-03			
Players	MATCHES		
	Match-1	Match-2	Match-3
Saurav Ganguli	55	66	77
Sachin Tendulkar	65	65	86
Yuvraj Singh	56	55	66

➔ Frames

The browser shows the web page through a window. We scroll the web page and see the entire document through the window of the browser. The window is called the Container. The browser screen could not be split into separate sections, show different but related information. It is possible to divide the container into several frames.

The HTML tags that divide a browser screen into two or more HTML recognizable unique sections is the <FRAMESET> and </FRAMESET> tags. Each unique part is called as a Frame. Frames have the following characteristics.

- **Each frame is given a name**
- **Each frame will be targeted by an HTML document**
- **Each frame resizes itself dynamically in respond to the changes in the size of visible area.**

■ Specifying FRAMESET using <FRAMESET> </FRAMESET> tag

This tag is used to show the header or footer of the table. Header or Footer is specified between <CAPTION> and </CAPTION> tags. The table caption can be made to appear above or below the

The splitting of a browser screen into frames is done by <FRAMESET> and </FRAMESET> tags. The <FRAMESET> tag require one of the following two attributes - rows or columns.

Part	Description
ROWS	Used to divide the screen into multiple rows. It can be set equal to a list of values. Depending on the required size of each row. The values can be : A number of pixels Expressed as a percentage of the screen resolution. The symbol *, which indicates the remaining space. Example-1 : <FRAMESET ROWS=”170,170,200”> . </FRAMESET> Example-2 : <FRAMESET ROWS=”30%,30%,40%”> . </FRAMESET>

COLS	Used to divide the screen into multiple columns. It can be set equal to a list of values. Depending on the required size of each column. The values can be : A number of pixels Expressed as a percentage of the screen resolution. The symbol *, which indicates the remaining space. Example-1 : <FRAMESET COLS=”100,250,100” . </FRAMESET> Example-2 : <FRAMESET COLS=”25%,50%,25%” . </FRAMESET>

- **Specifying HTML Documents in each Frame using <FRAME> tag**

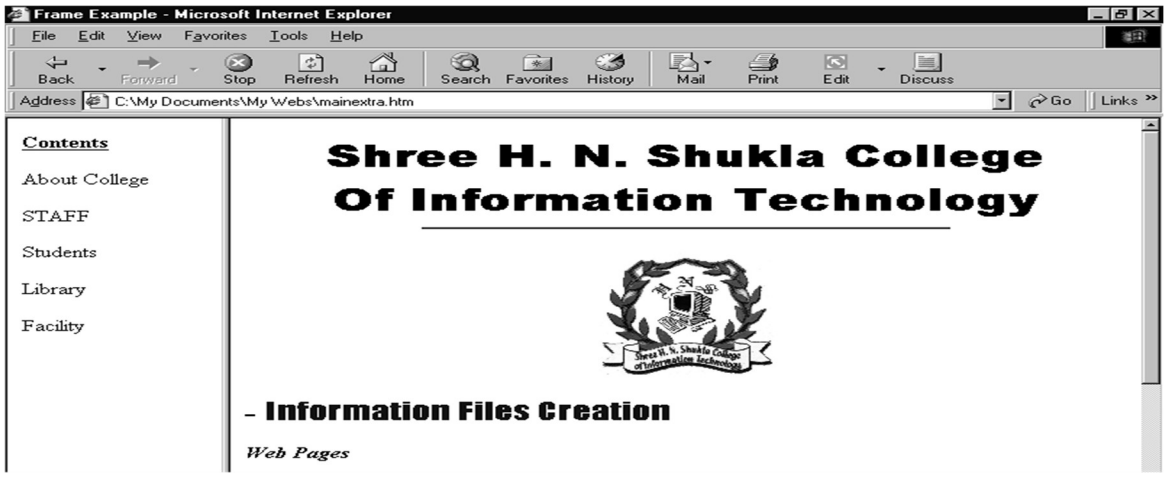
It is a singular tag which is used to specify the HTML documents to be open in each section (frame) of the browser. It has following attributes :

Attribute	Description
SRC	Used to specify the URL of the document to be loaded in frame.
MARGINHEIGHT	Specifies the amount of white space at top and bottom of frame
MARGINWIDTH	Specifies the amount of white space at left and right side of frame.
NAME	Used to specify the name of each Frame
NORESIZE	Doesn't allow viewer to resize the frame.
SCROLLING	Allows scrolling frame. Can be YES, NO or AUTO

