**Web Technologies BS606**

**Theory 3 Hours/Week 3 credits**

**Practical 2 Hours/Week 1 credit**

**Unit – I**

**Structuring Documents for the Web**: Introducing HTML and XHTML, Basic Text Formatting, Presentational Elements, Phrase Elements, Lists, Editing Text, Core Elements and Attributes, Attribute Groups

**Links and Navigation:** Basic Links, Creating Links with the <a>Element, Advanced E- mail Links.

**Images, Audio, and Video:** Adding Images Using the <img>Element, Using Images as Links Image Maps, Choosing the Right Image Format, Adding Flash, Video and Audio to your web pages.

**Tables:** Introducing Tables, Grouping Section of a Table, Nested Tables, Accessing Tables **Forms:** Introducing Forms, Form Controls, Sending Form Data to the Server

**Frames:** Introducing Frameset, <frame> Element, Creating Links Between Frames, setting a Default Target Frame Using<base> Element, Nested Framesets, Inline or Floating Frames with <iframe>.

**Unit – II**

**Cascading Style Sheets:** Introducing CSS, where you can Add CSS Rules.

**CSS Properties**: Controlling Text, Text Formatting, Text Pseudo Classes, Selectors, Lengths, Introducing the Box Model.

**More Cascading Style Sheets**: Links, Lists, Tables, Outlines, the: focus and: activate Pseudo Classes Generated Content, Miscellaneous Properties, Additional Rules, Positioning and Layout with CSS

**Page Layout:** Understating the Site’s Audience, Page Size, Designing Pages, coding your Design, Developing for Mobile Devices.

**Design Issues**: Typography, Navigation, Tables, Forms.

**Unit – III**

**Learning JavaScript**: How to Add Script to Your Pages, the Document Object Model, Variables, Operators, Functions, Control Statements, Looping, Events, Built- In Objects, **Working with JavaScript:** Practical Tips for Writing Scripts, Form Validation, Form Enhancements, JavaScript Libraries.

**Putting Your site on the web**: Meta tags, Testing your site, Talking the Leap to Live, Telling the World about your site, Understanding your visitors.

**Text**  Jon Duckett, Beginning HTML, XTML, CSS and JavaScript

**References**  Chris Bates, Web Programming

M. Srinivasan, Web Technology: Theory and Practice

Achyut S. Godbole, Atul Kahate, Web Technologies

Kogent Learning Solutions Inc, Web Technologies Black Book

Ralph Moseley and M. T. Savaliya, Developing Web Applications

P.J. Deitel & H.M. Deitel, Internet and World Wide Web How to program

**UNIT-I**

**CHAPTER-I**

**Structuring Documents for the web**

**MARKUP LANGUAGES:**

Markup languages are the languages that build the **web pages.** Web page is a **web document** available on **World Wide Web** that is accessible through the internet using a web browser like **Internet Explorer**. A webpage can contain text, graphics, audio, video and animations. Through a markup language, you can mark up such contents to indicate how they should appear over a web page. To mark up the contents, you can use various commands, called **Markup Indicators** or **Tags.**

So, with a markup language, you simply “mark up” a text document with tags that tell a web browser how to structure it to display. Thus, markup language is a way of writing layout information within documents.

The most widely known markup languages today are **HTML**, **DHTML, XML** and **XHTML**.

**HISTORY:**

In 1980 physicist Tim Berners-Lee who was contractor at CERN (European Council for Nuclear Research) proposed and prototype ENQUIRE, a system for CERN researchers to use and share documents. In 1989, Berners Lee wrote a memo proposing an Internet based Hypertext system, Berners Lee specified HTML and wrote the browser and server software in the last part of 1990.In that year Berners-Lee and CERN data system Engineering Robert cailiau collaborated on a joint request for funding, but the project was not formally adopted by CERN. In these personal notes from 1990 he lists “some of the many areas in which hypertext is used” and puts on encyclopedia first.

The first publicly available description of HTML was a document called HTML Tags, first mentioned on the Internet by Berners-Lee in late 1991.It describes 20 elements comprising the initial, relatively simple design of HTML.

HTML is a text and image formatting language used by web browsers to dynamically format web pages.

Many of text elements are found in the 1988 ISO (International Organization Standardization) technical report TR 9537 Techniques for using SGML (Standard Generalized Markup Language) which in turn covers the features of early text formatting languages such as that used by RUNOFF command developed in the early 1960’s for the CTSS (Compatible Time-Sharing System).

