UNIT 1

Hyper Text Markup Language

HTML Common tags- List, Tables, Images, Forms, Frames; Cascading Style Sheets.

1.1 Introduction

Hyper Text Markup Language (HTML) is the prominent/popular markup language for creating web pages and other information that can be displayed in a web browser. HTML is a set of markup symbols or code inserted in a file intended for display on a World Wide Web browser page. An HTML file is a text file containing small markup tags. The markup tags tell the Web browser how to display a Web page's words and images for the user. Any formatted text document is composed of a set of elements such as paragraphs, headings, and lists. Each element has to be surrounded by angular brackets. In html, formatting is specified by using tags.

HTML is a formal recommendation by the World Wide Web Consortium (W3C) and is generally adhered to the major browsers, for example, Microsoft's Internet Explorer and Netscape's Navigator, which also provide some additional non-standard codes. The current version of HTML is HTML 5.0. However, both Internet Explorer and Netscape implements some features differently and provide non-standard extensions. Web developers using the more advanced features of HTML may have to design pages for both browsers and send out the appropriate version to a user. Significant features in HTML are sometimes described in general as dynamic HTML. HTML5 is an extensible form of HTML which is also called Extensible Hyper text Markup Language (XHTML).

HTML was invented in 1990 by a scientist named Tim Berners-Lee. The purpose was to make it easier for scientists at different universities to gain access to each other's research documents. The project became a bigger success than Tim Berners-Lee had ever imagined. By inventing HTML he laid the foundation for the web as we know it today. HTML is a language, which makes it possible to present information (e.g., scientific research) on the Internet. What you see when you view a page on the Internet is your browser's interpretation of HTML. To see the HTML code of a page on the

Internet, simply click "View" in the top menu of your browser and choose "Source".

Basically an HTML document is a plain text file that contains text and nothing else.

When a browser opens an HTML file, the browser will look for HTML codes in the text and use them to change the layout, insert images, or create links to other pages.

Since HTML documents are just text files they can be written in even the simplest text editor.

Some of the most popular HTML editors, such as FrontPage or Dreamweaver will let you create pages more or less as you write documents in Word or whatever text editor you're using. It is possible to create WebPages without knowing anything about the HTML source behind the page. There are excellent editors in the market that will take care of the HTML parts. All you need to do is design layout of the page. However, if you want to make it above average in web design better, it is strongly recommended that you understand these tags. You can write your HTML by hand with almost any available text editor, including notepad that comes as a standard program with Windows. All you need to do is type in the code, then save the document, making sure to put **.html** extension or **.htm** extension to the file (for instance "mypage.html").

Type in the following text in "NOTEPAD":

```
<html>
<head>
<title>Title of page</title>
</head>
<body>
Hello World! This is my first homepage. <b>This text is bold</b>
</body>
</html>
```

Save the file as "first.html".

Start your Internet browser. Select "Open" (or "Open Page") in the File menu of your browser. A dialog box will appear. Select "Browse" (or "Choose File") and locate the HTML file you just created - "first.html" - select it and click "Open". Now you should see an address in the dialog box, for example "C:\MyDocuments\webdesign\first.html". Click OK, and the browser will display the page.

Explanation of above Example

The first tag in your HTML document is <html>. This tag tells your browser that this is the start of an HTML document. The last tag in your document is </html>. This tag tells your browser that this is the end of the HTML document. The text between the <head> tag and the </head> tag is header information. Header information is not displayed in the browser window. The text between the <title> tags is the title of your document. The title is displayed in your browser's caption. The text between the <body> tags is the text that will be displayed in your browser. The text between the and tags will be displayed in a bold font.

1.1.1 HTML Features

Simplicity

The first version of HTML was designed to be extremely simple, both to author and to write browsers for. This has played a major role in the incredibly rapid growth of the World Wide Web. HTML 3.0 provides a clean superset of HTML 2.0, adding high value features such as tables, text flow around figures and math, while still remaining a simple document format. The pressures to adopt the complexities of traditional SGML applications have been resisted, for example the Department of Defense's CALS table model or the ISO 12083 math DTD.

Scalability

As time goes by, people's expectations change, and more will be demanded of HTML. One manifestation of this is the pressure to add yet more tags. HTML 3.0 introduces a means for sub classing elements in an open-ended way. This can be used to distinguish the role of a paragraph element as being a couplet in a stanza, or a mathematical term as being a tensor. This ability to make fresh distinctions can be exploited to impart distinct rendering styles or to support richer search mechanisms, without further complicating the HTML document format itself. Scalability is also achieved via URI based links for embedding information in other formats. Initially limited to a few image formats, inline support is expected to rapidly evolve to cover drawing formats, video, distributed virtual reality and a general means for embedding other applications.

Platform Independence

HTML is designed to allow rendering on a very wide range of devices, from clunky teletypes, to terminals, DOS, Windows, Macs and high end Workstations, as well as non-visual media such as speech and Braille. In this, it allows users to exploit the legacy of older equipment as well as the latest and best of new machines. HTML 3.0 provides improved support for non-graphical clients, allowing for rich markup in place of the figures shown on

graphical clients. HTML can be rendered on a wide variety of screen sizes, using a scrolling or paged model. The fonts and presentation can be adjusted to suit the resources available in the host machine and the user's preferences.

Content not Presentation Markup

Information providers are used to tight control over the final appearance of documents. The need for platform independence weighs against this, but there is still a strong pressure to find appropriate means for information providers to express their intentions. The experience with proprietary document formats has shown the dangers of mixing presentation markup with content (or structural) markup. It becomes difficult to apply different presentation styles. It becomes painful to incorporate material from different sources (with different presentation styles). It becomes difficult to be truly platform independent. As a result, HTML 3.0 is designed for use with linked style information that defines the intended presentation style for each element. Style sheets can be expressed in a platform independent fashion or used to provide more detailed control for particular classes of clients or output media.

Support for Cascaded Style Sheets

For the Web, it is important to allow for a cascading of style preferences. The client has certain built-in preferences; the publisher may require a particular house style, e.g., for brand distinction; the author may feel the need to override the house style for special cases; the end-user may feel strongly about certain things, e.g. large fonts for easier visibility or avoiding certain colors due to an inability to distinguish between them. HTML 3.0 supports style sheets via the use of the LINK element to reference a style sheet with a URI. Authors can place overrides in separate style sheets or include them in the document head within the STYLE element. The effectiveness of caching mechanisms for speeding up the retrieval of style sheets is enhanced by the separation of style information into generic commonly used style sheets, and overrides specific to this document.

Support for Non-Visual Media

HTML 3.0 is designed to cater for the needs of the visually impaired. Markup for inline figures includes support for rich descriptions, along with hypertext links that double up as defining geometric hot zones for graphical browsers, simplifying the author's job in catering for the different groups of users. Table markup includes provision for abbreviated row and column names for each cell, which are essential for conversion to speech or Braille. Math markup treats formulae and equations as hierarchies of expressions. This allows disambiguating pauses to be inserted in appropriate places during conversion to speech.

1.2 Basic Concepts

The software tools you need

HTML is written in plain text. That means you do not need any fancy software programs like a word processor to create your HTML files. All you need is a simple text-editor that is already on your system. For Windows, **Notepad** and for MACs, that would be **SimpleText**.

Some important rules

As with most things in life, there are rules. In HTML, the rules are fairly simple. For starters, HTML tags are always surrounded by what are called angular brackets < and >. You will find these brackets on your keyboard just above the comma and period.

Elements

The words/letters between these two angular brackets are called **elements**. These are the pre-defined commands within HTML. Elements tell the browser how to display the web page. For example: <hr> tells the browser to display a horizontal rule;
 tells the browser to skip a line.

Container and empty tags

There are two kinds of tags: container and empty.

The **container tag** always wraps around text or graphics and comes in a set with an opening and a closing tag:

<html> opening tag
</html> closing tag

Notice the forward slash (/) on the closing tag. This tells the browser that the tag has ended. On the other hand, the **empty tag** stands alone. The tag **
** is one that adds a line break. Empty tags do not have to be wrapped around copy and do not require a closing.

Case sensitive

HTML is **not** case sensitive. That means, you can use either lowercase or uppercase characters. <HTML> is the same as <html>. For consistency, it is suggested to maintain your scripts in the same format either uppercase or lowercase.

1.2.1 Structure of HTML Program

All HTML documents are divided into two main parts: the head and the body. It goes something like this:

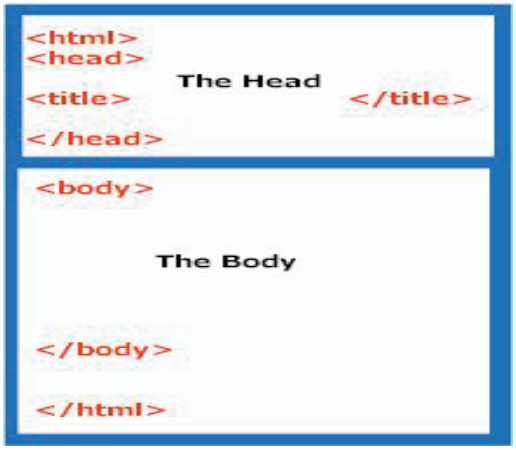


Fig. 1.1: Structure of HTML Program

We must have the <html>, <head> and <body> container tags in every HTML file.

The **<html>** tag tells the browser that this is an HTML document. You must begin and end your files with this tag. The **<head>** tag contains general information like the title of your document. The **<body>** tag holds all your content: words, pictures, artwork and other information.

Nesting

Embedding a tag inside an existing tag is called nesting. Notice how the tag <title> is nested inside the <head> tag, while <head> and <body> are nested inside <html>.

To save a file as HTML file, open the text you wish to edit or type it into Notepad, and choose "File...Save As..." from the menu bar. Under "File name" give your file a name and change its extension to ".html" Under "Save as type" switch to "All Files" then click "OK". Notepad will save your file as ASCII text, and the ".html" or ".htm" extension will allow your browser to recognize it as an HTML file. (Hint - When naming an HTML file, it is a good idea to use a name without any spaces or uppercase letters. So, if you wanted to name a file Test page, some options would be testpage.html or test_page.html)

1.3 HTML Common Tags

To build any web page you will need four primary tags: <html>, <head>, <title> and <body>. These are all container tags and **must appear as pairs with a beginning and an ending.**

<html>…</html>

Every HTML document begins and ends with the <html> tag. This tells the browser that the following document is an html file. Remember, tags tell the browsers how to display information.

<head>…</head>

The <head> tag contains the title of the document along with general information about the file, like the author, copyright, keywords and/or a description of what appears on the page.

<title>…</title>

Appears within the <head> tag and gives the title of the page. Try to make your titles descriptive, but not more than 20 words in length. The title appears at the very top of the browser page on the title bar.

<body>…</body>

The main content of your page is placed within the body tags: your text, images, links, tables and so on. Common tags of an HTML are listed in Table 1.1.

Table 1.1 Common HTML tags

Element: start/end tags	Description
<html></html>	Starts and ends a HTML document.
<title></title>	Text that appears in the title bar.
<head></head>	Information about the document.
<body></body>	The main part of the document.
	A paragraph.
<hr>	A horizontal line.
 	A line break.
 	A link.
<img src=”url”>	An image.
<! - - Comments - - >	Comments that are not displayed in the browser.
	A section in the document.
	An inline section in a document.
	An unordered list.
	An ordered list.
	A list item.
<table></table>	Encloses a table.
<tr></tr>	A row in a table.
<td></td>	A cell within a row.
<pre></pre>	Enclosed text that stays in its raw format.

1.3.1 Creating a Web Page using HTML

Using the primary HTML tags, you are now ready to create your first Web page.

Step 1 Open up a text editor (SimpleText for Mac or Notepad for Windows)

Step 2 Enter the following:

```
<html>
<head>
<title> This is my first web page</title>
</head>
<body>
Hello world. Welcome to HTML Programming. There's more to come.
</body>
</html>
```

Step 3 Save the document as: **firstpage.html** Your file can be saved as either an filename.htm or filename.html file. Remember to save your document on the computer in a place that you can find it again.

Step 4 To preview your new document, open Mozilla Firefox. On the tool bar (located up near the top of the browser):

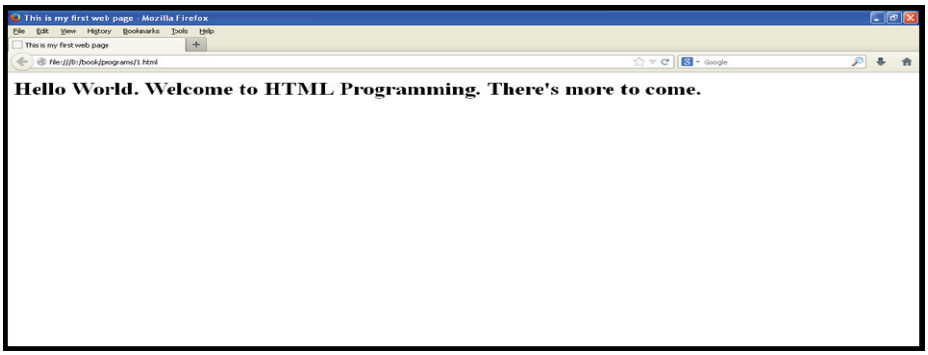
Select **File** menu.

Select **Open File**

A dialogue box appears. Go to the location where you saved your file and click on it. This will bring you back to the dialogue box, which should now be showing your file.

Click **Open**

Step 5 Your file should look like this in your **browser.**



1.3.2 Text Formatting Tags

After any length of time on the Internet, you will notice that a web page is made up of more than just plain words on a screen. There are headlines, paragraphs, graphics, colors and much more. Our next tags-headline, paragraph, line break and horizontal rule-will help us make our current page a lot more exciting. Let us learn how.

1.3.2.1 Headline tag

In HTML, bold copy is created by using the headline tag. There are six levels of headlines, ranging from **<h1>…</h1> to <h6>…</h6>**. Here is an example of the code for *all* the headline sizes:

```
<h1>Level 1 Headline</h1>
<h2>Level 2 Headline</h2>
<h3>Level 3 Headline</h3>
<h4>Level 4 Headline</h4>
<h5>Level 5 Headline</h5>
<h6>Level 6 Headline</h6>
```

Let us add a headline to our Web page document.