- **Different Examples of Frames**

1) **Frame with Two Columns**

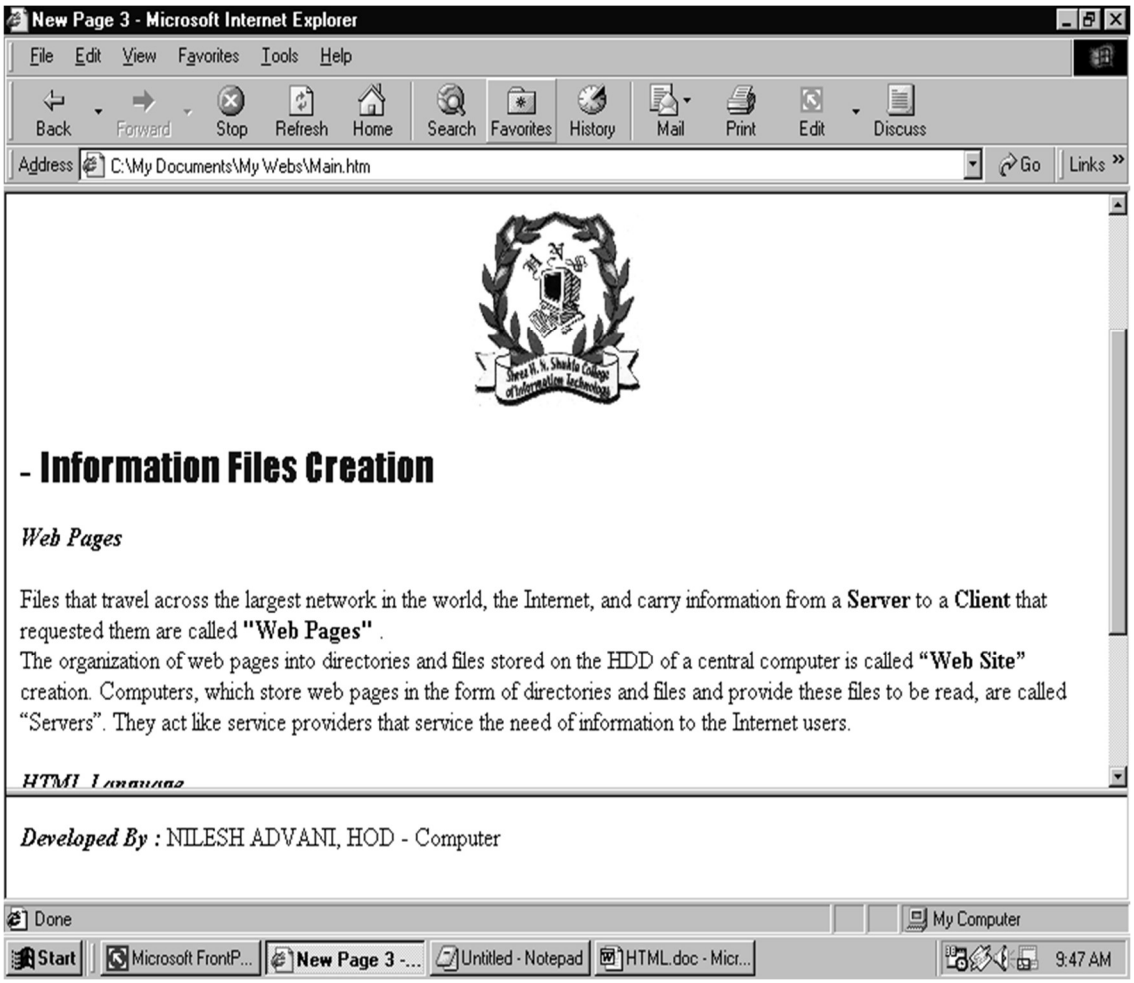
```
<html>
    <head>
    <title>Frame Example</title>
    </head>
    <frameset cols="150,*">
    <frame name="contents" target="main"
src="contents.htm">
      <frame name="main"
src="html_by_nilesh_advani.htm">
    </frameset>
</html>
```



2) Frame with Two Rows

```
<html>


<head>

<title>Frame Example</title>

</head>


<frameset rows="*,64">

 <frame name="main" src="html_by_nilesh_advani.htm">

 <frame name="footer" scrolling="no" noresize target="main" src="NileshAdvani.htm">


</frameset>


</html>
```

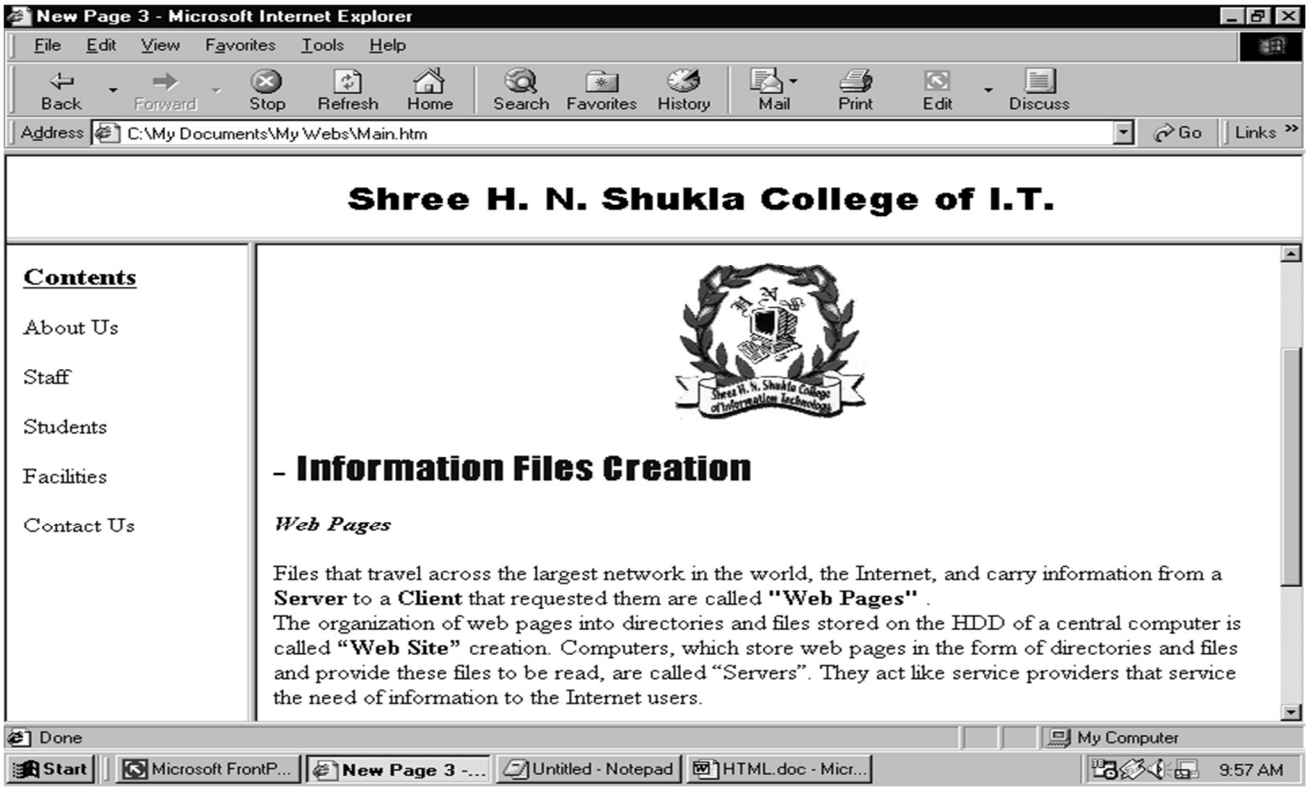
Output :