**HTML VERSION:**

* HTML 1991
* HTML+ 1993
* HTML2.0 1995
* HTML3.2 1997
* HTML4.01 1999
* XHTML 2000
* HTML5 2012

**INTRODUCING HTML AND XHTML**:

The backbone of the World Wide Web is made of **HTML files,** which are specially-formatted documents that can contains links, as well as images and other media.

HTML stands for **H**yper **T**ext **M**arkup **L**anguage, and it is the most widely used language to design Web Pages.

A webpage is accessed by entering a **URL**(Uniform Resource Locator-a mechanism used to locate a webpage on the internet.) addresses in the **Address field** of the web browser. A webpage may contain text, graphics, and hyperlinks to other web pages and files.

A **website** is composed of a group of web pages linked together. It is a central location that contains more than one web page.

**HTML** (**H**yper **T**ext **M**arkup **L**anguage), as its name suggests, is a markup language.

* ***Hypertext*** refers to the way in which Web pages (HTML documents) are linked together. When you click a link in a web page, you are using hypertext. So, hypertext is simply piece of text that works as a link.
* ***Markup Language*** describes how HTML works. With a markup language, you simply “**mark up**” a text document with tags that tell a Web browser how to structure it to display. So, markup language is a way of writing layout information within documents.

HTML was created by Berners-Lee in late 1991 but “HTML2.0” was the first standard HTML specification which was published in 1995.

HTML 4.01 was a major version of HTML and it was published in late 1999.

Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in2012.

**XHTML** stands for e**X**tensible **H**yper**T**ext **M**arkup **L**anguage. It is the next step in the evolution of the Internet. XHTML was developed by the W3C to help web developers make the transition from HTML to XML. Developers who migrates their content to XHTML will realize the following benefits.

1. XHTML documents are XML conforming. As such, they, are readily viewed, edited, and validated with standard XML tools.
2. XHTML documents can be written to operate better than they did before in existing browsers as well as in new browsers.
3. XHTML documents can utilize applications like-scripts and applets that rely upon either the HTML Document Object Model or the XML Document Object Model.
4. XHTML syntax is very similar to HTML syntax and almost all the valid HTML elements are valid in XHTML as well. Some of the important points to remember while writing a new XHTML document or converting existing HTML document into XHTML document.
5. XHTML document must have a DOCTYPE declaration at the top of the document.
6. All XHTML tags and attributes should be written in lower case only.
7. All the XHTML tags will have their closing tags.
8. All the attribute values must be quoted.
9. Attribute minimization is forbidden.
10. All the tags should be properly nested.

XHTML gives you a more consistent, well- structured format so that your web pages can be easily parsed and processed by present and future web browsers. XHTML makes your website easier to maintain, edit, convert and format in the long run.

XHTML combines strength of HTML and XML and XHTML pages can be rendered by all XML enabled devices. In XHTML, it is relatively easy to introduce new elements or additional element attributes.

**HTML Vs XHTML**

|  |  |
| --- | --- |
| **HTML** | **XHTML** |
| HTML (**H**yper **T**ext **M**arkup **L**anguage) is the main markup language for developing web pages and other information that can be displayed. | XHTML (e**X**tensible **H**yper **T**ext **M**arkup **L**anguage) is an extended version of HTML that is XML-based. It combines strength of HTML and XML. |
| HTML is derived from SGML | XHTML is derived from HTML and XML. |
| HTML is not case-sensitive | XHTML is case-sensitive. |
| XHTML, being XML, must be well-formed. Every element must have an end tag, or use the self-closing tag syntax. | HTML allows some end tags to be omitted. |
| HTML allows some attribute to be minimized. | All attributes must have a value in XHTML. |
| HTML allows quotes to be omitted i.e. it is not necessary to quote each attribute values. | All attribute values must be surrounded by double or single quotes. |
| HTML is about displaying information. | XHTML is about describing the information. |

**XML**:

XML stands for e**X**tensible **M**arkup **L**anguage. XML is a markup language much like HTML. It was designed to describe data.XML tags are not predefined in XML. You must define your own tags.XML is self describing. It uses a DTD (Document Type Definition) to formally describe the data.

The tags used to make up HTML documents and the structure of HTML documents is predefined. The author of HTML documents can only use tags that are defined in the HTML standard.

XML allows the author to define his own tags and his own document structure.