Step 1 Load your text editor and open your file: firstpage.html

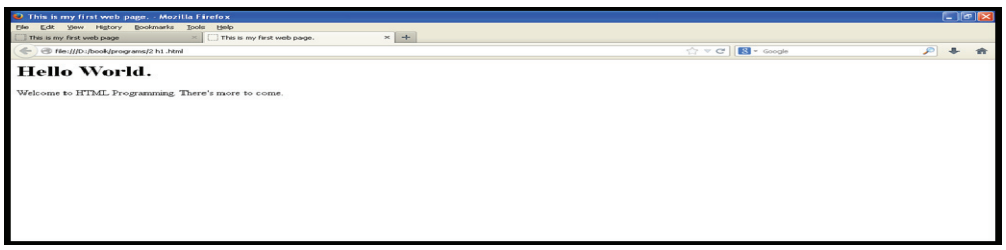
Step 2 Add the <h1> tag to the words "Hello world." as shown in bold.

```
<html>
<head>
<title>This is my first web page.</title>
</head>
<body>
<h1>Hello world.</h1> Welcome to HTML Programming. There's more to
come.
</body>
</html>
```

Step 3 Save the file

Step 4 Open up the file in Mozilla Firefox.

Your new page should look like this in your browser.



1.3.2.2 Paragraphs, line breaks and horizontal rule

To add space between paragraphs you use the paragraph tag:

** … **

This is a container tag and must have a beginning and an ending.

To add a single line of space, you use break tag: **
**

This is an empty tag and stands alone. You can use the <br> tag to insert one or more blank lines.

To create a horizontal line on your page you use the empty tag:

<hr>

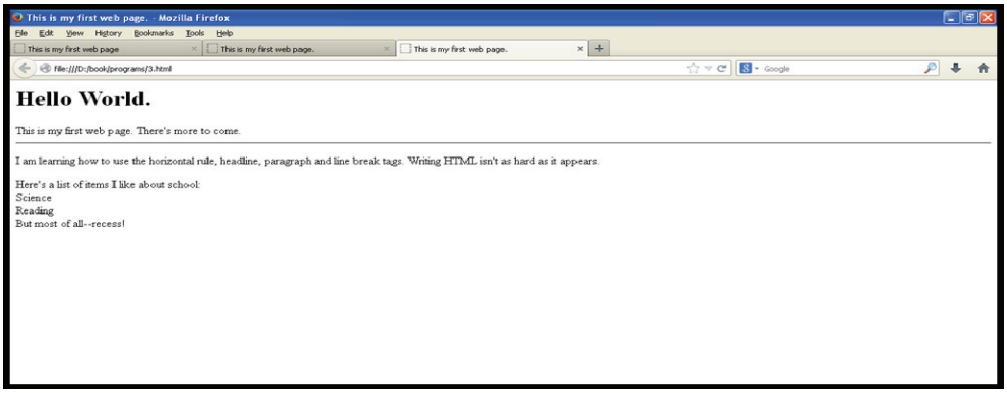
Step 1 Load your text editor.

Step 2 Open the file: **firstpage.html**.

Let us add some more text so that we can use the new tags that we have learned. Add tags and text that appear in bold.

```
<html>
<head>
<title>This is my first web page</title>
</head>
<body>
<h1>Hello world.</h1> This is my first web page. There's more to come.
<hr>
<p>
I am learning how to use the horizontal rule, headline, paragraph and line break tags. Writing HTML isn't as hard as it appears.
</p>
<p>Here's a list of items I like about school: <br>
Science<br>
Reading<br>
But most of all--recess!<br>
</p>
</body>
</html>
```

Your latest revision should look like this in your browser.

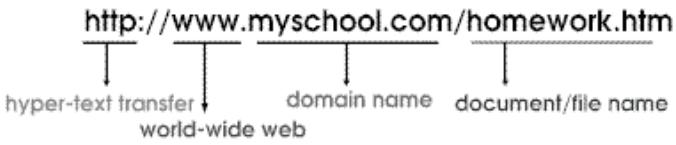


1.4 Hyper Links

Links are the power behind the World Wide Web. Through links, millions of pages, filled with information and knowledge, are only a click away. Every Web site can become a virtual library—a place to learn and grow! And it is all so very simple to accomplish.

Uniform Resource Locator

Every Web site has an address called the **Uniform Resource Locator** or URL. But what does it mean? Let us examine one more closely:



Hyper Text Link

There are two things you need to create a link:

1. the name of a file or the URL you want to link to
2. the link hotspot—the highlighted text or graphic that will be clicked on to get to where you want to go.

To create a link in HTML, you need the **anchor** tag:

<a>…

Inside the tag, you need the attribute: **HREF** (hypertext reference).

An example of a link looks like this:

Vikram uses the browser called Firefox.

The code looks like this:

Vikram uses the browser called

Firefox.

Anchor Tag Attributes:

HREF="..."

Creates a link to the specified document. Concatonate **#name** to the URL name if referencing an anchor within the page

NAME="..."

The name of the anchor, to be referenced. Creates an anchor that can be linked to.

STYLE="..." elements seperated by ";"

COLOR:color or **COLOR:RGB(#,#,#)**

Color of link

text-decoration:none - Eliminate underlining on the link

TARGET="window name"

Opens the URL in the window with the specified name. If there is no such window, a new one is opened.

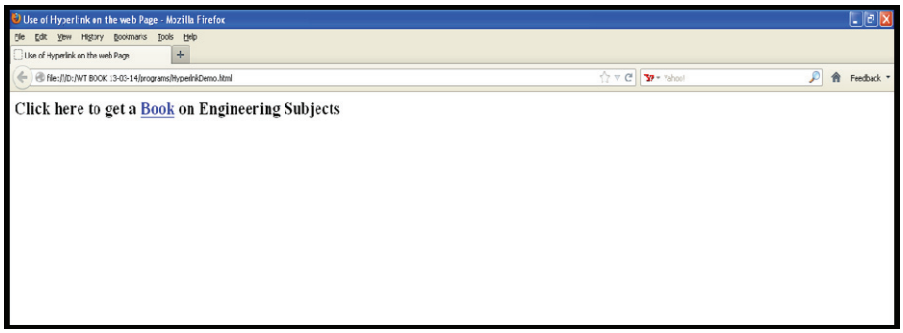
TITLE="..."

Information about the link. Eg: Displayed in Internet Explorer when mouse points to the link.

Program on hyperlinks - HyperlnkDemo.html

```
<html>
  <head>
        <title> Use of Hyperlink on the web Page </title>
  </head>
  <body>
    <h2>Click here to get a
          <a href="http://www.ebooks.com">Book</a>
                on Engineering Subjects</h2>
   </body>
</html>
```

Your output should look like this in your browser.



1.5 Images

From the start, images have always been memory and space hogs. But during the past few years, designers have worked hard to keep image files small.

The Image tag

To place an image onto a Web page you will need to use the image tag. It is an empty tag (remember, that means there is no closing tag, only a beginning).

<img attributes>

Image Tag Attributes:

ALT="..."

Text string displayed when images cannot be displayed.
ALIGN="LEFT" or **"RIGHT"** or **"CENTER"** or **"TOP"** or **"MIDDLE"** or **"BOTTOM"**

Specifies the alignment of the image relative to the text that follows.

BORDER=# - Border around image in pixels - 0 removes borders around link images.

HEIGHT = # - Height of the image in pixels.

HSPACE=# - Horizontal space around the image in pixels.

SRC="..." - The URL of the image. Relative reference allows browser to download the image once if it is referenced multiple times.

VSPACE=# - Vertical space around the image in pixels.

WIDTH = # - Width of the image in pixels.

Note: # represents integer value.

SRC attribute

To help the browser identify and find an image, you use the following command:

<img src="filename.gif">

While the name of the file is important, it is equally important to direct the browsers to where the file is located. Notice that all the files are in the same folder or what is also called a directory. So the browser will first look into the main folder (Mikes Bike Shop), find the HTML file (shop.html), load the text and then load the graphic files (horn.gif and bike.gif).

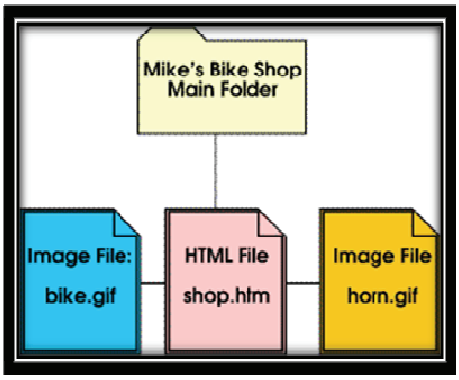


Image hotspots

Just like text links, image links can also be created. Just include the image tag within the anchor tag.

Here is the code for a link to the Blast-Off page:

<a href-"graphicsc10.html"><img src=atom.jpg width="119"

height="120" alt="Take this link to the Blast-Off page"

border="0">

And here is the image hotspot. Place your cursor over it and then click.

1.6 Lists

Lists come in a variety of three forms such as numbered, bulleted & definition lists. The numbered lists are called **ordered lists** and the bulleted lists are **unordered lists.** Lists are nested. There is a tag that identifies the type of list (eg. numbered or bulleted). Then within that tag there is another tag that itemizes the list. Maybe some definitions would help.

a. Ordered List: The ordered list is a container tag and is used for numbered lists.

Syntax: <ol type="1" | "a" | "I" | "i" start="n">…..

b. Unordered List: The unordered list is a container tag and is used for bulleted lists.

Syntax: <ul type="disc" | "square" | "circle" > …..

**
… -** The listed item tag is a container tag and is **nested** within the ordered or unordered tags.

c. Definition List: It is used to create the lists with definition term and definition description.

<dl>….. </dl>

This tag is used for the third category i.e., definition list, where numbers or bullets are not used in front of the list item. Instead, it uses definition for the items.

<dt>…..</dt>

This is a sub tag of the <dl> tag called as definition term, which is used for marking the items whose definition is provided in the next data definition.

<dd> ….</dd>

This is a sub tag of the <dl> tag. Definition of the terms are enclosed within these tags.

The definition may include any text or block.

Program for an ordered (numbered) list goes like this: lists.html

```
<html>
<head>
<title>Demonstration of Various Types of Lists</title>
</head>
<body>
<h1>Ordered List</h1>
```

```
<ol type="I">
<li>My first item on the list.</li>
<li>My second item on the list.</li>
<li>My third item on the list.</li>
<li>My fourth item on the list.</li>
</ol>
<h1>Unordered List</h1>
<ul type="square">
<li>My first item on the unordered list.</li>
<li>My second item on the unordered list.</li>
<li>My third item on the unordered list.</li>
<li>My fourth item on the unordered list.</li>
</ul>
<h1>Definition List</h1>
<dl>
<dt>HTML</dt>
<dd>Hyper Text Markup Language is used to create web pages.</dd>
<dt>JSP</dt>
<dd>Java Server Pages are used to create dynamic web pages.</dd>
</dl>
</body>
</html>
```

In the browser it will appear like this in browser:

1.7 Applying Attributes to the Tags

Bold headlines and organized lists are nice but there has got to be more to a web page than that. Attributes are just special codes placed within the HTML tags that **describe how the tags will look**.

Color Attribute: For instance, let us say you want to have a green background on your Web page with red text. You would type this code:

<body bgcolor="#bee3c2" text="#ff0000">

Hello. This is an example for Body Tag Attributes.</body>

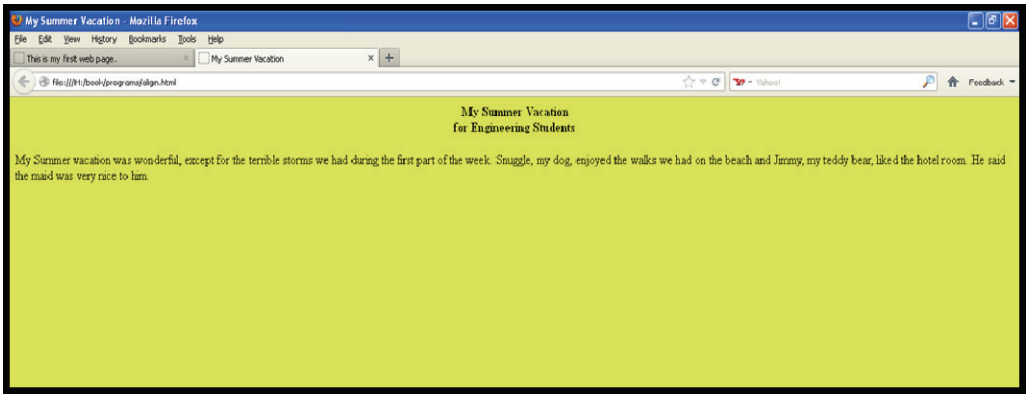
Notice the attributes: **bgcolor and text**. They are placed within the <body> tag. Attributes never stand alone. Instead, they always appear inside a body tag. Colors are described in hexadecimal code and color names. Hex code is six digits or letters that represent a color. Also, hex codes always start with a # sign.

Align Attribute: Another attribute that comes in handy is: **align**. Often used with headlines and graphics, this attribute let us you place text or pictures to the left, center or right side of the margin.

Here is a simple example for the align attribute: attributes.html

```
<html>
<head>
<title>My Summer Vacation</title>
</head>
<body bgcolor="#e3f04a" text="#000000">
<h4 align="center">My Summer Vacation<br> for Engineering
Students</h4>
<p>My Summer vacation was wonderful, except for the terrible storms we
had during the first part of the week. Snuggle, my dog, enjoyed the walks
we had on the beach and Jimmy, my teddy bear, liked the hotel room. He
said the maid was very nice to him.</p>
</body>
</html>
```

Here is the resulting page in a browser:



You can use the align attribute with all sorts of tags like: headlines, paragraphs, rules, graphics and so on.

Let us apply what we have learned.