New Page 3 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Address C:\My Documents\My Webs\Main.htm
Go Links
- Information Files Creation
Web Pages
Files that travel across the largest network in the world, the Internet, and carry information from a Server to a Client that requested them are called "Web Pages" .
The organization of web pages into directories and files stored on the HDD of a central computer is called "Web Site" creation. Computers, which store web pages in the form of directories and files and provide these files to be read, are called "Servers". They act like service providers that service the need of information to the Internet users.
Developed By : NILESH ADVANI, HOD - Computer
Done
My Computer
Start
Microsoft FrontP...
New Page 3 -...
Untitled - Notepad
HTML.doc - Micr...
9:47 AM

3) Frame with Multiple Rows and Columns

```
<html>

<head>

   <title>Frame Example</title>

</head>


<frameset rows="64,*">

   <frame name="banner" scrolling="no" noresize target="contents" src="heading.htm">

    <frameset cols="150,*">

        <frame name="contents" target="main" src="contents.htm">

        <frame name="main" src="html_by_nilesh_advani.htm" scrolling="auto">

    </frameset>

</frameset>

</html>
```

Output



➔ Dynamic HTML (DHTML)

An HTML document can be viewed as a container having several objects like paragraphs, images, tables, lists, etc. This method of designing is called Document Object Model (DOM). In DOM, each object of a document can have properties associated with it. It is

also possible for some objects to have access to databases or any scripting languages such as VB Script, Java Script, etc. Such facilities are provided by Dynamic HTML (DHTML). It is beyond the scope of this book to completely discuss the features of DHTML. Dynamic HTML has the following features :

- **Style Sheets**
- **Absolute Positioning (Layers)**
- **Multimedia Effects**
- **Database Access Facility**
- **Dynamic Fonts**
- **Scripting**

■ Style Sheet

The browser has a default style to present the web page as per the mark-ups given in the HTML document. We can change those default styles and create a style to match our taste using Style Sheets.

A Style Sheet is a list of statement which declares the styles of various components such as headings, paragraphs, etc. The only style sheet format that is used in all HTML documents is the text/css format. The acronym of CSS is for Cascading Style Sheets.

Style Sheets are powerful mechanism for adding style (e.g. fonts, colors, spacing, etc.) to Web pages. They enforce standards and uniformity throughout a web site and provide numerous attributes to create dynamic effects.

The advantages of a Style Sheet includes the ability to make global changes to all documents from a single location. Style Sheets are said to cascade when they combine to specify the appearance of page.

The style assignment process is accomplished with the <STYLE> and </STYLE> tags. The syntax for making the assignment is simple. Between <STYLE> and </STYLE> HTML tags, specific style attributes are listed. The <STYLE> and </STYLE> tags are written within the <HEAD> .. </HEAD> tags.

The syntax of defining Style Sheets is :

```
<STYLE Type="text/css">
        tag-1 {attribute:value; attribute:value,…}
        tag-2 {attribute:value; attribute:value,…}
        .
        tag-n {attribute:value; attribute:value,…}
</STYLE>
```

Note :

In the <STYLE> tag, the expression "Type = text/css" indicates that the style sheet conforms to CSS syntax.

The attributes that can be specified to the <STYLE> tag are Font attributes, Color and Background attributes, Text attributes, Border attributes, Margin attributes and List attributes.

- **Font Attributes**

Attribute	Values
font-family	A comma-delimited sequence of font family names
font-style	Normal, italic or oblique
font-weight	Normal, bold, bolder, lighter, or one of the nine numerical values (100, 200, 300, 400, 500, 600, 700, 800, 900)
font-size	A term that denotes size (xx-small, x-small, small, medium, large, x-large, xx-large), relative size (larger, smaller), a number (of pixels), percentage

For example,

```
<STYLE Type="text/css">
    H1 {font-family : arial, impact, courier; font-weight : bold}
    P {font-size : 12pt; font-style : italic}
</STYLE>
```

The above style sheets indicates that now in all H1 tags font-family will be of arial, impact, courier means if arial is not found, it will use impact else courier. The P tags will be of 12 point size and italic style.

- **Color and Background Attributes**

Attribute	Values
Color	Sets an element's text-color. May be color name or a hex-code.
background-color	Specifies the color in an element's background. May be color name or a hex-code.
background-image	Sets the background image path.

background-repeat	With a background image specified, sets up how the image repeats through the page. Repeat-x (repeats horizontally), Repeat-y (repeats vertically), repeat (both), no-repeat.