**The Main Difference between XML and HTML**

* XML is not a replacement for HTML
* XML and HTML were designed with different goals:
* XML was designed to describe data and to focus on what data is.
* HTML was designed to display data and to focus on how data looks.
* HTML is about displaying information; XML is about describing information.

**WORLD WIDE WEB:**

WWW is the acronym for the World Wide Web. It is also commonly known as ‘The Web’. The WWW is the hypertext based information retrieval tool. One can easily surf the web by jumping from one document to another using the links in those documents. These documents can be in many formats such as text, images, graphics, animation, sound and video. They may also be a combination of all these. All the information on internet is presented to the user as a document. These documents are called as WebPages. All these web pages are linked to each other or even to section within a web page and these links are known as Hyperlinks.

**BASIC HTML:** HTML stands for **H**yper **T**ext **M**arkup **L**anguage.

**H**yper

**T**ext

**M**arkup

**L**anguage

* Hypertext is simply a piece of text that works as a link
* Markup Language is a way of writing layout information within documents.

**Introduction**: The language used to describe WebPages is called **HTML.** HTML is not a programming language, it a markup language. A markup language is a set of mark up tags. HTML uses Markup tags to describe web pages. When you save a HTML file, you can use either .htm or the .html extension. With the new software it is perfectly safe to use.html. It allows images and objects (**a thing that can be seen and touched, but it is not alive**) to be embedded and can be used to create interactive forms. It can embed scripts in language such as **Java script** which affect the behavior of HTML web pages.HTML can also be used to include **Cascading Style** **Sheet (CSS**) to define the appearance and layout of text and other material.

Html is a syntax used to format a text document on the web. HTML is the language interpreted by a browser. Such as i) Internet explorer ii) Netscape navigator etc. WebPages are also called html documents.

HTML is a markup language. It tells the web browser how to display content. HTML Separates “content” (words, images, audio, video, and so on) from “presentation”.HTML uses a predefined set of elements to identify content types.

**Tags:** Any formatted text document is composed of set element. Such as paragraphs, heading, lists. HTML formatting is specified by using tags. A tag is a format name surrounded by the angle brackets. End tags which switch a format off also contain a forward slash

**For example**: <h1>Text in an H1 style</h1>

**HTML Tags:** A web browser reads an HTML document top to bottom, left to right. Each time the browser finds a tag. it is displayed accordingly (paragraphs look like paragraphs, tables look like tables, etc).

Tags have 3 major parts: opening tag(s), content (s), and closing tag(s). Recall that a completed tag is termed an element.

There are probably hundreds of HTML tags, Tables, Images ,Lists, Forms, and everything else being displayed on an web page requires the use of a tag to two.

**HTML Code:**

< Opening tag>content</ Closing tag>

**Ex**:<p>A Paragraph Tag</p>

**Types of Tags:**

Tags are instructions that are embedded directly into the next of a HTML document. Each HTML tag describes that the browser should do something instead of simply displaying the text. In HTML, the tags begin with (<) and end with (>).

HTML tags can be of two types.

They are: 1. Paired Tags/Container Tags

2. Unpaired Tags (Singular Tags)/Empty Tags

**Paired Tags:**

A tag is said to be a paired tag if the text is placed between a tag and its companion tag .In paired tags, the **first tag referred to as Opening Tag** and the **second tag is referred to as Closing Tag.**

**Ex:** <i>This text is in italic.</i>

**Note:** Here <i> is called opening tag and </i> is called closing tag.

**Unpaired Tags:**

An unpaired tag does not have a companion tag. Unpaired tags are also known as Singular or Stand-Alone Tags.

**Example:**

<br>,<hr> etc

These tags do not require any companion tag.

**Note:** HTML tags are not case insensitive.

**A number of points should be noted about HTML Tags:**

* Tags are delimited by angled brackets:<h1>
* They are not case sensitive:<HEAD>,<head>,and <hEaD> are equivalents;
* Styles must be switch off by an end tag. There are a few exceptions to this rule, their differences will be noted in their descriptions;
* White space, tabs, and new lines are ignored by the browser; they can be used to make the HTML source more readable without affecting the way that the page is displayed. Actually they are not ignored, but multiple white spaces are replaced by a single space.
* If a browser doesn’t understand a tag it will usually ignore it.

**Elements**: Element give structure to a HTML document and tells the browser how you want your website to be presented. Generally, elements consist of a start tag, some content, and an end tag.

An element consists of **three** basic parts: an **opening** tag, the element’s content, and finally, **a closing** tag.