Step 1 Load your text editor and open your HTML document we have been working on: **firstpage.html**

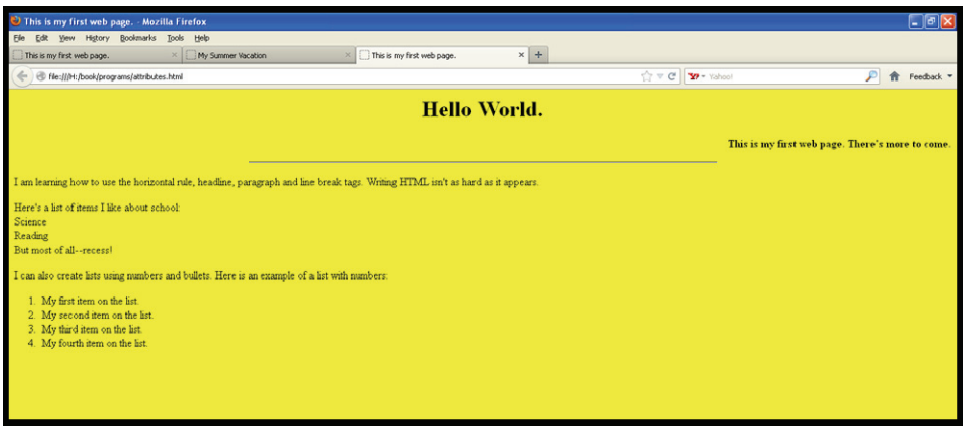
Step 2 Let us add some attribute codes to your original HTML file. Enter the code listed below in red:

```
<html>
<head>
<title>This is my first web page.</title>
</head><body bgcolor="#ffff00" text="#000000">
<h1 align="center">Hello World.</h1>
<p align="right"><b>This is my first web page. There's more to
come.</b>
<hr align="center" width="50%">
<p>
I am learning how to use the horizontal rule, headline, paragraph and line
break tags. Writing HTML isn't as hard as it appears.
</p>
<p>Here's a list of items I like about school:<br>
Science<br>
Reading<br>
But most of all--recess!<br>
<p>I can also create lists using numbers and bullets. Here is an example
of a list with numbers:
```

```
<ol>
<li>My first item on the list.</li>
<li>My second item on the list.</li>
<li>My third item on the list.</li>
<li>My fourth item on the list.</li>
</ol>
</p>
</body>
</html>
```

Step 3 Save as your file with attributes2.html

Step 4 Load the file in the browser resulting page should look like this in your **browser.**



Notice the new changes:

- Background color is now yellow, text is black.
- The heading is centered.
- First paragraph is aligned to the right
- Horizontal rule is now 50% of the margin and centered.

1.8 Advanced Text Formatting

In this section, we will cover the following tags and attributes:

- Font styles: **…**
- Bold: **…**
- Italic: **…**
- Indented text: **<blockquote>…</blockquote>**

- Smaller type: **<small>…</small>**
- Larger type: **<large>…</large>**
- Centered type: ** … **

Font tag

Not long ago, **Times Roman** was the only font you could use with HTML. It is still the font of choice (the default) for the leading browsers. Today, we can choose different font styles as well as several other font options by using the tag:

…

We will cover three attributes used with the font tag:

…
…
…

Font Size - Font size must be an integer values which specifies the size of the font.

Font Face- Include more than one type face. For instance, let us say you want the following text to appear in Arial type face. This is how you would do it.

……

Font Color - Use the attribute **color** within the font tag to set font color.

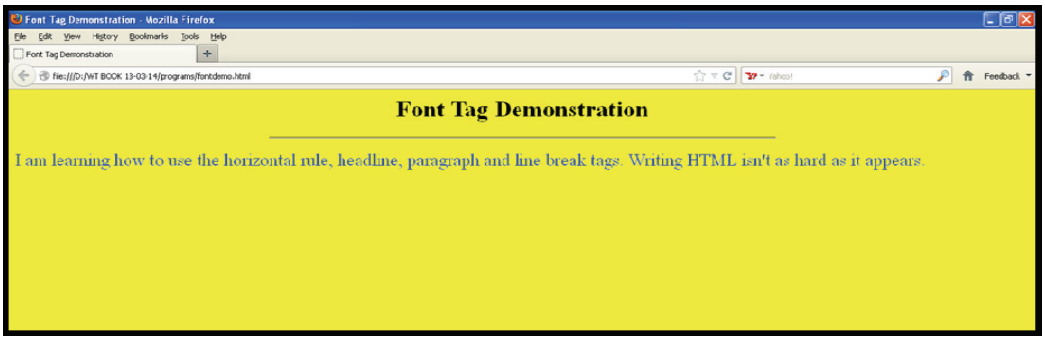
……

Let us apply what we have learned – fontdemo.html

```
<html>
<head>
<title> Font Tag Demonstration</title>
</head>

<body bgcolor="#ffff00" text="#000000">
<h1 align="center">Font Tag Demonstration</h1>
<hr align="center" width="50%">
<p><font size="5" face="arialblack" color="#0033ff">
I am learning how to use the horizontal rule, headline, paragraph and line
break tags. Writing HTML isn't as hard as it appears.</font>
</p>
</body>
</html>
```

Here is how it will appear in your browser.



Other text formatting tags

The other text formatting tags are fairly simple to use and understand.

Bold and Italic tags

To create bold text use: **…**

To create italicized text use: **…**

Blockquote

Sometimes you may want to indent a sentence or an entire paragraph. Then use the blockquote tag:

<blockquote>…</blockquote>

Small and big tags

To create a copy that is slightly smaller than normal: **<small>…</small>**

To create a copy that is slightly larger than normal: **<big>…</big>**

Center tag

HTML allows us to center text without using an attribute. So when you want to center your copy, it's simple. Use the following tag:

** … **

1.9 Tables

Tables have literally changed the look of the Web page. Originally, tables let people present data (mostly numbers) in a column format. Designers quickly figured out ways to improve the layout of their pages using tables. Although a bit complicated and sometimes difficult to understand, tables do improve your layout and let you present your material in a more eye-appealing way. The best way to understand tables is to look at the following illustration.

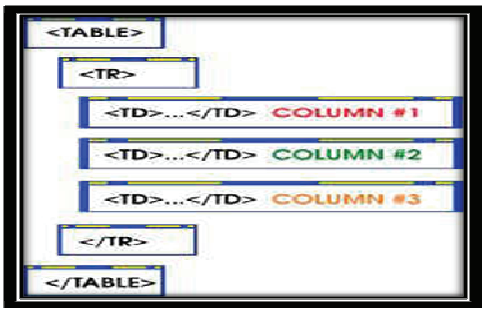


Fig. 1.2: Table Structure

The structure of a table

Every table is wrapped by the **<table>...</table>** tag. Then, the columns are surrounded by the **<tr>...</tr> row** tag. Table can have one column or as many as you want. In the above illustration, we are looking at three columns. Now each column has its own tag: **<td>...</td>**. Although these columns appear horizontally in code, they actually represent vertical columns.

Table Tag

To create a table you must used the following container tag:

<table>...</table>

Row Tag

Each table has at least one row and within that row are columns or what are also referred to as cells. The row tag looks like this:

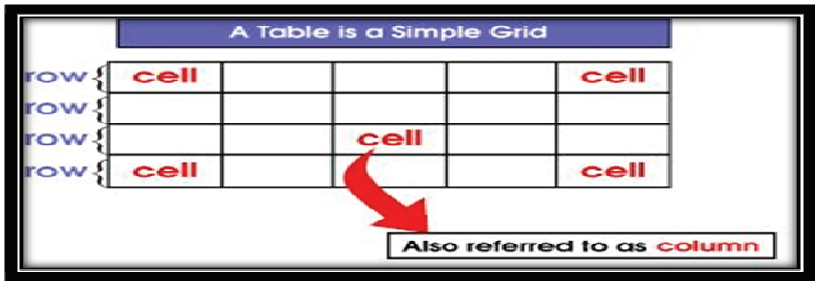
<tr>...</tr>

Column/Cell tag

And finally, to complete your table you must have the column or cell tag-this is where your information and images go. The column/cell tag looks like this:

<td>...</td>

Notice that I keep on using the words column and cell together.



Notice how this table is like a grid made up of rows and cells. The word "cell" is also referred to as a "column" and there lies some confusion. So when you work with tables, try to remember that the word cells and columns are often used to mean the same thing.

Table Tag Attributes

BORDER["# of pixels"] - The table is to have a border.

Optional: width in pixels (default=1). 0 => no border

CELLPADDING="# of Pixels" - Number of pixels to pad each cell (default=1)

CELLSPACING="# of pixels" - Number of pixels between cells (default=2)

NOWRAP - Text is not wrapped. May use **
** to break the line

SUMMARY= "..." - Describes the structure of the table (for handicapped use).

WIDTH="xx%" or "#" - Width of the screen to be used by the table. Either a % of width, or number of pixels.

Note: '#' is integer value.

Sub Tags:

1. **<CAPTION>...</CAPTION> -** Places a caption above the table. This tag appears between the <TABLE> and the first <TR> tags.

   Attribute: ALIGN=BOTTOM/TOP - Caption is on the bottom/top of the table.

2. **<TR>...</TR> -** Delimits a row of the table.

   Attributes: These may also be used with **<TH>** and **<TD>**

   ALIGN=LEFT or **RIGHT** or **CENTER**

   BGCOLOR="#XXXXXX" - Specifies the background color.

   VALIGN=TOP or **MIDDLE** or **BOTTOM**

3. **<TH>...</TH> -** Indicates a cell with a table heading. The characters are set off in some way (bold).
   - If the heading is to be at the top of the table, then the tags all go in the first row.
   - If they are to be on the left, the cells will be the first in each row.

4. **<TD>...</TD> -** table data, a regular cell within the table.

   Attributes: for <TD> and <TH>

COLSPAN=# - The number of columns this element spans.

NOWRAP - Text is not wrapped. May use **
** to break the line.

ROWSPAN=# - The number of rows this element spans.

WIDTH="xx%" or "#" - Width of the screen to be used by the column. Either a % of the table, or number of pixels.

Step 1 Open up a text editor (remember, Simple Text for Mac and Notepad for Windows).

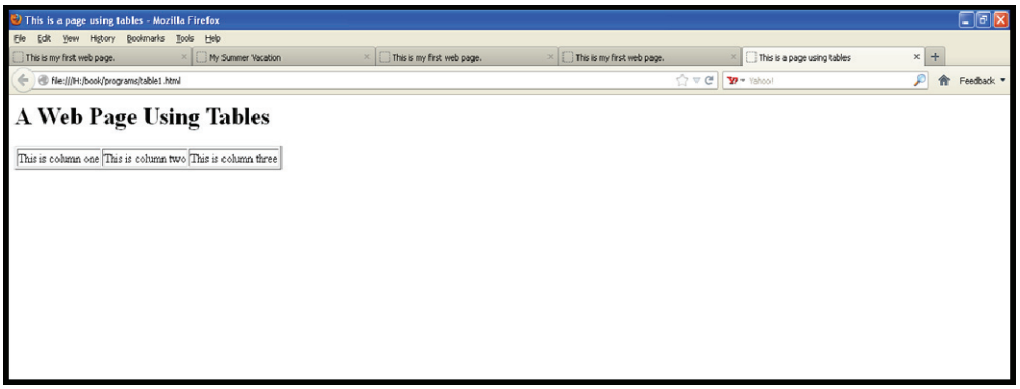
Step 2 Enter the following code:

```
<html>
<head>
<title> This is a page using tables </title>
</head>
<body bgcolor="ffffff" text="000000">
<h1>A Web Page Using Tables</h1>
<table border="1">
<tr>
<td>This is column one</td>
<td>This is column two</td>
<td>This is column three</td>
</tr>
</table>
</body>
</html>
```

Step 3 Save the file as **table1.html.**

Step 4 Opening Mozilla Firefox.

Step 5 View the page in the **browser**.



Now let us add some copy to our columns. And see what happens. Add the code listed in bold.

Step 1 Open the file table1.html

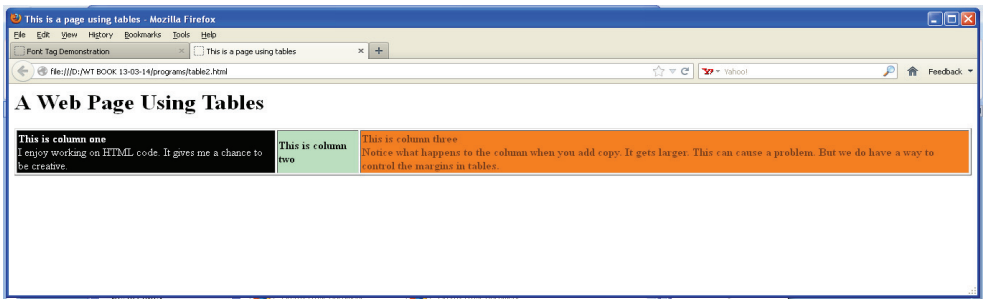
Step 2 Add the code highlighted in bold:

```
<html>
<head>
<title> This is a page using tables </title>
</head>
<body bgcolor="#ffffff" text="#000000">
<h1>A Web Page Using Tables</h1>
<table border="1">
<tr>
<td bgcolor="#000000">
<font color="#ffffff">
<b>This is column one</b>
<br>
I enjoy working on HTML code. It gives me a chance to be
creative.</font>
</td>
<td bgcolor="#bee3c2"> <b>This is column two</b></td>
<td bgcolor="#ff8000">
<font color="#804000">
<b>This is column three<br>Notice what happens to the column
when you add copy. It gets larger. This can cause a problem. But
we do have a way to control the margins in tables.</b>
</font>
</td>
</tr>
</table>
</body>
</html>
```

Step 3 Save the file as **table2.html**

Step 4 Opening Mozilla Firefox

Step 5 View the page in the **browser.**



A table with multiple rows and columns/cells

Now let us examine the column/cell tag. We learned about a couple of attributes--bgcolor and width. You can use these in the table tag (<table>), and the column/cell tag (<td>).

Step 1 Open up your text editor.

Step 2 Enter the following code.