For example,

```
<STYLE Type="text/css">
    H1 {font-family : arial black; color : red; background-image : url(c:\hns.jpb);
font-size : 24pt, font-style : bold}
    H2 {font-family : arial; color : blue; background-color : gray;
        font-size : 20pt, font-style : bold}
    P {color : black; font-size : 12pt}
</STYLE>
```

The above style sheets will put all attributes of H1, H2 and P tags according to the following :H1 : Font family will be of arial black, Text color of H1 heading will be of red, background image at the back of H1 heading will be hns.jpg image, font size of H1 text will be of 24 points with bold face style.

H2 : font family will be of arial, text color blue, background color will be of gray color, font size of 20 points with bold face style. P : font color of paragraph of black with fonts size of 12 points.

- **Text Attributes**

Attribute	Values
text-decoration	Adds decoration to an element's text. None, underline, overline, line-through, blink
vertical-alisgn	Determines vertical position. Baseline, sub, super, top, text-top, middle, bottom, text-bottom, also percentage of the height.
text-transform	Can be Capitalize, uppercase, lowercase, or none
text-align	Specifies alignment of text. Left, right, center, or justify
text-indent	Indents the first line. A percentage of elements' with or a length.

- **Border Attributes**

Attribute	Values
border-style	Solid, double, groove, ridge, inset, outset
border-color	A color name or color code
border-width	Thin, medium, think or length
border-top-width	Thin, medium, think or length
border-bottom-width	Thin, medium, think or length
border-left-width	Thin, medium, think or length
border-right-width	Thin, medium, think or length
border-top	Specifies width, color and style
border-bottom	Specifies width, color and style
border-left	Specifies width, color and style
border-right	Specifies width, color and style
Border	Sets all the properties at once

- **Margin related Attributes**

Attribute	Values
margin-top	Percent, length or auto
margin-bottom	Percent, length or auto
margin-left	Percent, length or auto
margin-right	Percent, length or auto
Margin	Percent, length or auto

- **List Attributes**

Attribute	Values
list-style	Disc, circle, square decimal, lover-roman, upper-roman, lower-alpha, upper-alpha, none

■ **External Style Sheets <LINK>**

External Style Sheets are composed of standard text, which consists of a set of tag In the case an external style sheet, it is stored in a separate file. This includes only Tags and its attributes with values. The file name must be with .CSS extension. Then wherever required link these External Style Sheets with following code : (Consider an External Style Sheet named "stylesheet.css")

<LINK rel="stylesheet" HREF="c:\stylesheet.css" TYPE="text/css">

➔ Forms

Till now we have studied that how to show the information on the Web page with different tags and their attributes. The HTML documents open and show the messages.

But what if we want to get the details from our users. Because till now it was one-way communication. So, forms are the tools which can be used to interface with the users or visitors of our web site.

Using forms, we can design a web page on which a user can communicate with us in many ways like giving his information, wish, opinion, suggestion, etc. When an advertisement is made for a seminar, applicants can apply in a form attached to the advertisement.

So, forms are used to get the details from user/visitor. But here using HTML we can only design the layout of forms, we can not program the information which is given by user, we can not check the details or store them in database using HTML tags.

Because only form designing is supported by HTML tags but its programming, checking, storing, etc. is provided by any of the Scripting Languages like ASP, VBScript, Java Script, CGI, etc. which can store the details and program the information which is given by users.

In Form we can have no. of screen objects using which we can get the details from the user/visitor in number of ways. Following are the objects which are provided in Form to get the details :

- **Check Boxes**
- **Drop Down List**
- **Text Area**
- **Radio Button**
- **Text Box**
- **Submit and Restore Buttons**

A form and all its objects are defined between <FORM> and </FORM> tags which has following attributes :

- ACTION - METHOD - ENCTYPE
-
- ACTION

Forms are used to get inputs from users. When user click the Submit button the information is transferred to the Server. At that time some actions means some sort of checking, storing, etc. activities are performed. The Action attribute informs the browser the location of the server to which the form's information is transferred which includes the program to be executed after clicking the SUBMIT button. Generally it is a file which is created in PERL and stored in CGI-BIN directory of the server. For example,

ACTION = "/cgi-bin/nilesh/check.cgi"

- METHOD

The method attribute has only two choices, GET and POST. With the POST method, the information from the user is given to the database and the back-end program can read the data which is given. In the case of GET method, the data received in the form are placed from the server to the clients form.

- ENCTYPE

This attribute is used to inform the server the way to handle the encryption process. Usually it is set to application/x-www-form-urlencoded.

For example,

```
<FORM ACTION=”/cgi-bin/nilesh/check.cgi” METHOD=”post”
ENCTYPE=”application/x-www-form-urlencoded”>
.
</FORM>
```

- **Drop Down List**

A drop down list presents a list to the user. The user can select his choice from the list. This is done using <SELECT> and </SELECT> tag. The drop down list has following attributes :

Attribute	Values
NAME	Specifies the name of the drop down list (considered as a variable name in which the selected value)
SIZE	Specifies the no. of choice that are display in drop down list.
MULTIPLE	Specifies whether to allow Multiple selections or not.