* **<p>**-opening paragraph words
* **Element Content**-paragraph words
* **</p>-**closing tag

Every(web)page requires four critical elements: the html, head, title, and body elements.

**The <html>Element….</html>:** <html>begins and ends each and every web page. Its sole purpose is to encapsulate all the HTML code and describe the HTML document to the web browser. Remember to close your HTML documents with the corresponding </html>tag at the bottom of the document.

**HTML Code**:

<html>

</html>

<P> WELCOME TO ALL OF YOU</P>

(Start Tag) (End Tag)

**Example**:

**The <head>Element…</head>:** The <head> element is “next”. As long as it falls somewhere between your <html> tag and your web page content (<body>). The head functions “behinds the scenes. “Tags placed within the head element are not directly displayed by web browsers.

Other elements used for scripting (JavaScript)and formatting(CSS) will eventually be introduced and you will have to place them within your head element. For now, your head element will continue to lay empty except for the title element that will be introduced next.

Here’s a sample of the initial set up.

**HTML Code:**

<html>

<head>

</head>

</html>

**The <title>Element….</title>:** Place the <title>tag within the <head> element to title your page. The words you write between the opening and closing <title></title>tags will be displayed at the top of a viewer’s browser. Here’s the html code:

**HTML Code:**

<html>

<head>

<title>My WebPage!</title>

</head>

</html>

**The <body>Element…</body>:** The <body> element is where all content is placed. (paragraphs, pictures, tables, etc). As the menu on the left suggests, we will be looking at each these elements. For now, it is only important to understand that the body element will encapsulate all of your webpage’s viewable content.

**HTML Code:**

<html>

<head>

<title>My Web Page!</title>

</head>

<body>

Hello World! All my content goes here!

</body>

</html>

**Steps to create a HTML document.**

1. Open a text editor(notepad) Start->run->notepad.

2. Type the html code in a blank text file.

3. Save the file by selecting the required drive and folder with .HTML or .HTM extension.

**Note:** Files are displayed with the default browser icon.

**Steps to execute the HTML document**.

Double click on the selected file to display the output in the default browser.

(OR)

1. Open the default browser application & select the file using file->open command.

2. Select browse and give the file name and destination->ok.

**Make the changes in notepad and press F5(Refresh) to display the changes in the browser.**

**Structure of HTML Program.**

Every HTML program has a rigid structure. The entire web page is enclosed within <HTML></HTML>tags. Within these tags two separate sections are created using the <HEAD></HEAD> tags and the <BODY></BODY> tags. These sections are described below.

<HTML> <html> tag indicates that this web page in written in HTML

**Document Head** <HEAD>

**Section** <TITLE>

Title pages <title> tag contains the wed page title

</TITLE>

</HEAD> <head> tag contains information about the web.

**Document Body <**BODY>

**Sectio**n Content <body> tag contains the content of the web page.

</BODY>

</HTML> </html> marks the end of the web page.

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

**Document Body**: This area is used to display the content of the webpage with required formats presented in the form of sub tags inside the start and the end of the main body of textual information<BODY></BODY>.The body tag is the second and main part of every html document. This is a paired element where starting and closing is mandatory. The data given in between these tag is displayed in the work area of the browser window.

By default, the background color of the body is displayed in white and the text color is indicated with black. In order to change the default properties and format the display structure of the <BODY> tag attributes are used.

**<!- - First Program in HTML - ->**

<html>

<head>

<title>Sample Program</title>

</head>

<body>

This is My First Web Page

</body>

</html>

**HTML Attributes:**

In HTML, elements (or tags) have attributes/properties. Attributes allow to add extra instruction to selected tag. Because each tag has its own unique attributes, any attribute cannot be just applied to any tag.

* HTML elements can have **attributes**.
* Attributes provide **additional information** about an element.
* Attributes are always specified **in the start tag**.
* Attributes are set of keyword with inbuilt functionality differs for each element.
* Attributes values should always be enclosed in quotes. “Values”.
* Attribute names and attribute values are case-insensitive.

Each element has its own set of attributes used to change the properties of the tag based on user choice and formats apart from this there are some common attributes implement for all the tags in HTML as

1. **Title**: Used to specify a tool tip text for selected element
2. **ID**: Used to specify and internal identification for the element.
3. **Style**: Used to specify dynamic style specifications.
4. **Class**: Used to call a specified class declared in the style sheet program.

**HTML Comments**:

Comments are piece of code which is ignored by any web browser. It is good practice to comment your code, especially in complex documents to indicate sections of a document, and any other notes to anyone looking at the code. Comments help you and others understand your code.