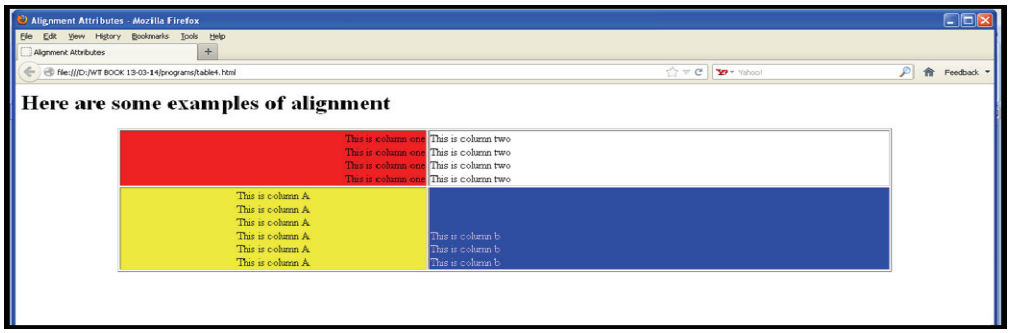
```
<html>
<head>
<title> Alignment Attributes </title>
</head>
<body>
<h1>Here are some examples of alignment</h1>
<table border="1" width="80%" align="center">
<tr>
<td bgcolor="#ff0000" width="40%" align="right">
This is column one<br>
This is column one<br>
This is column one<br>
This is column one<br>
</td>
<td width="60%">
This is column two<br>
This is column two<br>
This is column two<br>
This is column two<br>
</td>
</tr>
<tr> <td bgcolor="#ffff00" width="40%" align="center">
This is column A<br>
This is column A<br>
This is column A<br>
```

```
This is column A<br>
This is column A<br>
This is column A<br> </td>
<td bgcolor="#0000ff" width="60" valign="bottom">
<font color="#FFFFFF">
This is column b<br>
This is column b<br>
This is column b<br>
</font>
</td>
</tr>
</table>
</body>
</html>
```

Step 3 Save the file as **table4.html.**

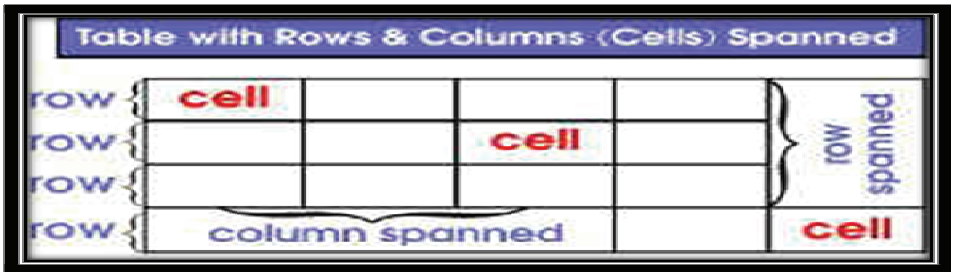
Step 4 Open Mozilla Firefox.

Step 5 View the page in the **browser.**



Spanning rows and columns

To begin, let us look at our previous illustration, but this time with some rows and columns expanded.



Use the following attributes to expand a column or a row.

- **colspan:** expands across a number of columns
- **rowspan:** expands across a number of rows

Let us try this with some code:

Step 1 Open up your text editor.

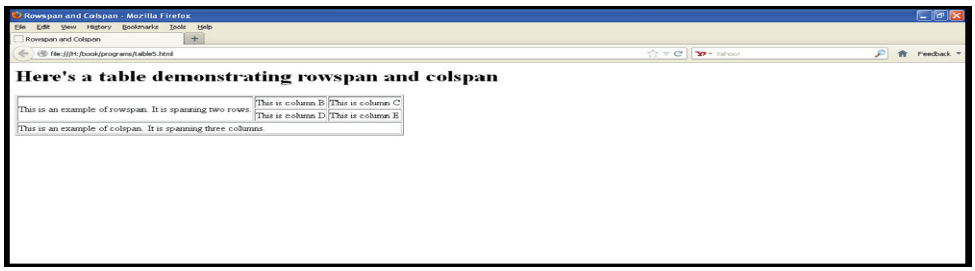
Step 2 Enter the following code.

```
<html>
<head>
<title> Rowspan and Colspan </title>
</head>
<body>
<h1>Here's a table demonstrating rowspan and colspan</h1>
<table border="1">
<tr>
<td rowspan="2">This is an example of rowspan. It is spanning two
rows.</td>
<td>This is column B</td>
<td>This is column C</td>
</tr>
<tr>
<td>This is column D</td>
<td>This is column E</td>
</tr>
<tr>
<td colspan="3">This is an example of colspan. It is spanning three
columns.</td> </tr>
</table>
</body>
</html>
```

Step 3 Save the file as **span.html.**

Step 4 Open Mozilla Firefox.

Step 5 View the page in the **browser.**



1.10 Frames

Frames provide a pleasing interface which makes your web site easy to navigate. When we talk about frames, actually we are referring to frameset, which is a special type of web page. The frameset contains a set of references to HTML files, each of which is displayed inside a separate frame. There are two tags related to frames i.e., frameset and frame.

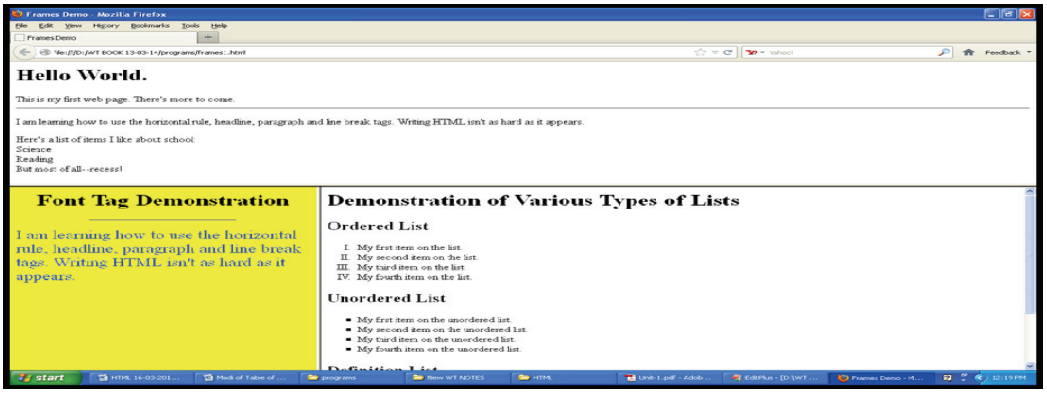
<frameset rows=" % , %" | cols=" % , %">.....</frameset>

<frame name="name" src="filename" scrolling =" yes" | "no"
frameborder ="0"|"1" noresize>

Example Program for implementing frames : frames1.html

```
<html>
<head>  <title> Frames Demo</title>
<frameset rows="40%,60%" >
<frame src="firstpage.html" scrolling="no" >
<frameset cols="10%,90%" >
<frame src="fontdemo.html">
<frame src="lists.html" name="f1">
</frame>
</frame>
</frameset>
</frame>
</frameset>
</head>
</html>
```

View the output page in the browser:



Frames Case Study

Design & Development of web application for an online book store web site.

The static home page must contain three **frames**.

Top frame: Logo and the college name and links to Home page, Login page, Registration page, Catalogue page and Cart page (the description of these pages will be given below).

Left frame: At least four links for navigation, which will display the catalogue of respective links. For e.g.: When you click the link **"CSE"** the catalogue for **CSE** Books should be displayed in the Right frame.

Right frame: The *pages to the links in the left frame must be loaded here*. Initially this page contains the description of the web site. The website should contain the following pages. Home page, Registration and user Login, User profile page, Books catalog and a Shopping cart.

HomePage.html

```
<html> <head> <title>SCITS FORM</title> </head>
<frameset rows="10%,10%,*" noresize border=0>
<frame src="top.html" name="logohome" noresize></frame>
<frame src="links.html" name="links" noresize></frame>
<frame src="bottom.html" name="det" noresize></frame>
</frameset>
</frameset>  </html>
```

top.html

```
<html>
<head>
<title>SCITS FORM</title>
</head>
<body bgcolor=pink>
<font color=red><center> <h2>SREE CHAITANYA INSTITUTE OF TECHNOLOGICAL SCIENCES</h2></center>
</font>
<body>
</html>
```

Links.html

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

```
<title>SCITS FORM</title>
</head>
<body bgcolor=pink>
<pre>
<a href="home.html" target="rs">Home</a>
<a href="login.html" target="rs">Login</a>
<a href="reg.html" target="rs">Registration</a>
<a href="catalogue.html" target="rs">Catalog</a>
<a href="cart.html" target="rs">Cart</a>
</pre>
<body>
</html>
```

Bottom.html

```
<html>
<head>
<title>SCITS FORM</title>
</head>
<frameset cols="20%,*" noresize border=0>
<frame src="6.html" name="ls" noresize></frame>
<frame src="home.html" name="rs" noresize></frame>
</frameset>
</html>
```

1. Home.html

```
<html>
<head>
<title>SCITS FORM</title>
</head>
<body bgcolor=pink>
<h2>This is Home Page.....</h2>
</body>
</html>
```

2. Login.html

```
<html>
<head>
<title>login form</title>
</head>
<body bgcolor=pink>
<H1><CENTER>LOGIN FORM</CENTER></H1>
<P ALIGN=CENTER>
<form method="post" ACTION="#">
      ENTER USERNAME: <input type="text" name="login" value="
"></input></br>
      ENTER PASSWORD: <input type="password" name="password"
value=""></input></br> <BR>
         &nb
sp;         &
nbsp;         
;    <input <type="submit" name="button"
value="submit"></input>
<input type="reset" name="button1" value="reset"></input>
</form>
</body>
</html>
```

3. Reg.html

```
<html>
<head>
<title>Registration Page</title>
</head>
<body bgcolor=pink>
<CENTER><H1>Registration Form</H1></CENTER>
<form method="post" ACTION="#">
Name<input type="text" name="name"></input></br>
Password<input type="password" name="pass"></input></br>
E-mail id<input type="text" name="email"></input></br>
Phone number<input type="text" name="phn"></input></br>
```

```
Sex<input type="radio" name="sex" value="male">male</input>
<input type="radio" name="sex" value="female">female</input></br>
Date of birth<select name="selector">
<option selected="selected" value=" ">select date</option>
<option value="01">01</option>  <option value="02">02</option>
<option value="03">03</option>  <option value="04">04</option>
<option value="05">05</option>  <option value="06">06</option>
<option value="07">07</option>  <option value="08">08</option>
<option value="09">09</option>  <option value="10">10</option>
<option value="11">11</option>  <option value="12">12</option>
<option value="13">13</option>  <option value="14">14</option>
<option value="15">15</option>  <option value="16">16</option>
<option value="17">17</option>  <option value="18">18</option>
<option value="19">19</option>  <option value="20">20</option>
<option value="21">21</option>  <option value="22">22</option>
<option value="23">23</option>  <option value="24">24</option>
<option value="25">25</option>  <option value="26">26</option>
<option value="27">27</option>  <option value="28">28</option>
<option value="29">29</option>  <option value="30">30</option>
<option value="31">31</option>
</select>
<select name="selector">
<option selected="selected">select month</option>
<OPTION>JAN</OPTION>  <OPTION>FEB</OPTION>
<OPTION>MAR</OPTION>  <OPTION>APR</OPTION>
<OPTION>MAY</OPTION>  <OPTION>JUN</OPTION>
<OPTION>JUL</OPTION>  <OPTION>AUG</OPTION>
<OPTION>SEP</OPTION>  <OPTION>OCT</OPTION>
<OPTION>NOV</OPTION>  <OPTION>DEC</OPTION>
</select>
<select name="selector">
<option selected="selected">select year</option>
<option>1988</option>  <option>1989</option>
```

```
<option>1990</option>  <option>1991</option>
<option>1992</option>  <option>1993</option>
<option>1994</option>
</select></br>
Languages known:<input type="checkbox">Telugu</input>
<input type="checkbox">Hindi</input>
<input type="checkbox">English</input>
<input type="checkbox">Tamil</input></br>
Address<TEXTAREA NAME="" ROWS="2" COLS="15"></TEXTAREA>
<BR>
<BR>
          &nb
sp;         &
nbsp;         
;    <input type="submit" name="button"
value="submit"></input>
<input type="reset" name="button1" value="reset"></input>
</form> </body> </html>
```

4. Catalogue.html

```
<html>  <head> <title>Catalogue Page</title> </head>
<body bgcolor=PINK>
<H1><CENTER>Catalogue Page</CENTER></H1>
<table align="center" border="3">
<tr>
<td><img src="7023.jpg" width=25 height=40></td>
<td>Book:XML Bible</br>Author:Winston</br>Publication:Wiley</td>
<td>$40.5</td>
      <td><form method="post"><input type="button" value="add to
cart"></input></td>
</tr>
<tr>
<td><img src="7030.jpg" width=25 height=40></td>
<td>Book:AI</br>Author:S.Russel</br>Publication:Princeton hall</td>
```

```
<td>$63</td>
<td><input type="button" value="add to cart"></input></td>
</tr>
<tr>
<td><img src="7033.jpg" width=25 height=40></td>
<td>Book:Java 2</br>Author:Watson</br>Publication:BPB
publications</td>
<td>$35.5</td>
<td><input type="button" value="add to cart"></input></td>
</tr>
<tr>
<td><img src="7195.jpg" width=25 height=40></td>
      <td>Book:HTML in 24 hours</br>Author:Sam
Peter</br>Publication:Sam publication</td>
<td>$50</td>
<td><input type="button" value="add to cart"></input></td>
</tr>
</table>
</form>
</body>
</html>
```

5. cart.html

```
<html>
<head>
<title>Cart Page</title>
</head>
<body bgcolor=pink>
<CENTER><H1>Cart Page</H1></CENTER>
<table align="center">
<tr>
<td><b>Book name</b></td>
<td><b>Price</b></td>
```

```
<td><b>Quantity</b></td>
<td><b>Amount</b></td>
</tr>
<tr>
<td>java 2</td>
<td>$35.5</td>
<td>2</td>
<td>$70</td>
</tr>
<tr>
<td>XML bible</td>
<td>$40.5</td>
<td>1</td>
<td>$40.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total amount-</td>
<td>$130.5</td>
</tr>
</table>
</body>
</html>
```

6. html (Branches)

```
<html>
<body bgcolor=pink>
<a href="7.html" target="rs">CSE</a></br>
<a href="8.html" target="rs">ECE</a></br>
<a href="9.html" target="rs">EEE</a></br>
<a href="10.html" target="rs">CIVIL</a>
</body>
</html>
```

7. html (CSE SUBJECTS)

```
<html>
<body bgcolor=pink>
<H1>CSE SUBJECTS</H1>
Operating systems</br>
Compiler design</br>
Computer networks</br>
Information security</br>
Artificial Intelligence and Neural Networks</br>
Object Oriented Analysis and Design</br>
Computer Networks and Case Tools Lab</br>
Operating systems and Compiler Design Lab
</body>
</html>
```

8. html (ECE SUBJECTS)

```
<html>
<body bgcolor=pink>
<H1>ECE SUBJECTS</H1>
Management science</br>
Telecommunication switching systems</br>
Digital signal processing</br>
VLSI design</br>
Micro wave engineering</br>
Micro processors and interfacing</br>
Electronic computer aided design lab</br>
advanced english communication skills lab
</body>
</html>
```

9. html (EEE SUBJECTS)

```
<html>  <body bgcolor=pink>
<H1>EEE SUBJECTS</H1>
Digital Signal Processing</br>
Microprocessors and Microcontrollers</br>
```