Each option of the list is mentioned with the between <OPTION> and </OPTION> tags. Then whatever choice is selected, its value is stored in the name variable which is defined in the <SELECT> tag. For example,

```
<SELECT NAME=”sport” SIZE=3>
        <OPTION SELECTED> Cricket </O
        <OPTION> Rugby </OPTION>
        <OPTION> WWF </OPTION>
        <OPTION> Snooker </OPTION>
        <OPTION> Football </OPTION>
        <OPTION> Tennis </OPTION>
        <OPTION> Swimming </OPTION>
</SELECT>
```

In the above example if cricket is selected SPORT variable will have a value Cricket. By default the choice value is stored in the name variable of <SELECT> but if we want to store different values depending on the selection. It is possible with VALUE attribute of

<OPTION> tag. If we use SELECTED attribute with <OPTION>, by default that option is selected.

```
<SELECT NAME=”sport” SIZE=3>
    <OPTION VALUE=”1” SELECTED> Cricket </OPTION>
    <OPTION VALUE=”2”> Rugby </OPTION>
    <OPTION VALUE=”3”> WWF </OPTION>
    <OPTION VALUE=”4”> Snooker </OPTION>
    <OPTION VALUE=”5”> Football </OPTION>
    <OPTION VALUE=”6”> Tennis </OPTION>
    <OPTION VALUE=”7”> Swimming </OPTION>
</SELECT>
```

In the above example if we select CRICKET, now the value 1 will be stored in variable SPORT or according to the values specified. If value is not specified, default value is stored.

■ **Check Boxes**

Check boxes are another form of user interface to get input. These are the type of objects which has only two answers i.e. either "YES" or "NO". Select (marked) check box indicates that the choice is selected (YES) and unmarked check box indicates that the choice is not selected (NO). As each value of check box has to be checked differently, each check box has different variables in which value is stored.

The <INPUT> tag is used for that. <INPUT> tag is multipurpose tag, which is used to create different types of form objects depending on the TYPE specified.

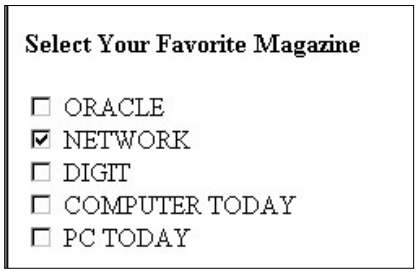
For example,

```
<INPUT TYPE="checkbox" NAME="hockey"> Hockey
```

The above example indicates that one check box will be created with variable name as hockey and the text also will be displayed as hockey and selected choice YES/NO will be stored in hockey variable. The CHECKED attribute is used if we want to select the check box in advance means if we want to keep any default choice, we can use this option.

Consider following example,

```
<B><P>Select Your Favorite Magazine </P> </B>

<P> <INPUT TYPE="checkbox" NAME="oracle"> ORACLE

<BR><INPUT TYPE="checkbox" NAME="network" CHECKED> NETWORK

<BR><INPUT TYPE="checkbox" NAME="digit"> DIGIT

<BR><INPUT TYPE="checkbox" NAME="ctoday"> COMPUTER TODAY

<BR><INPUT TYPE="checkbox" NAME="pctoday"> PC TODAY
```



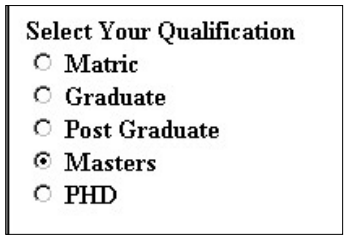
■ **Radio Buttons**

Radio buttons are another type of Form Object which are used to select any single choice from the user. The main thing about radio button is that you can select any one choice out of the given set of choices.

<INPUT> tag is used to specify the radio button with specifying "radio" type. As we have to select any single choice out of the given choices, we have to keep a single variable to accept the details from user because the single selected value will be stored in variable.

For example,

```
<P> <B> Select Your Qualification </B>

<BR> <INPUT TYPE="radio" NAME="qualification" VALUE="matric"> Matric

<BR> <INPUT TYPE="radio" NAME="qualification" VALUE="graduate"> Graduate

<BR> <INPUT TYPE="radio" NAME="qualification" VALUE="pg"> Post Graduate

<BR> <INPUT TYPE="radio" NAME="qualification" VALUE="masters" checked> Masters

<BR> <INPUT TYPE="radio" NAME="qualification" VALUE="phd"> PHD
```



■ **Text Box**

In all above explain form components are used to only select the defined item. No element is used to input the value by user. The text field provides a single-line field in which the user can type his text input and this text input is sent to the server. A text field is created using <INPUT> tag with TYPE=”text” attribute. It also has other attribute which are explained as below.

Attribute	Values
Name	Specifies the name of the text box (considered as a variable name in which the selected value)
VALUE	Specifies the initial (Default) value of the text box.
SIZE	Specifies the Maximum no. of character that should be allowed in the text box.
TYPE	Here the type is TEXT which specifies the Text Box.

For example,

```
<P> <B> Enter Your Name : <INPUT NAME=”user” TYPE=”text”
SIZE=20 VALUE=”nilesh”> </B> </P>
```

Output,

The above example creates one text box where variable name is USER, type is TEXT, it will allow maximum 20 characters and the default value is "NILESH".

- **Password**

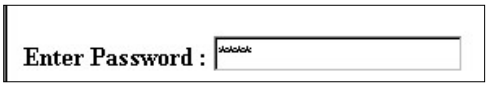
Password box has same syntax and attributes as Text Box control. The only difference is its type is specified as TYPE="password" with all same attributes of Text Box. The given information comes in a Password format which is replaced by ** in browser. But the value is stored in variable under specified variable name.**

For example,

<P> <B> Enter Password : <INPUT NAME="passwd" TYPE="password"

SIZE=20 VALUE="abcd"> </B> </P>

Output,



The above example creates one password field where variable name is PASSWD, type is PASSWORD, it will allow maximum 20 characters and the default value is "abcd".

■ Text Area

The text field can be used to get only a single-line field as an input. The text area is a multi-line area in which the user can type the input. For example, a text area can be used to obtain the experience of a user with a company, the suggestion of a user on a new product, address of customer, etc. can be accepted using Text Area is it allows multiple lines to input the data as a single unit under a common variable name.