HTML comment lines are indicated by the special beginning tag<!—and ending tag-- >

Placed at the beginning and the end of every line to be treated as a comment.

**<BODY>:**

**Purpose:** Contains the body of the HTML document, which indicates the content that will actually appear in the web browser. The entire content of the web pages is placed in the page’s**<BODY>**element.

**Attributes:**

* **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. Ex:”#RRGGBB”,”#34FF78”.
* **BACKGROUND:**  Specifies the name of the GIF file that will be used as the background of the document. This GIF is used to give a background. Also the colors for background.
* **TEXT:** Changes the body text colors from its default value to the color specified with this attribute.
* **ALINK, VLINK, LINK:** Specifies the color of Active Link, Visited Link and the color for the link which is not visited. (**NOTE:** Alink, Vlink, are the attributes of body to change the color of another anchor tag text).
* **BGPROPERTIES:** Indicates the horizontal scroll bar for the web page values(Scroll/Fixed)
* **BOTTOM MARIN:** Specifies the empty space in the bottom of the web page.
* **MARGIN HEIGHT/WIDTH:** Gives the height and the width of the document towards right left top bottom.
* **RIGHT/LEFT MARGIN:** Margin distance form the border of the web page and the text given towards right and left any integer values can be given.
* **SCROLL:** Specifies the vertical scroll bar and the values are Yes/No.

**Example to implement body tag and its attributes.**

<html>

<head>

<title>Sample Program On Body Tag and Different Attributes</title>

</head>

<body bgcolor="blue" text="white">

WELCOME TO COMPUTER SCIENCE LAB

</body>

</html**>**

**BASCI TEXT FORMATTING:**

It is a process of changing the original state of the document such as text color background color using text formatting tags. Every document contains some type text that is represented by using the formatting elements such as

* **Heading Elements**:<H1>, <H2>, <H3>, <H4>, <H5>, <H6>
* **Paragraph Element**:<P>
* **Line Break Element**:<BR> or <BR/>
* **Pre-formatted Text Element**:<PRE>

**HEADINGS:**

**Purpose:**

Headings break a document into sections. Generally, a book uses headings and sun headings to divide each chapter into sections, and you can do the same with your web page.

Headings are important in HTML documents. Almost every document starts with a heading. You can use different sizes for your headings. HTML defines six levels of headings. The heading elements are H1,H2,H3,H4,H5, and H6 with H1 being the highest (or most important) level and H6 the least.

By default, most browsers use Times New Roman fonts for headings. The font size decreases as the heading level increases. (Default sizes for first through sixth-level headings are respectively,24,18,14,12,10 and 8-point font.)

**Note:** While displaying any heading, browsers automatically adds one line before and after that heading.

**Attributes:**

* **Align:** Specifies the alignment of text in the heading. Set to **LEFT**(the default),**CENTER,RIGHT Or JUSTIFY**.
* **Title:** Holds additional information (as might be displayed in tool tips) for the element.

**Example: This example demonstrates how to use headings.**

**<**html>

<head>

<title>Using Heading Tags</title>

</head>

<body>

<h1 align="right" title="Heading1">This is Heading One</h1>

<h2 align="center" title="Heading2">This is Heading Two</h2>

<h3 align="left" title=Heading3">This is Heading Three</h3>

<h4 >This is Heading Four</h4>

<h5>This is Heading Five</h5>

<h6>This is Heading Six</h6>

</body>

</html>

**PARAGRAPH:** Paragraphs are defined with the **<P>** tag

**Purpose**: Formats text into a paragraph and adds space before the paragraph. Author traditionally divide their thoughts and arguments into sequences of paragraphs. The organization of information into paragraphs is not affected by how the paragraphs are presented: paragraphs that are right-justified contain the same thoughts as those that are left-justified.

Paragraphs appear more often than any other text block in web pages. The HTML markup for defining a paragraph is straightforward. In HTML, paragraphs are defined with the <P> tag. Each paragraph of text should go in between an opening<P> and closing</p> tag.

**Note:** Browser automatically add an empty line before and after a paragraph.

**Attributes:**

* **Align**: Sets the alignment of the text in the paragraph. Set to **LEFT** (The Default) **RIGHT, CENTER, or JUSTIFY**.
* **Title**: Holds additional information (as might be displayed in tool tips) for the element.