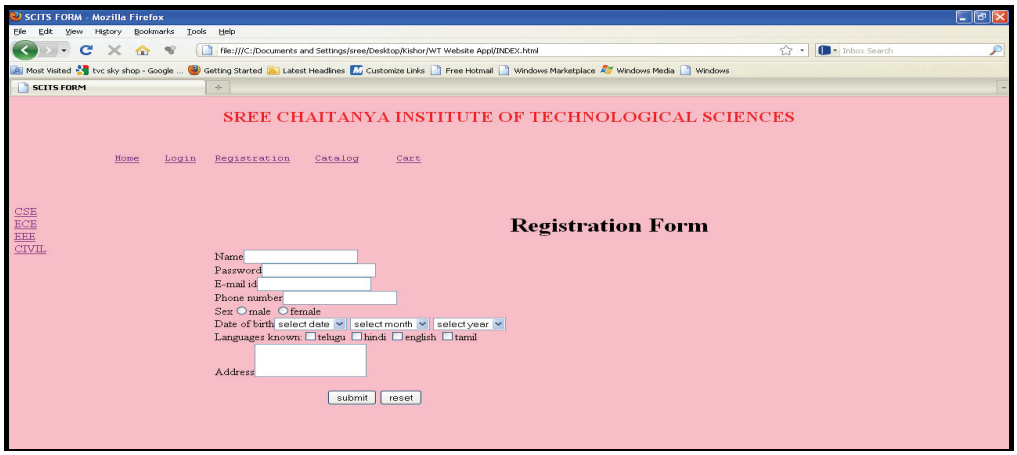
```
Management Science</br>
Instrumentation</br>
Switch Gear and Protection</br>
VLSI Design</br>
Advanced English Communications Skills Lab</br>
Power Electronics and Simulation Lab</br>
</body>  </html>
```

10. html (CIVIL SUBJECTS)

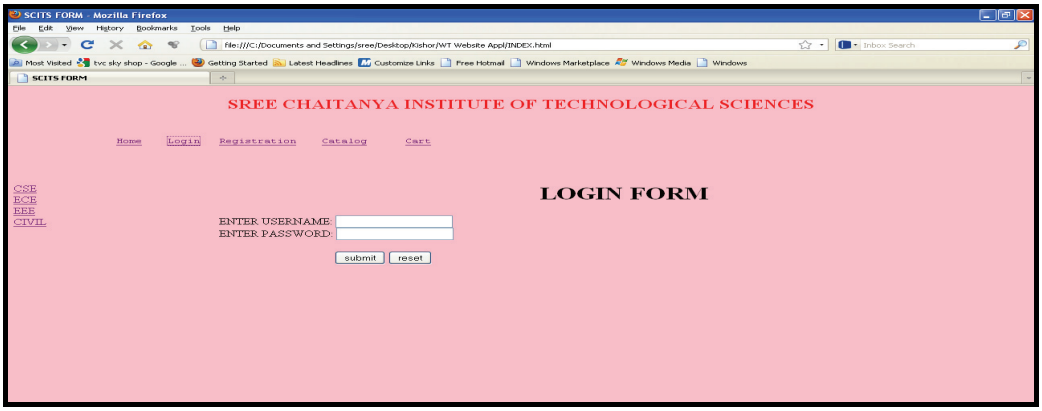
```
<html>
<body bgcolor=pink>
<H1>CIVIL SUBJECTS</H1>
Geotechnical engineering</br>
Environmental engineering</br>
Design of steel structres</br>
Water resource engineering</br>
Estimating and costing</br>
Transportation engineering</br>
Geo technical engineering lab</br>
Environmental engineering lab
</body>
</html>
```

Output Screens

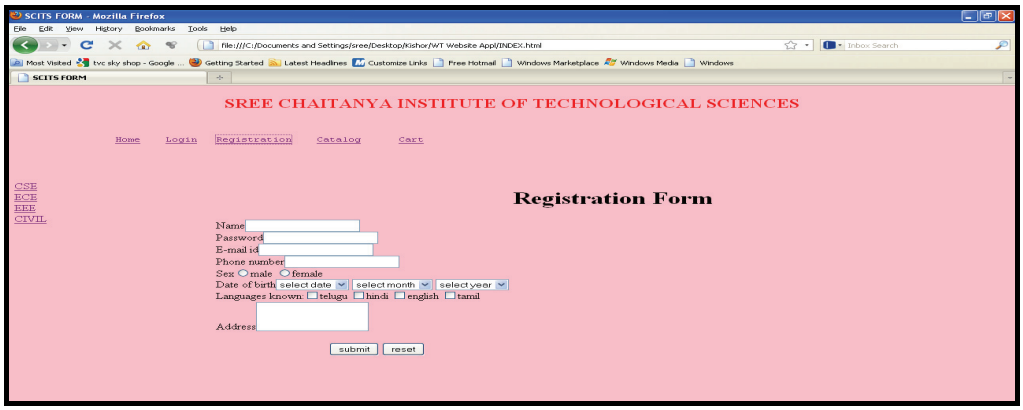
Home Page



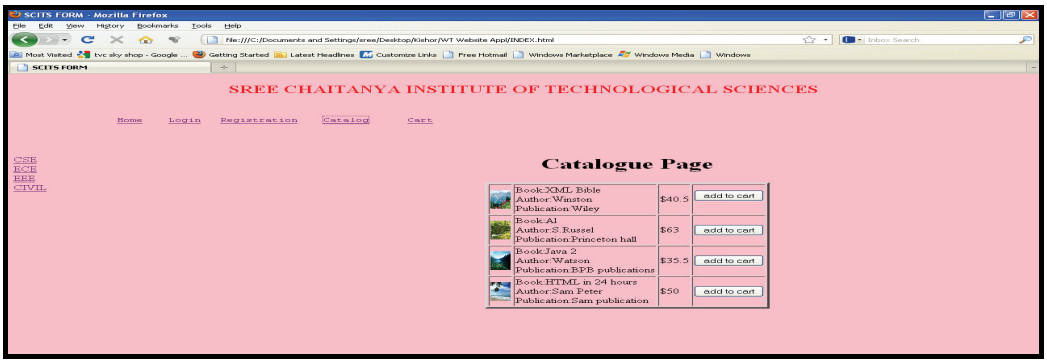
Login Page



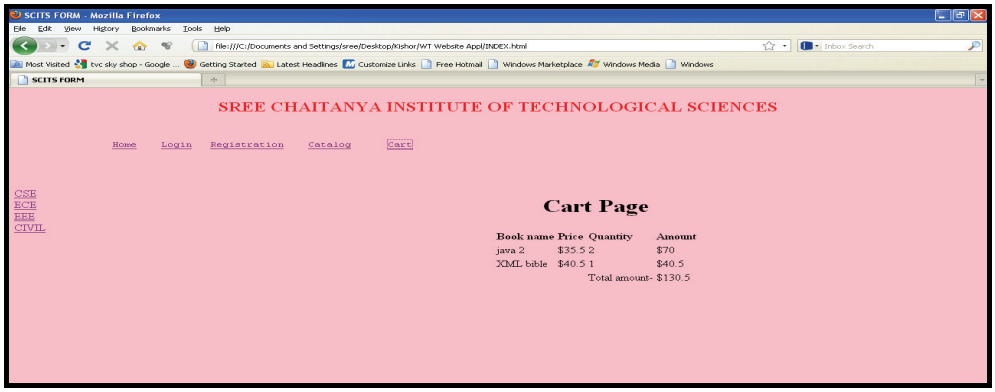
User Profile



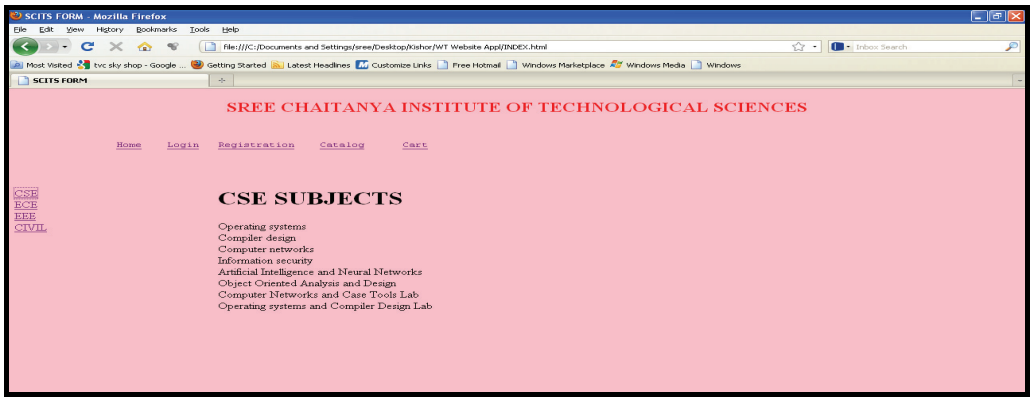
Catalogue Page



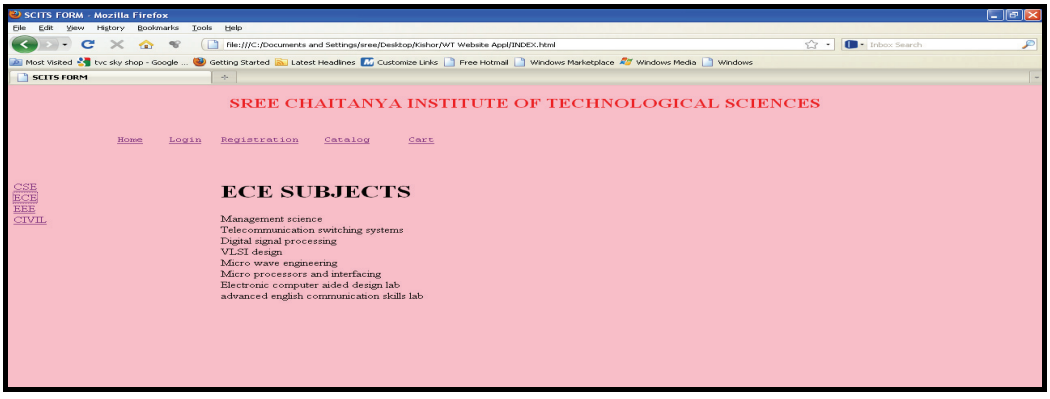
Cart Page



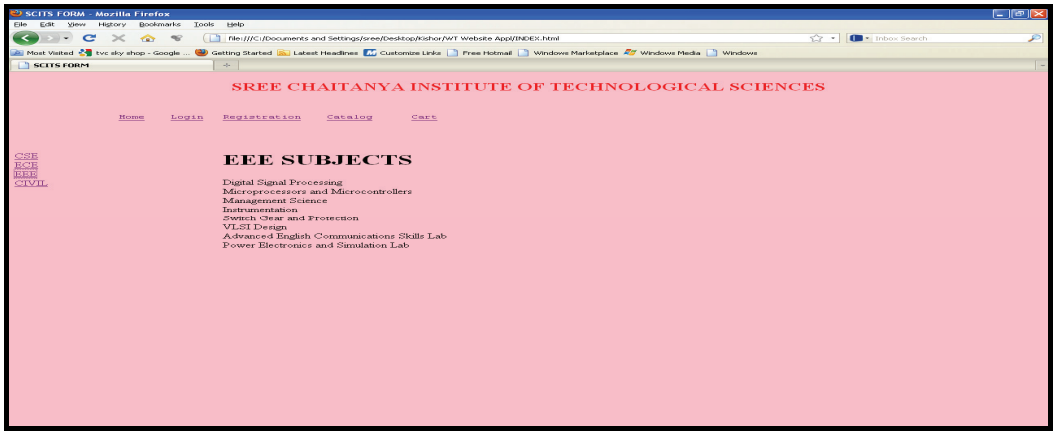
CSE Subjects



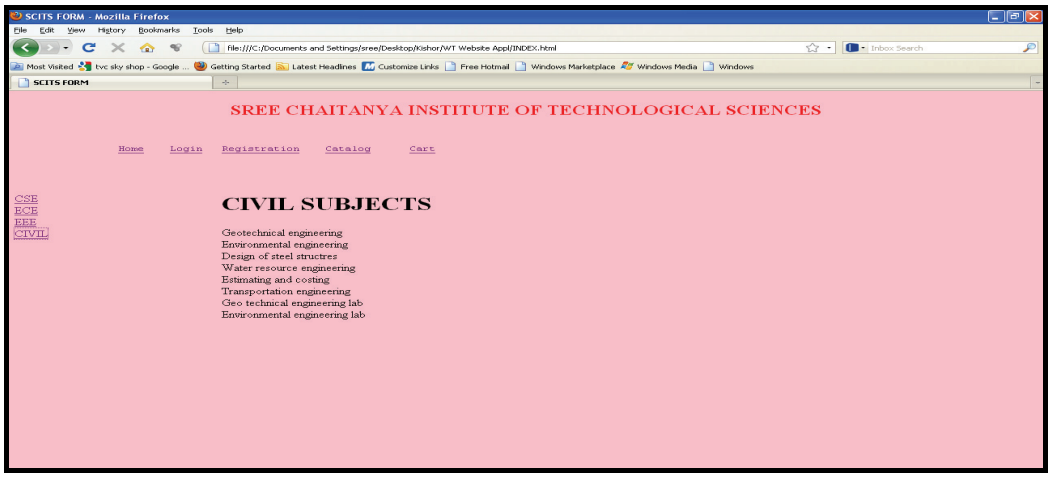
ECE SUBJECTS



EEE Subjects



CIVIL Subjects



1.11 Forms

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

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

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

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

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

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

Password: It works exactly as text, but the content is not displayed on the screen. Instead, an '*' is used.

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

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

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

Reset: This creates a Reset button which clears the data in form fields.