A text area can be created by using the <TEXTAREA> and </TEXTAREA> tag. The text written between these tags comes as a default text in the text area. The <TEXTAREA> tag has three attributes. They are :

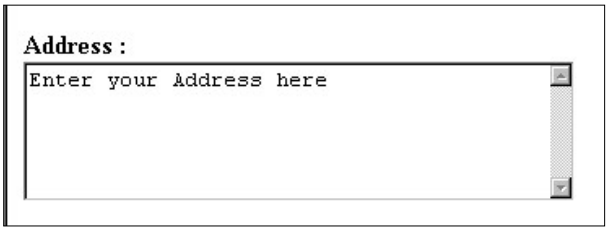
Attribute	Values
NAME	Specifies the name of the text area (considered as a variable name in which the selected value)
ROWS	Required no. of rows in text area.
COLS	Required no. of columns in text area.

For example,

<B> Address </B> <TEXTAREA NAME=”address” ROWS=5 COLS=40>

Enter Your Address Here </TEXTAREA>

Output,



The above text area has 5 ROWS and 40 COLS which are visible, even you can enter more text in this. But this indicates the no. of rows and columns which are shown in the screen. The inserted value will be stored in a variable called “ADDRESS”.

■ SUBMIT and RESET Buttons

These are two most important buttons. “SUBMIT” button is used to transfer the details from client to server and vice versa whereas “RESET” button is used to refresh all the user inputs, which resets all the values again to their default or blank whatever is applicable.

The SUBMIT button is used at the end of the user input. After the user has typed all the entries in the form, he can click the submit button. Only when the user clicks the submit button is the data given by the user sent to the server.

The SUBMIT button is created as follows :

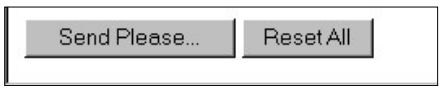
```
<INPUT TYPE=”submit” VALUE=”Send Please…”>
```

The RESET button is used to reset the values of all the values in the form of the user. After typing some entries in the form, if the user wants to change them, he can do so by clicking the reset button.

The RESET button is created as follows :

```
<INPUT TYPE=”reset” VALUE=”Reset All”>
```

Output,



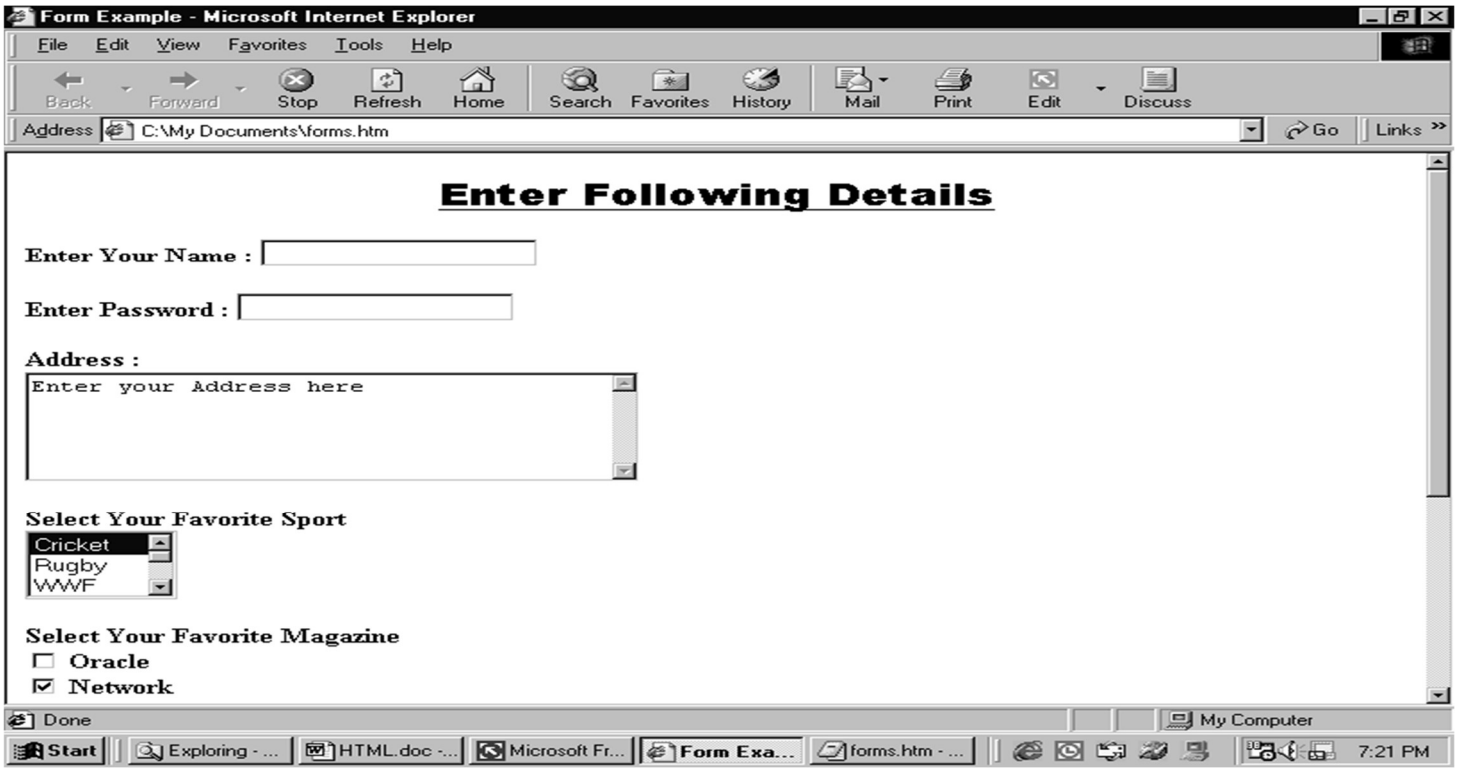
Consider following example of form, which includes all form objects :

```
<HTML>

<HEAD>

<TITLE> Form Example </TITLE>

</HEAD>

<BODY>

<P align="center"> <FONT face="arial black" size=5> <U> Enter Following Details </FONT> </P> </U>



<P>

<FORM ACTION="/cgi-bin/nilesh/check.cgi" METHOD="post">

<B> Enter Your Name : </B> <INPUT NAME="name" TYPE="text">

<P>

<B> Enter Password : </B> <INPUT NAME="password" TYPE="password">

<P>

<B> Address : </B><BR><textarea rows="5" name="S1" cols="40">Enter your Address here

</textarea>
```

```
<B><P> Select Your Favorite Sport <BR>
<SELECT NAME="sport" SIZE=3>
    <OPTION SELECTED> Cricket </OPTION>
    <OPTION> Rugby </OPTION>
    <OPTION> WWF </OPTION>
    <OPTION> Snooker </OPTION>
    <OPTION> Football </OPTION>
    <OPTION> Tennis </OPTION>
    <OPTION> Swimming </OPTION>
</SELECT>

<B><P>Select Your Favorite Magazine </B>
<BR><INPUT TYPE="checkbox" NAME="oracle"> Oracle
<BR><INPUT TYPE="checkbox" NAME="network" CHECKED>
Network
<BR><INPUT TYPE="checkbox" NAME="digit"> Digit
<BR><INPUT TYPE="checkbox" NAME="ctoday"> Computer Today
```

```
<BR><INPUT TYPE="checkbox" NAME="pctoday"> PC Today


<P> <B> Select Your Qualification </B>

<BR> <INPUT TYPE="radio" NAME="qualification" VALUE="matric">
Matric

<BR> <INPUT TYPE="radio" NAME="qualification"
VALUE="graduate"> Graduate

<BR> <INPUT TYPE="radio" NAME="qualification" VALUE="pg">
Post Graduate

<BR> <INPUT TYPE="radio" NAME="qualification"
VALUE="masters" checked> Masters

<BR> <INPUT TYPE="radio" NAME="qualification" VALUE="phd">
PHD

<P>

<INPUT TYPE="SUBMIT" value="Send Please...">

<INPUT TYPE="reset" VALUE="Reset All">

</FORM>

</BODY></HTML>
```