**Example: This example demonstrates how to use Paragraph.**

<html>

<head>

<title>Paragraph Example</title>

</head>

<body>

<P>First Paragraph</P>

<P>Second Paragraph</P>

<P>Third Paragraph</P>

</body>

</html>

**LINE BREAK:**

**Purpose:** Break defines with **<BR> :** To break a line without starting a new paragraph in the webpage,<BR> tag is used. In other words, HTML <BR> or <BR/>tag is used to give a line break. Whenever you The <BR> tag, anything following it starts on the next line. The <br> tag is an empty tag. It has no closing tag.

**Attribute:**

* **Title:** Holds additional information (as might be displayed in tool tips) for the element.

**Example: This example demonstrates how to use <BR> Tag.**

<html>

<head>

<title>Break Tag Example</title>

</head>

<body>

Good Morning<BR>

Have a Nice Day<BR>

Sweet Dreams<BR>

</body>

</html>

**PRE-FORMATTED TEXT:** It defines with **<PRE></PRE>** This element display the text as it is given in the source with all formats (as spacing,tabulation etc).This is used when the tab setting to be printed in the same manner given in the source code of the web page.

**Purpose:** Tells the browser that the enclosed text is preformatted and shold not be reformatted

**Attributes**:

* **TITLE:** Holds additional information (as might be displayed in tool tips)for the element
* **WIDTH:** Sets the maximum number of characters per line.

**Example: This example demonstrates how to use <PRE>Tag**

<html>

<head>

<title>Using &lt;PRE&gt; Tag</title>

</head>

<body>

<pre>

Name Email Class

...................................................................................................

David david@gmail.com MSCs

Rahul rahulmala@gmail.com MSCs

Raju raju123@yahoo.com MSCs

</pre>

</body>

</html**>**

**PRESENTATIONAL ELEMENTS:**

In Html, presentational elements specify how content should look over a web page. Such elements affect only the presentation of the web document.

The following defines a list of presentational elements commonly used in HTML.

* **BOLD:** Displays text in BOLD FACE style. The tags used are**<B>………..</B>.**
* **ITALICS:** Displays text in ITALICS style. The tags used are**<I>……………</I>.**
* **<TT>:** Displays “teletype” text. The <TT>tag is used for displaying text in a fixed width font similar to that of a typewriter.
* **<BIG></BIG>:** Displays text larger than usual.
* **<SMALL></SMALL>:** Displays text smaller than usual.
* **<S>and <STRIKE>:** The text can be marked with a strike through character, place it within the **<S>………</S>**tags.
* **SUBSCRIPT:** The subscript tag**, <SUB>,**used to generate a subscript function which displays the text in lower position to the points specified by the used.

**EX:** H<SUB>2</SUB>0 **Output:** H2O

* **SUPERSCRIPT:** The superscript tag**, <SUP>,**used to generate a superscript function which display the text in raised position and set the vertical alignment of the text enclosed in **<SUP></SUP>**tag.

**EX**: AX<SUP>2</SUP>+BX+C=0 **Output**: AX2+BX+C=0.

* **UNDERLINE**: Display text as **UNDERLINE** style. The tags used are **<U>………</U>** tags.
* **FONT**: To define font size, color and face for a text.. The tags used are **,<FONT>…</FONT>**
* **CENTER:** To set text/image as center aligned. The tags used are**<CENTER>…..</CENTER>.**
* **DIV**: To set alignment of text/image as left/center/right. The tags used are **<DIV>…</DIV>.**
* **HORIZONTAL RULE:** To create a horizontal line. This is also called Horizontal Rule in HTML. The Tags used are <HR/>

**HORIZONTAL RULE**:

**Purpose**: The tag<HR> draws lines and horizontal rules. This tag draws a horizontal line across the whole page, whenever specified. The attributes to the **<HR>** tag are:

**Attributes**:

* **ALIGN**: Align the on the browser screen, which is by default, aligned to the center of the screen. The possible values are left, right, center.
* **WIDTH**: Sets the width of the rule. It can be set to a fixed number of pixels or to a percentage of the available screen width.
* **SIZE**: Changes the size of the rule.
* **COLOR**: Used to fill the color inside the ruler.