<select name="string">…..</select>

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

<option value="string" selected>……</option>

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

<text area name="string" rows="n"cols="n">…….</textarea>

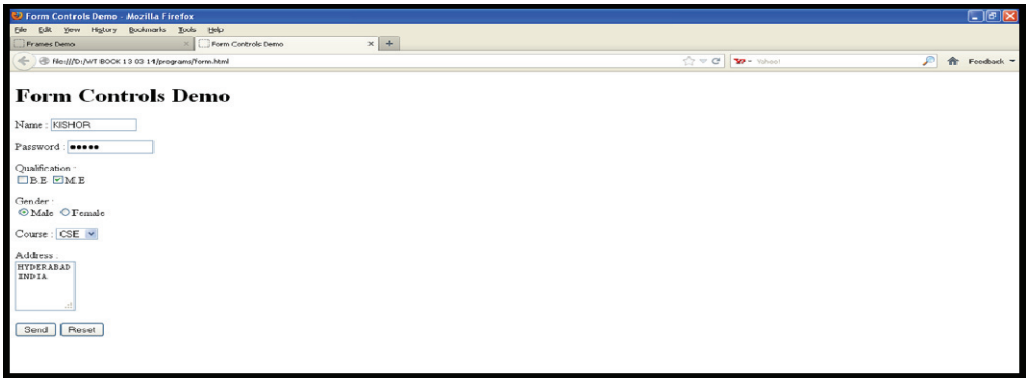
This creates a free format of plain text into which the user can enter anything they like. The area will be sized at rows by columns but supports automatic scrolling.

Example Program for implementing Form Fields: form.html

```
<html>
<head>
<title>Form Controls Demo</title>
</head>
```

```
<body> <form action="target.html" method="get">
<h1>Form Controls Demo</h1>
<p align="left">Name : <input type="text" maxlength=30 size=15>
<p align="left">Password : <input type="password" maxlenght=10
size=15>
<p align="left">Qualification : <br>
<input type="checkbox" name="q" value="be">B.E
<input type="checkbox" name="q" value="me">M.E
<p align="left">Gender : <br>
<input type="radio" name="g" value="m">Male
<input type="radio" name="g" value="f">Female
<p align="left">Course :
<select name="course" size=1>
<option value=cse >CSE</option>
<option value=it>CSIT</option>
</select>
<p align="left">Address : <br>
<textarea name="addr" rows=4 cols=5 scrolling="yes"></textarea>
<br><br>
<input type="submit" name="s" value="Send">
<input type="reset" name="c" value="Reset">
</form>
</body>
</html>
```

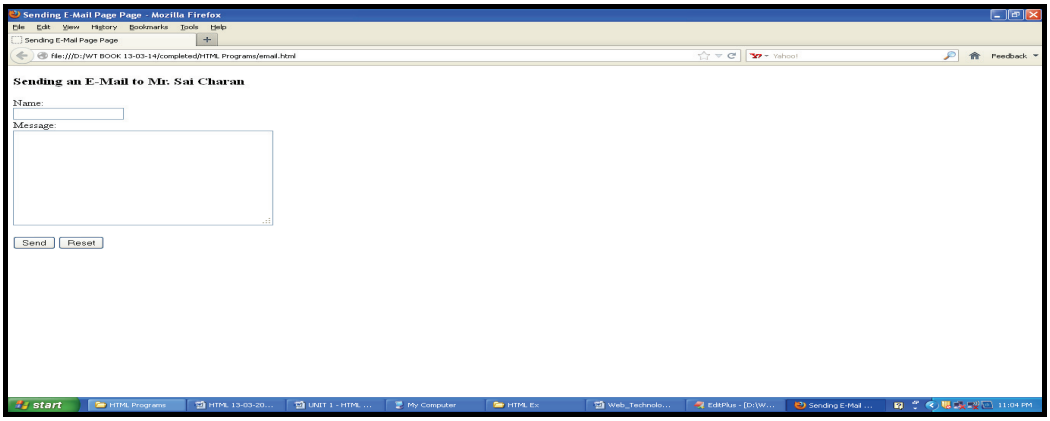
View the output page in the browser:



Example Program for implementing Form Fields: email.html

```
<html>
  <head>
        <title> Sending E-Mail Page Page </title>
  </head>
  <body>
<form action="MAILTO:peddi.kishor@gmail.com" method="post"
enctype="text/plain">
        <h3>Sending an E-Mail to Mr. Sai Charan</h3>
        Name:<br/>
        <input type="text" name="name" value=" " size="20">
        <br/>
        Message:<br/>
        <textarea cols="40" rows="10" name="message">
        </textarea>
        <br/><br/>
        <input type="submit" value="Send">
        <input type="reset" value="Reset">
  </form>
  </body>
</html>
```

View the output page in the browser:



1.12 Cascading Style Sheets

One of the most important aspects of HTML is the capability to separate presentation and content. A style is simply a set of formatting instructions that can be applied to a piece of text. There are three mechanisms by which we can apply styles to our HTML documents. Style can be defined within the basic HTML tag. Style can be defined in the <head> tag. Styles can be defined in external files called style sheets which can then be used in any document by including the style sheet via a URL.

A style has two parts: a selector and a set of declarations.

The selector is used to create a link between the rule and the HTML tag.

The declaration has two parts: a **property** and a **value**. Declarations must be separated using colons and terminated using semicolons.

Selector {property: value; property: value …….}

Properties and values in styles:

Fonts:

font-family: family name;

font-style: normal | italic | oblique;

font-weight: normal | bold | bolder | lighter;

font-size: small | medium | large | smaller | larger;

Backgrounds and Colors:

color: value;

background-color: value;

background-image: URL;

background-repeat:repeat-X;

background-position:center|top|bottom|left|right|center;

Text:

text-decoration: none | underline | overline| line-through| blink;

text-transformation: none | uppercase | lowercase| capitalize;

text-align: left | right | center | justify;

line-height : length | percentage;

letter-spacing: length | percentage;

word-spacing: length | percentage;

link:

a:link{color:green;font-size:18px;}

a:hover{color:orange;}

a:visited{color:blue,text-transform:uppercase;}

a:active{color:black;}

list:

list-style-type: lower-greek |upper-latin|roman;

list-style-image: url("new.gif");

border: border-style:dashed |solid |double |dotted;

border-color:color;

border-width:top |bottom |left |right;

table:

width:length| percentage;

border:solid| dotted...;

vertical-align:bottom|center;

Grouping Selectors

As you add more and more styles to your pages, you might find yourself making the same stylistic change to multiple XHTML elements. In these cases, you might consider grouping your CSS selects. This can shorten the amount of code and make for a quicker download.

h1 {color: blue; font-family: verdana}

h2 {color: blue; font-family: verdana}

h3 {color: blue; font-family: verdana}

They could be optimized and grouped like this:

h1, h2, h3 {color: blue; font-family: verdana}

This rule specifies that all the text within h1, h2, h3 tags will display in the Verdana font with a blue color.

Class Selectors

In the preceding examples, you learned how to create styles based on XHTML elements using selectors. Selectors are related to XHTML elements. If you wanted to apply a style to something that was not a tag (let us say there was a certain sentence in your document that you wanted to be bold, but it was

not anything other than text from a structural definition), this kind of situation is where Class selectors are of value. Another situation in which a class selector comes in handy is when you want to style the appearance of XHTML tags differently. Earlier you learned how to redefine the appearance of an XHTML element (the h1 tag). Suppose you do not want to format every h1 tag the same way. This is a perfect place to consider creating a Class selector. Consider the following code:

- Class selectors are written in this syntax: always beginning with a dot (.) and a unique name.
- The Class selector is applied to the XHTML element using the class attribute and the name you assigned to the class. In this example, we gave the class the name of "mytext". Notice that the dot (.) is not included in the class attribute.

Example – classselector.html

```
<html>
<head>
<style type="text/css">
<!--
.mytext {
  color: #FF6600;
}
-->
</style>
</head>
<body>
<p class="mytext">This is how you use class selectors.
</p>
<p> </p>
</body>
</html>
```

By creating a Class selector, you could now apply this formatting anywhere in your document independent of the XHTML element. It will apply this format only where it encounters this class, not across every instance of an XHTML element like previously shown.

Here is an example of Class selector being applied only to a specific portion of a paragraph:

```
 <html>
<head>
<style type="text/css">
<!--
.mytext {color: #FF6600; font-family:verdana; font-weight:bold;}
-->
</style>
</head>
<body>
<p>This is how you use <span class="mytext"> class</span> selectors.
</p>
<p> </p>
</body>
</html>
```

- The Class selector syntax is displayed with the font-family and the font-weight properties applied.
- The tag is used to designate the area of text that is to be formatted using the Class selector. Notice that there is a closing tag to end the formatting.

Block-Level and Inline-Level Elements

Block-level elements act like boxes that start at the margin of one line of text and end so that the content after the closing element is forced to start from a new line of text. The content of a block-level element can be, and typically is, several lines long. Basically, block level elements start and end a line of text. For example, the paragraph element is a block-level element. It starts at one margin and anything that comes after the closing element is forced to appear on a new line. Any formatting applied to a block-level element will affect everything within it.

```
<html>
<head>
  <style type="text/css">
<!--
.mytext {
  color: #CC9900;
}
-->
</style>
```

```
</head>
<body>
<div class="mytext">
  <p>Greenwaves, Inc </p>
  <p>Company Information</p>
</div>
</body>
</html>
```

A element is used to create a range or an invisible box around the two <p> elements. The body Class is attached to the element, which causes both paragraphs to be formatted with the body Class. So, instead of applying the body Class to both paragraph elements, it was applied once to the element. This results in less code, which is a good thing.

Inline elements: As described, inline-elements act like boxes as well, except they are used within block-level elements. This is the difference between the two. For example, if you have a element that formats a paragraph of text with a style sheet and you want to format a single word within that element, you would use the inline element .

```
<html>
<head>
  <style type="text/css">
<!--
.color1 {
  font-family: "Times New Roman", Times, serif;
  color: #FF3300;
}
.color2 {
  font-family: Arial, Helvetica, sans-serif;
  color: #339933;
}
-->
</style>
</head>
<body>
 <h1 class="color1">How are we <span class="color2"> doing </span>
today? </h1>
</body>
</html>
```

Pseudo-Classes

So far, you have learned about selectors, classes, and Ids. Things are about to get interesting with pseudo-classes. These are class selector that let the designer apply styles to element that do not exist within the document. A pseudo class lets you apply styles to elements that you know will exist, you just do not know when. For example, users are likely to move their cursor over a link at some point. The anchor tag is probably the most common pseudo-class. Here is an example:

```
<html>
<head>
  <style type="text/css">
<!--
a:link {
  color: #FF6600;
}
a:visited {
  color: #FFCC33;
}
a:hover {
  color: #0066FF;
}
a:active {
  color: #99CC33;
}
-->
</style>
</head>
<body>
<a href="http://www.cnn.com">link to cnn
</a>
</body>
</html>
```

The pseudo class is automatically applied to every anchor tag in the document. You do not have to add any additional code in the <body> portion of the document. The following chart identifies the pseudo-classes available in the CSS2 recommendation. Note that pseudo-class selectors and links must be in a specific order to work properly.

:link, :visited, :hover, :active, :first-child, :lang

Types of Style Sheets

There are three popular types of style sheets: Internal (embedded), External, and Inline. There is also another type of style sheet called as Imported.

Internal Style Sheets

Here is an example of 'internal' style sheet. Notice that the <style> element is used inside the <head> portion of the page. All of the formatting information is contained within the HTML page. These types of style sheets will only affect the appearance of this single page. They can be effective for creating efficient designs within the single page.

Example:

```
<html>
<head>
 <style type="text/css">
<!--
h1 {
  color: #FF0000;
}
-->
</style>
</head>
<body>
 <h1>How are we doing today? </h1>
</body>
</html>
```

External Style Sheet

Here is an example of an external style sheet. Notice that the <link> element is using the <href> attribute to point to the "external.css" file. The external.css file is the document that contains all the formatting information for this XHTML page. Because the formatting information remains independent of the XHTML page, it can easily be applied to several XHTML pages and updated more efficiently.

```
<html>
<head>
<link href="external.css" rel="stylesheet" type="text/css" />
</head>
<body>
<h1> Check it Out! </h1>
<p> My style is really cool! </p>
</body>
</html>
```

The above program references an external CSS called external.css. You can give your style sheet any name you want; just be sure to save it in text mode only and upload it to the same directory. Make sure to give it a .css extension. You do not have to put an external style sheet inside a comment tag because no style information is stored within that code.

Your **external.css** document could look like this:

```
h1 {color: red; font-family: sans-serif}
p {background: black; color:yellow; font-family: verdana;}
```

Inline Style Sheet

An inline style sheet only applies to the parts of an HTML document that are specified and will override any style settings being applied by an external or internal style sheet.

```
<html>
<body>
<h1 style="font-family:s ans-serif; color: gray"> Check it Out! </h1>
<h1> My style is really cool! </h1>
</body>
</html>
```

In the above example, Inline style will override both external and internal style. Internal will override external.

ID Vs. Class Selectors

At this point, you might be wondering about the difference between a Class and an ID selector. Functionally, it does not really matter because they both accomplish the same thing. However, there are a few differences.

Ids start with a # symbol instead of a dot (.). Also, it is considered an error to use the same ID selector more than once. In fact, Ids should be used only once within a given document, and they should always have a unique name different from other Ids in the document. This makes them great for the absolute positioning, where they would only want to use them once. Classes are great for formatting text because they can be used multiple times on a web page. Because of this, you will find yourself probably using classes most of the time.