Output in Browser:

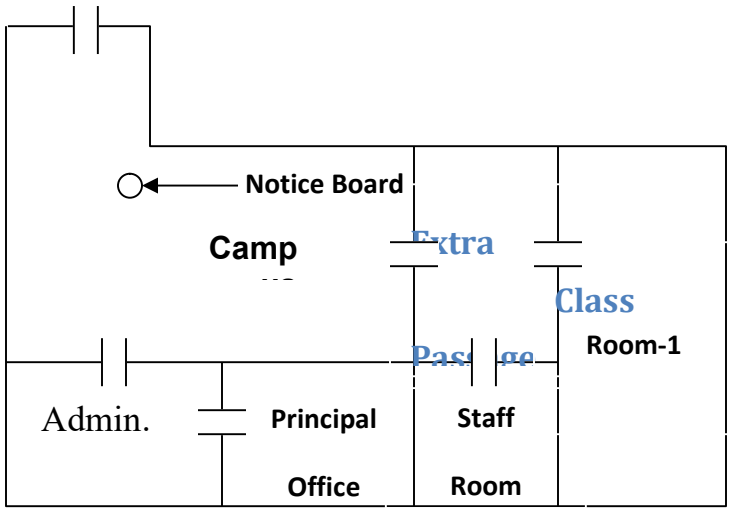

Form Example - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Address C:\My Documents\forms.htm
Go Links
Enter Following Details
Enter Your Name :
Enter Password :
Address :
Enter your Address here
Select Your Favorite Sport
Cricket
Rugby
WWF
Select Your Favorite Magazine
Oracle
Network
Done
My Computer
Start
Exploring - ...
HTML.doc -...
Microsoft Fr...
Form Exa...
forms.htm - ...
7:21 PM


Select Your Favorite Magazine
Oracle
Network
Digit
Computer Today
PC Today
Select Your Qualification
Matric
Graduate
Post Graduate
Masters
PHD
Send Please...
Reset All
Done
My Computer
Start
Exploring - ...
HTML.doc -...
Microsoft Fr...
Form Exa...
forms.htm - ...
7:22 PM

➔ Image Map

We have already studied that we can put a hyperlink on an image also in HTML document. Here we are going to learn how several hot spots (parts of pictures) can be made which deals with that location is loaded and opened. When we click at another location, the URL which deals with that location is loaded and opened.

For example, suppose a web page is designed to give the details of your college campus. The plan of the college campus can be shown as following figure.



The Map of Ground Floor of H.N.S College

In the home page, this picture can be shown using the <IMG> tag. In this picture, when we click on the Admin. Section, a new HTML document must be opened which gives details of the Administration Section or if we click on Staff Room, it should open a new HTML document which should give the details of Staff Members etc. Similarly, every part of this picture is to be made a hot spot to bring in a different HTML document. This is called as an Image Map. So we can say that , "When an image is divided into hot spots (parts), it is called as an Image Map".

An image map is an image which contains several hot spots that may result in loading different URLs depending upon the location where the user clicks. Image maps are of two types :

- **Server side image map**
- **Client side image map**

■ **Server Side Image Map**

The internet and web browsing is done in the client-server model. The HTML document is supplied by the web server and the user is in a node. The browser is the client's software. It gets the HTML document from the server and shows it to the user.