**Example: This example demonstrates how to use <HR>Tag**

<html>

<head>

<title>Using <HR>Tag</title>

</head>

<body>

Here is what &ltHR&gt;looks like:

<hr>

<br>

Here is what &ltHR align=left width=80 color=red%&gt; looks like:

<hr align=left width=80% color="red">

<br>

here is what &ltHR align=center width=80% color=pink&gt; looks like:

<hr align=center width=80% size=10 color="pink">

<br>

here is what &ltHR align=right width=80% color=green&gt; looks like:

<hr align=right width=80% color=green>

<br>

here is &ltHR aligh=center size=10&gt; looks like:

<hr align=center size=10 color="brown">

<br>

here is &ltHR align=center size=10 noshade&gt; looks like:

<hr align=center size=10 noshade>

</body>

</html>

**<CENTER> TAG**:

**Purpose:** The text in the webpage is brought to the center using **<CENTER>** tag.

**Attribute**:

* **TITLE**: Holds additional information (as might be displayed in tool tips)for the element

**Example: This example demonstrates how to use <CENTER>Tag**

<html>

<head>

<title>Using the &lt;CENTER&gt; tag</title>

</head>

<body>

<center title="CenterTag">

As you can see,

<br>

the &lt;CENTER&gt;tag

<br>

can center multi-line text.

</center>

</body>

</html>

**FONTS:**

Texts are typically displayed with a standard font(Times New Roman under Windows, Helvetica under Macintosh),in medium size, and in black colour.

**Purpose:** The **<FONT>** tag allows you to change the font and can take up to three attributes. **Attributes:**

* **SIZE**: Size defines the text size. It can be any value between 1 and 7.The actual size depends on the browser.
* **COLOR**: Color defines the color of the text enclosed between the **<FONT>** and**</FONT**> tags. The color is given as a three-hexadecimal-value code in the following form #RRGGBB, for the red, green and blue values respectively.
* **FACE:**. Face defines the name of the font to be used. It is possible to indicate several names, in which case the browser will use the first font in finds. You can combine all these attributes to get original formatting
* **TITLE:** Holds additional information(as might be displayed in tool tips)for the element.

**The following table shows the Font Attributes**.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Example** | **Purpose** |
| Size=”number” | Size=”2” | Defines the font size |
| Size=”+number” | Size=”+1” | Increases the font size |
| Size=”-number” | Size=”-1” | Decreases the font size |
| Face=”face-name” | Face=”times new roman” | Defines the font-name |
| Color=”color-value” | Color=”#eeff00” | Defines the font color |
| Color=”color-name” | Color=”red” | Defines the font color |

**Example: This example demonstrate how to set font and font size.**

<html>

<head>

<title>Using <Font>Tag</title>

</head>

<body bgcolor="pink">

<font size="7" face="Vladimir script" color="white" title="using Font Tag">

This is Font tag using text.

</font>

</body>

</html>

**Example: This example demonstrates how to use Presentational Elements.**

<html>

<head>

<title>Presentational Elements</title>

</head>

<body>

This is <B> Bold</B><br>

This is <i> Italic</i><br>

This is <U> Underline</U><br>

<Strike>Text-Crossed</strike><br>

This is <Big> Big Font</Big><br>

This is <Small> Small Font</small><br>

H<SUB>2</SUB>O<br>

December 25<sup>th</sup>2019<br>

<TT>Fixed- Width Font</TT><br>

</body>

</html>

**PHRASE ELEMENTS:**

In HTML, phrase elements add meaning to fragments of text. Like Presentational elements(<B>,<U>,<I> etc.),phrase elements are not just for presentational purposes; they also describe something about their content.

For Example, the words written in an <EM> element will look just like the words in an <I> element, but the <EM> element is supposed to indicate the addition of emphasis. Similarly, using the <ABBR>element indicates that the word or phrase contained within is an abbreviation or acronym.

In general, phrase elements are more useful when it is necessary to define a particular kind of data within a document, rather than for display purposes.

The following is the list of Phrase elements commonly used in HTML.

|  |  |
| --- | --- |
| **Element** | **Description** |
| <EM>…….</EM> | To Emphasized Text.(Same as <I>) |
| <STRONG>…….</STRONG> | To define important Text.(Same as <B>) |
| <ABBR>……….</ABBR> | To define an abbreviation or an acronym, like-“Mr”,”Dec.”,”CPU” |
| <DFN>……….</DFN> | To mark the term as being defined ,but the actual definition should be given by the surrounding<P> element. |
| <BLOCKQUOTE>……..</BLOCKQUOTE> | To define a section that is quoted from another source. It is used for indicating long quotations i.e. quotations that span multiple lines. Text inside a <BLOCKQUOTE> element is usually indented from the left and right edges of the surrounding text. |
| <CITE> …………..</CITE> | To specifies a citation. It is used to define the title of a work. Like title of a book, title of a movie etc. |
| <CODE>………..</CODE> | To displays some characters as code usually in courier font (i.e., fixed-width font) |
| <ADDRESS>……….</ADDRESS> | To define the contact information for the author/owner of a document or an article. |
| <KBD>……………</KBD> | To define that a section of text is keyboard input. For Example, if you want to tell a reader to enter some text, you can use the <KBD>….</KBD> element to indicate what should be typed in. |
| <VAR>……..</VAR> | To define a variable. It is useful when you are writing a page that deals with programming or mathematical expressions. |
| <SAMP>…………</SAMP> | To define sample output from a computer or script. |