HTML program for demonstrating a cascading style sheet – stylesheets.html

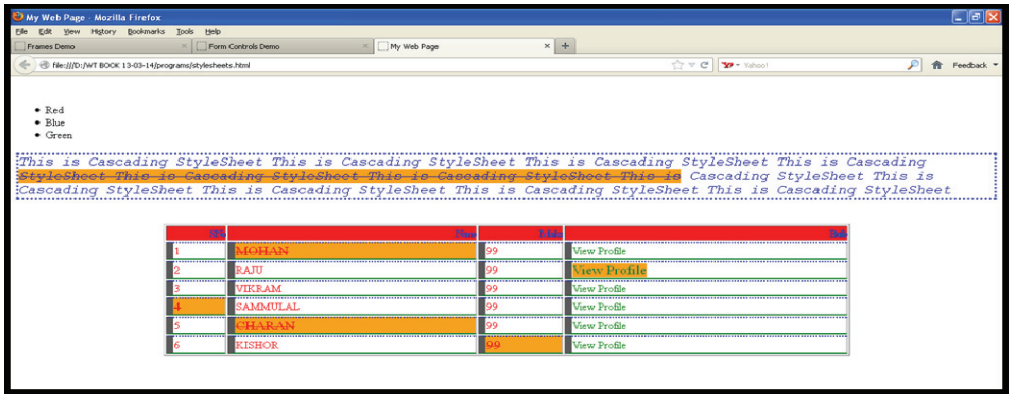
```
<html>
<head>
<title>My Web Page</title>
<link href="CSS\MyStyle.css" rel="stylesheet" type="text/css"/>
<style type="text/css">
li{
list-style-image:url('CSS//new.gif');
}
.newstyle{
text-decoration:line-through;
font-size:20px;
background-color:orange;
}
p{
border-style:dotted;
font-family:courier;
font-size:20px;
color:blue;
font-weight:normal;
```

```
font-style:italic;
}
a:visited{color:black;}
a:hover{visibility:hidden;}
a:active{text-transform:uppercase;color:blue;}
a:link{color:green;text-decoration:blink;}
td{
color:red;
border-top-style:dotted;
border-bottom-style:solid;
border-width:2px;
border-left-width:10px;
border-top-color:blue;
border-bottom-color:green;
}
th{
color:blue;
letter-spacing:-3px;
text-align:right;
background-color:red;
text-decoration:blink;
text-transform:capitalize;
}
</style>
</head>
<body>
<br><ul>
<li>Red</li>
<li>Blue</l>
<li>Green</li></ul>
<p>This is Cascading StyleSheet This is Cascading StyleSheet This is
Cascading StyleSheet This is Cascading
<span class="newstyle">StyleSheet This is Cascading StyleSheet This is
```

```
Cascading StyleSheet This is</span> Cascading StyleSheet This is
Cascading StyleSheet This is Cascading StyleSheet This is Cascading
StyleSheet This is Cascading StyleSheet
</p><br>
<table width=70% align=center border>
<tr>
<th>S.No</th>
<th>Name</th>
<th>Marks</th>
<th>Profile</th>
</tr>
<tr>
<td>1</td>
<td class="newstyle">MOHAN</td>
<td>99</td>
<td><a href="Profile.html">View Profile</a></td>
</tr>
<tr><td>2</td>
<td>RAJU</td>
<td>99</td>
<td><a href="Profile1.html" class="newstyle" >View Profile</a></td>
</tr><tr>
<td>3</td>
<td>VIKRAM</td>
<td>99</td>
<td><a href="Profile.html">View Profile</a></td>
</tr><tr>
<td class="newstyle">4</td>
<td>SAMMULAL</td>
<td>99</td>
<td><a href="Profile.html">View Profile</a></td>
</tr><tr> <td>5</td>
<td class="newstyle">CHARAN</td>
<td>99</td>
```

```
<td><a href="Profile.html">View Profile</a></td>
</tr><tr>
<td>6</td>
<td>KISHOR</td>
<td class="newstyle">99</td>
<td><a href="Profile.html">View Profile</a></td>
</tr>
</table>
</body>
</html>
```

View the output page in the browser:



More Example Programs

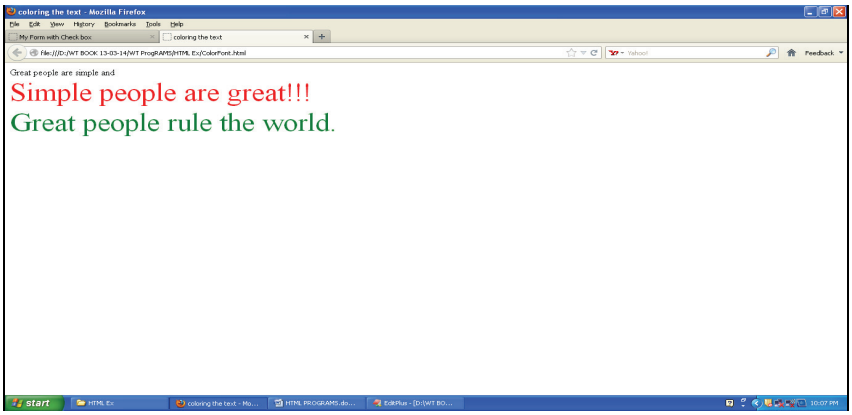
1. HTML program for font tag.

Font.html

```
<html>
<html>
<head>
        <title>coloring the text</title>
</head>
<body>
        <basefont face="arial" size="5" color="blue">
        Great people are simple and <br>
        <font face="Times new roman" color="red" size="8">
        Simple people are great!!!<br>
```

```
        <font face="GoudyHandtooled BT" color="green" size="14">
        Great people rule the world.
    </body>
</html>
```

Output:

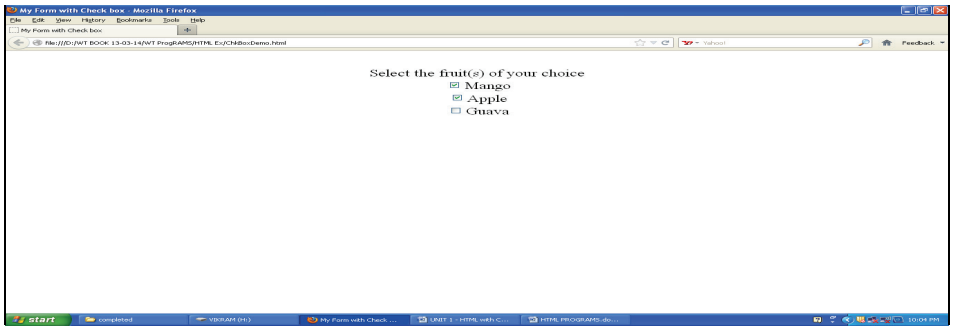


2. HTML Program for check boxes.

Checkbox.html

```
<html>
    <head>
        <title>My Form with Check box</title>
    </head>
    <body>
        <form name ="checkboxForm">
        <div align="center"><br>
        Select the fruit(s) of your choice<br/>
<input type="checkbox" name="option1" value="mango"
checked="checked">Mango<br/>
<input type="checkbox" name="option2" value="apple">Apple<br/>
<input type="checkbox" name="option3" value="guava">Guava<br/>
        </div>
        </form>
    </body>
</html>
```

Output:

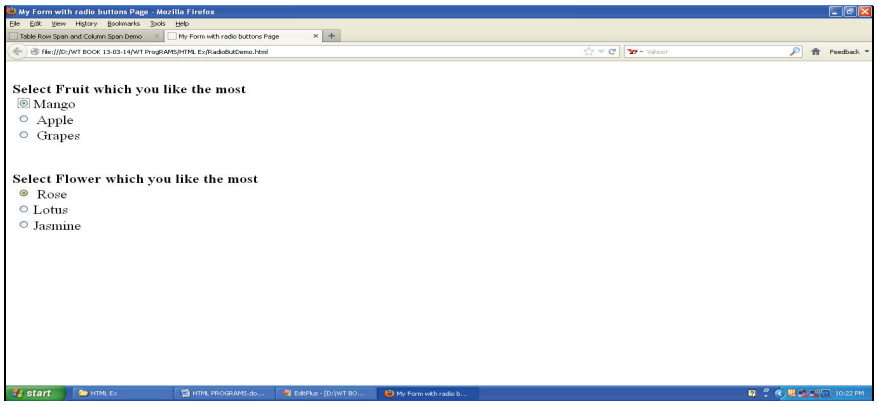


3. HTML Program for Radio Buttons.

radiobuttons.html

```
<html>
<head>
        <title>My Form with radio buttons Page</title>
</head>
<body>
<form name="myform">
        <div align="left"><br>
        <b>Select Fruit which you like the most</b><br/>
        <input type="radio" name="group1" value="Mango">Mango<br/>
        <input type="radio" name="group1" value="Apple" checked>
        Apple<br/>
        <input type="radio" name="group1" value="Grapes" checked>
        Grapes
        <br/><br/><br/>
        <b>Select Flower which you like the most</b><br/>
        <input type="radio" name="group2" value="Rose"> Rose<br/>
        <input type="radio" name="group2" value="Lotus">Lotus<br/>
        <input type="radio" name="group2" value="Jasmine"
checked>Jasmine<br/>
        </div>
</form>
</body>
</html>
```

Output:

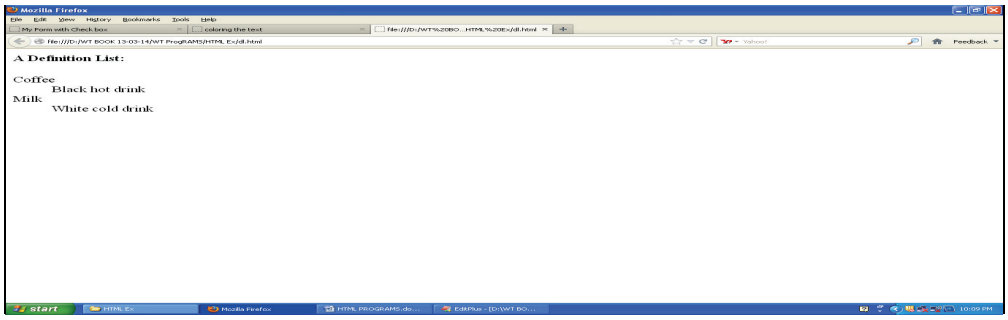


4. HTML program for Definition Lists.

definitionlist.html

```
<html>
<body>
<h4>A Definition List:</h4>
<dl>
<dt>Coffee</dt>
 <dd>Black hot drink</dd>
 <dt>Milk</dt>
 <dd>White cold drink</dd>
</dl>
</body>
</html>
```

Output:



5. HTML Program for Frames.

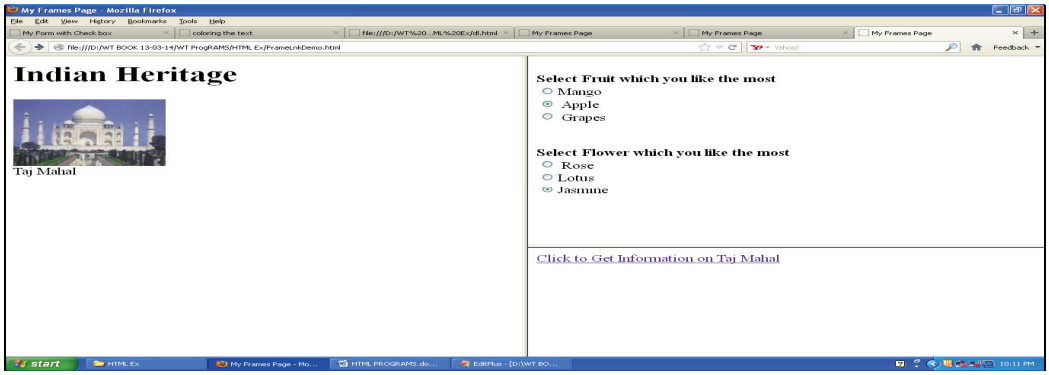
frames.html

```
<html>
<head>
        <title>My Frames Page</title>
</head>
        <frameset cols="50%,50%">
                <frame src="TajImg.html" name="Left_Vertical" noresize />
                <frameset rows="*,170">
                        <frame src="RadioButDemo.html" name="Right_top" />
                        <frame src="test_link.html" name="Right_Bottom"/>
                </frameset>
        </frameset>
</html>
```

test_link.html

```
<html>
  <head>
  </head>
   <body>
        <a href="Tajmahal.html" target="Right_Bottom"> Click to Get Information on Taj Mahal </a>
   </body>
</html>
```

Output:

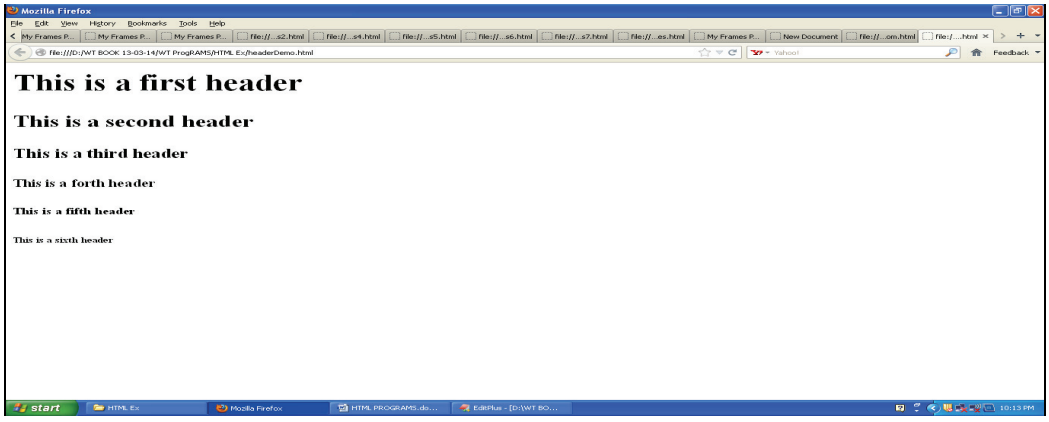


6. HTML Program for Heading Tags

headings.html

```
<html>
<head>
</head>
<body>
        <h1>  This is a first header </h1>
        <h2>  This is a second header </h2>
        <h3>  This is a third header </h3>
        <h4>  This is a forth header </h4>
        <h5>  This is a fifth header </h5>
        <h6>  This is a sixth header </h6>
</body>
</html>
```

Output:



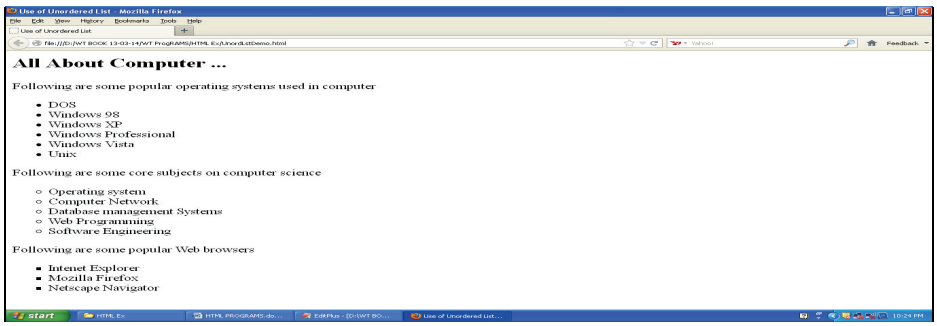
7. HTML Program for unordered list.

unorderedlist.html

```
<html>
<head>
        <title> Use of Unordered List </title>
</head>
<body>
<h2>All About Computer ...</h2>
```

```
    Following are some popular operating systems used in computer
    <ul type="disc">
        <li>DOS</li>
        <li>Windows 98</li>
        <li>Windows XP</li>
        <li>Windows Professional</li>
        <li>Windows Vista</li>
        <li>Unix</li>
    </ul>
    Following are some core subjects on computer science
    <ul type="circle">
        <li>Operating system</li>
        <li>Computer Network</li>
        <li>Database management Systems</li>
        <li>Web Programming</li>
        <li>Software Engineering</li>
    </ul>
    Following are some popular Web browsers
    <ul type="square">
        <li>Internet Explorer</li>
        <li>Mozilla Firefox</li>
        <li>Netscape Navigator</li>
    </ul>
    </body>
</html>
```