When there is an image map, the browser loads the supporting URL by either of the following two methods.

- **From the server in the case of a server side image map.**
- **The supporting of the original document in the case of a client side image map.**

In the case of a server side map, when the user clicks on a hot spot of an image map, the x and y coordinates of the hot spot are examined by the client and a request is made to the server to send the corresponding URL.

The server responds by sending the required URL. Hen the client (browser) shows the page to the user. Since the messages are to be sent to the server and the required information got, the server side map is slow.

The server side image map is specified by the anchor tag <A> with the HREF attribute to the map file. In order to specify that this is a server side image map, the ISMAP attribute is specified in the <IMG> tag. A simple example of image map is shown as follows :

```
<HTML>
    <HEAD>
        <TITLE> Image Map Example </TITLE>
   </HEAD>

   <BODY>
        <P ALIGN=”center”>
        <A HREF=”elephant.map”>
            <IMAGE SRC=”d:\animals\elephant.gif” ISMAP
WIDTH=400>
        </A> </P>
   </BODY>
</HTML>
```

In order to mention that an image map is a client side one, we must give the use map attribute to the <IMG> tag. We must give the <MAP> and <AREA> tags to define the hot spots in the image. The <MAP> tag has a name attribute. This name attribute is referenced within the <IMG> tag as the USEMAP attribute. For example, consider the following example,

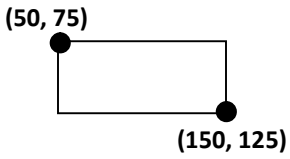
```
<IMG SRC=”d:\college.gif” USEMAP=#campus WIDTH=400>
    <MAP NAME=campus>
        .
        .
    </MAP>
```

In the above example “college.gif” is the image shown. The attribute USEMAP = #campus says that this is a client side image map and its mapping name is a “CAMPUS”. This name is specified by the <MAP tag as <MAP NAME=campus> and referenced in the <IMG> tag is USEMAP = #campus. In between the <IMG> and </MAP> tags, we have to specify the hot spots and their corresponding URLs using the <AREA> tag. In the <AREA> tag we must mention the shape of the hot spot, the coordinates to represent the HREF value. The shape of the hot spot may be any of the following :

- Rectangle (rect)
- Circle
- Ploygon (poly)

- **RECTANGLE**

The rectangle is specified by the word RECT and the left top and right bottom (x, y) coordinates. For example, suppose a rectangle has the left top and right bottom coordinates as shown in Figure.



Suppose this rectangular area is to be defined as a hot spot to link HREF = "admin.htm" The <AREA> tag is given as follows :

<AREA SHAPE = "rect" COORDS = 50, 75, 150, 125 HREF = "admin.htm">

- **CIRCLE**

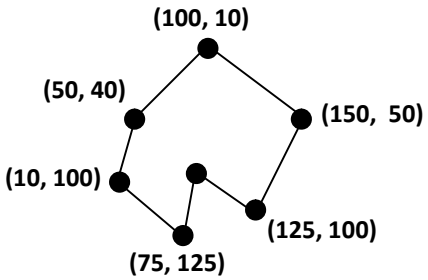
A circle is specified by the coordinates of the center and the radius. For example, consider the following.

<AREA SHAPE = "circle" COORDS = 120, 75, 100) HREF = "xyz.htm">

This defines a circle with center (125, 75) with a radius of 100 as a hot spot havin a link to the HTML document "xyz.htm".

- POLYGON

A polygon is referenced by the x and y coordinates of its corner points. For example, suppose we want to define a polynomial shown in following figure :



The polynomial is represented as a hot spot linking to "ground.html" as follows :

<AREA SHPCE = poly COORDS = 100, 10, 150, 50, 125, 100, 110, 75, 75, 125, 10, 100, 50, 40, 100, 10 HREF = "ground.html">

■ Example of Image Map

Consider following example of Image Map where links of three parts are given. Following is a map of our college for Ground Floor. In this map link on three parts is specified which are Notice Board, Campus and Admin. Section. All these three respectively are made by following symbols :

- Notice Board ➔ Circle
- Campus ➔ Polygon
- Admin. Section ➔ Rectangle

Consider following HTML code and output in browser :

```
<HTML>
    <HEAD>
        <TITLE> Image Map Example </TITLE>
    </HEAD>
    <BODY>
        <CENTER> <FONT FACE=”arial black” SIZE=5>
            <P ALIGN = “Shree H.N. Shukla College”>
            <P ALIGN = “Of Information Technology”>
        </CENTER> </FONT>
        <FONT FACE = “times new roman” SIZE = 5>
        <P ALIGN=”center”> <U> College Map </P> </U>
        </FONT>
```

```
<IMG SRC = “collegemap.bmp” USEMAP = “#hnscampus” WIDTH=400>
<MAP NAME=”hnscampus”>
    <AREA SHAPE=”circle” COORDS = 50, 150, 100 HREF = “notice.htm”>
    <AREA SHAPE=”poly” COORDS = 0, 0, 0, 100, 100, 175, 175, 400, 400, 600, 600,
                                0, 0, 0 HREF = “campus.htm”>
    <AREA SHAPE=”rect” COORDS = 250, 300, 400, 600 HREF = “admin.htm>
</MAP>

<BR>
# <A HREF = “guest.htm”> Guest Book </A>
  # <A HREF = “mailto:hnsmail@hns.com”> Contact Us </A>
  # <A HREF = “nilesh.htm”> Develped by : NILESH ADVANI </A>

</BODY>
</HTML>
```

Output in Browser :