Output:

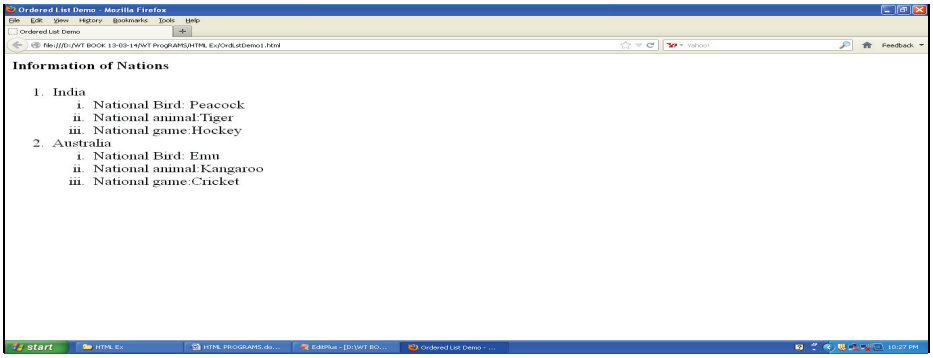


8. HTML program for Ordered Lists.

orderedlist.html

```
<html>
<head>
        <title> Ordered List Demo </title>
</head>
<body>
        <h4>Information of Nations</h4>
        <ol type="1">
        <li>India
                <ol type="i">
                        <li>National Bird: Peacock</li>
                        <li>National animal:Tiger</li>
                        <li>National game:Hockey</li>
                </ol>
        </li>
        <li>Australia
                <ol type="i">
                        <li>National Bird: Emu</li>
                        <li>National animal:Kangaroo</li>
                        <li>National game:Cricket</li>
                </ol>
        </li>
        </ol>
</body>
</html>
```

Output:

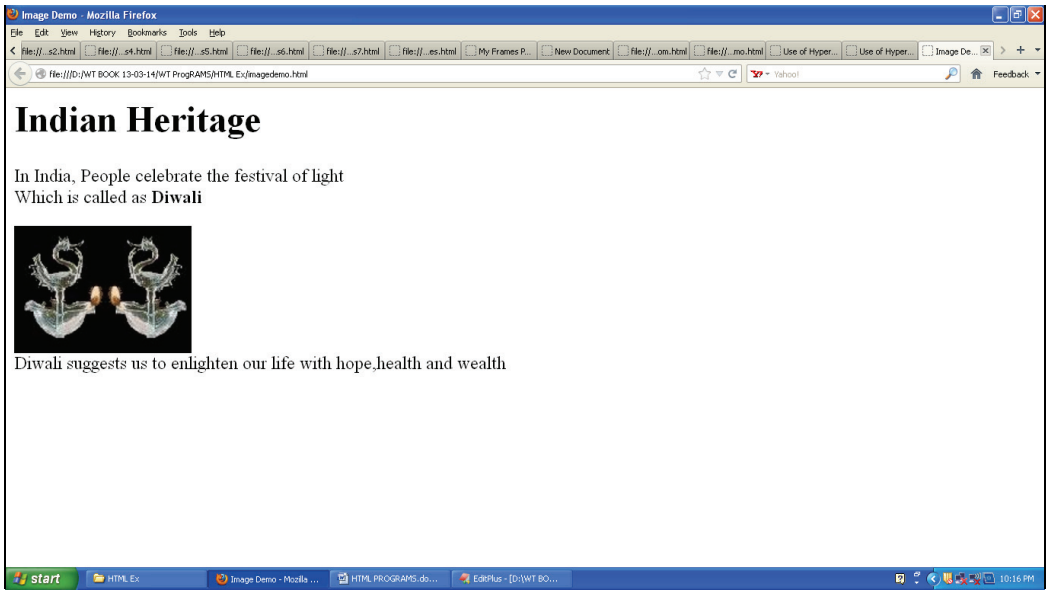


9. HTML Program for Image Demo.

image.html

```
<html>
<head>
        <title> Image Demo </title>
</head>
<body>
        <h1> Indian Heritage </h1>
        <p>
        In India, People celebrate the festival of light <br/>
        Which is called as <strong>Diwali</strong>
        </p>
        <img src= "lamp1.jpg" alt= "Light!!" />
        <br/>
        Diwali suggests us to enlighten our life with hope,health and wealth
        <br/>
</body>
</html>
```

Output:



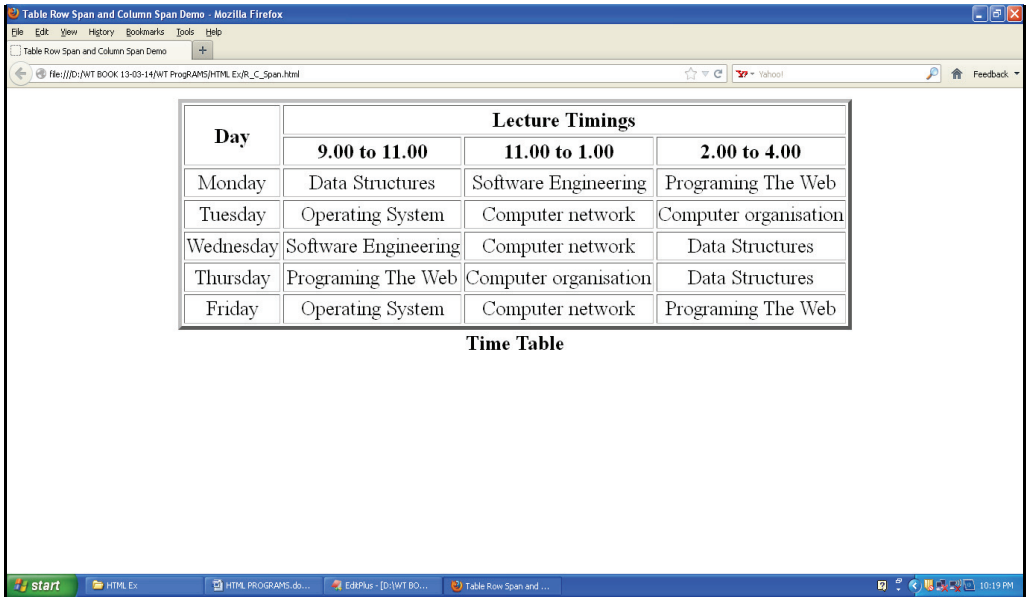
10. HTML Program for Table Row Span & Column Span.

tablespan.html

```
<html>
<head>
    <title> Table Row Span and Column Span Demo</title>
</head>
<body>
<center>
<table border="3" align="center">
<caption align="bottom">
    <strong>Time Table</strong>
</caption>
    <tr align="center">
        <th rowspan=2> Day
        <th colspan=3>  Lecture  Timings
    </tr>
    <tr>
        <th>9.00 to 11.00
        <th>11.00 to 1.00
        <th>2.00 to 4.00
    </tr>
    <tr align=center>
        <td> Monday
        <td>Data Structures
        <td>Software Engineering
        <td>Programing The Web
    </tr>
    <tr align=center>
        <td>Tuesday
        <td>Operating System
        <td>Computer network
        <td>Computer organisation
    </tr>
    <tr align=center>
        <td>Wednesday
        <td>Software Engineering
        <td>Computer network
```

```
            <td>Data Structures
        </tr>
        <tr align=center>
            <td>Thursday
            <td>Programing The Web
            <td>Computer organisation
            <td>Data Structures
        </tr>
        <tr align=center>
            <td>Friday
            <td>Operating System
            <td>Computer network
            <td>Programing The Web
        </tr>
        </table>
    </center>
    </body>
</html>
```

Output:



Day	Lecture Timings		
	9.00 to 11.00	11.00 to 1.00	2.00 to 4.00
Monday	Data Structures	Software Engineering	Programing The Web
Tuesday	Operating System	Computer network	Computer organisation
Wednesday	Software Engineering	Computer network	Data Structures
Thursday	Programing The Web	Computer organisation	Data Structures
Friday	Operating System	Computer network	Programing The Web

Time Table

11. HTML program for Inline Style Sheet

InlineSS.html

```
<html>
 <head>
<title>Inline CSS Demo</title>
</STYLE>
</head
 <body>
<p STYLE="text-align:LEFT; font-size: 30; font-style: BOLD; margin-LEFT:100">INLINE CSS</p>
<h3 STYLE="color:RED;font-size: 40">Welcome to Inline Cascading Style Sheet Demonstration....</h3>
</body>
</html>
```

Output:

12. HTML program for Internal Style Sheet

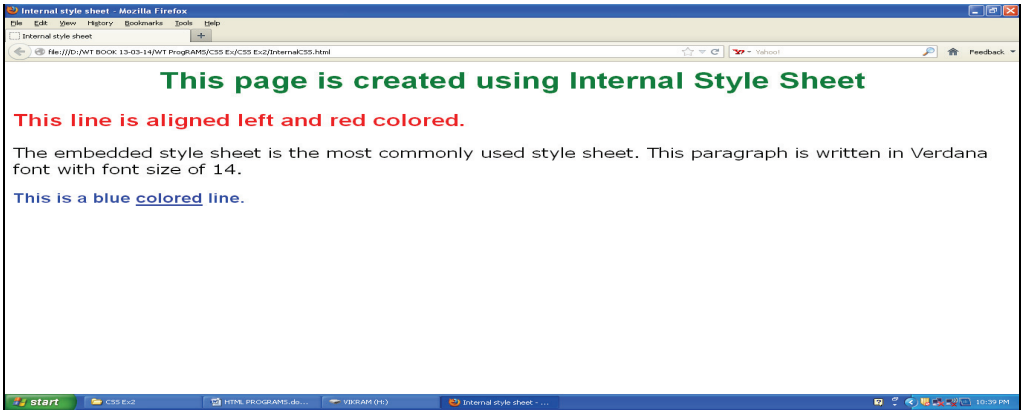
InternalSS.html

```
<html>
<head>
<title>Internal style sheet</title>
<style type="text/css">
```

```
        h1
        {
         font-family:Arial;
         color:green
        }
        h2
        {
         font-family:Arial;
         color:red;
         left:20px
        }
        h3
        {
         font-family:arial;
         color:blue;
        }
        p
        {
         font-size:14pt;
         font-family:verdana
        }
</style>
</head>
<body>
      <h1>
            <center>
            This page is created using Internal Style Sheet
            </center>
      </h1>
      <h2>
            This line is aligned left and red colored.
      </h2>
```

```
<p>
        The embedded style sheet is the most commonly used style sheet.
        This paragraph is written in Verdana font with font size of 14.
</p>
<h3>
This is a blue <a href=”colorname.html”>colored</a> line.
</h3>
</body>
</html>
```

Output:



13. HTML program for External Style Sheet

extSS.html

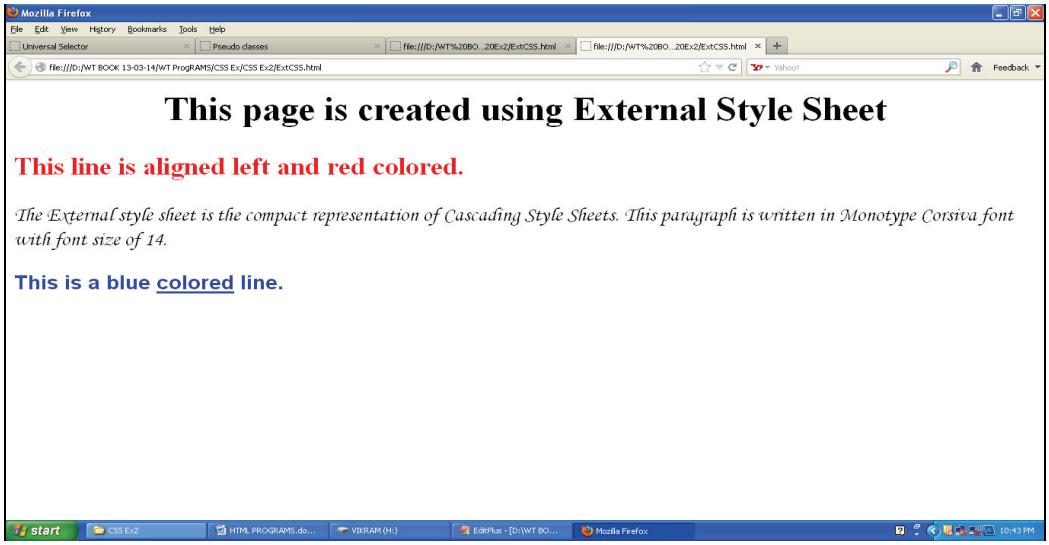
```
<html>
<head>
<link rel="stylesheet" type="text/css" href="ext.css" />
</head>
<body>
<h1 class="special"> <center> This page is created using External Style Sheet</center> </h1>
        <h2>
            This line is aligned left and red colored.
        </h2>
```

```
        <p>
        The External style sheet is the compact representation of Cascading Style Sheets.
        This paragraph is written in Monotype Corsiva font with font size of 14.
        </p>
        <h3>
            This is a blue <a href="colorname.html">colored</a> line.
        </h3>
    </body>
</html>
```

ext.css

```
h1
{
 font-family:Arial
}
h2
{
 font-family:times new roman;
 color:red;
left:20px
}
h3
{
font-family:arial;
color:blue;
}
p
{
 font-size:14pt;
 font-family:Monotype Corsiva
}
```

Output:



Review Questions

1. What is HTML? Explain Common HTML Tags in detail.
2. List and Explain types of Lists with an example.
3. What are Hyperlinks? How to create images as Hyperlinks in HTML with an example program.
4. How to insert images into an HTML Page? Explain in detail.
5. Describe in detail about <TABLE> tag & its sub tags with examples.
6. Discuss in detail about Frames.
7. What is <FORM> tag? Explain in detail about Form Field tags with examples.
8. Define CSS. Describe types of CSS in detail.
9. List out the rules for writing CSS.
10. Write the various approaches of CSS.
11. List out the properties of CSS.
12. Write the difference between Linked and Embedded Style Sheet.
13. Explain with suitable example how to display hyperlink on web browser. Can an image be used as a link?
14. What is list property? Explain the orde