**<!—using phrase element-->**

<html>

<head>

<title>Phrase Example</title>

</head>

<body>

This is <em>Emphasized</em><br>

This is <strong>Strong</strong><br>

The <abbr title="Central Processing Unit">CPU</abbr>is the brain of the computer.

<p><DFN>Computer</DFN>is an electronic device used to process the data.

</p>

<blockquote>

Internet is a global "network of networks". It can also be described as information highway. It is the world's largest network of computers. The networks are related to academics, research, government and include users of all types of profile i.e., corporate, individuals, professionals, children, schools, institutes etc.</blockquote>

<p><cite>Web Technologies</cite>by Tem Berners Lee in 1990's</p>

<code>E=MC<sup>2</sup></code>

<address>#303,Street No.22,Banl Colony-New Delhi.</address>

<p>Regular text.<KBD>This is inside KBD element</KBD>Regular text.</p>

<p>A regular expression that has a <var>variable</var>contained within it.</p>

<p>Result produced by the program is <samp>Hello Everyone</samp></p>

</body>

</html>

**ADDRESS:**

Address specifies information such as authorship and contact details for the current document. Browsers should render the content with paragraph-breaks before and after the **<ADDRESS**.tag.(Displays text styled as an address)

**<ADDRESS>……..</ADDRESS>**

**Example: This example demonstrates how to use address**

<html>

<head>

<title>Using Address Tag</title>

</head>

<body>

<address>Donald Duck<br>

BOX555<br>

Disneyland<br>

USA

</address>

</body>

</html>

**Character Entities:**

Some characters have a special meaning in HTML, like the less than sign(<)that defines the start of HTML tag. If we want the browser to actually to display these characters we must insert character entities in the HTML source.

A character entity has three parts-an ampersand (&),an entity name or a # and an entity number, and finally a semicolon(;).To display a less than sign in HTML document we must write-&it; or &#60;

The advantage of using a name instead of a number is that a name is easier to remember.

The disadvantage is that not all browsers support the newest entity names.

The support for entity numbers is very good in almost all browsers

Entities are case sensitive.

**Non-Breaking Space**:

The most common character entity in HTML is the non breaking space.Normally HTML will truncate space in the text. It you write 10 spaces in your text HTML will remove 9 of them.To add spaces to your text, use the &nbsp; character entity.

**Table: Character Entities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Result** | **Description** | **Entity Name** | **Entity Number** |
|  | **Non-breaking space** | **&nbsp;** | **&#160;** |
| **<** | **Less than** | **&it;** | **&#60;** |
| **>** | **Greater than** | **&gt;** | **&#62;** |
| **&** | **Ampersand** | **&amp** | **&#38;** |
| **“** | **Quotatiton mark** | **&quot;** | **&#34;** |
| **‘** | **Apostrophe** | **&apos;** | **&#39;** |

**Example: This example demonstrates how to use character entities.**

<html>

</head>

<title>Using of character entities</title>

</head>

<body bgcolor="cyan">

Some &nbsp;&nbsp;character&nbsp;&nbsp;&nbsp;&nbsp;have a

&nbsp;&nbsp;&nbsp;&nbsp;special&nbsp;&nbsp;&nbsp;&nbsp;in HTML<br>

&quot;The most common character entity in HTML is non-breaking space &amp;HTML

will truncate spaces in the text &quot;

</body>

</html**>**

**LISTS**:

List is the process of displaying the data in bullet/numbering format.

HTML support with 3 types of lists as Order list, Un Order list and Definition list. Use the type and start attributes to fine turn your lists accordingly.

1. **Ordered Lists:** An ordered list is also a list of items. The list items are marked with numbers. An ordered list starts with the **<ol>** tag. Each list item starts with the **<li>**tag. Ordered simply means numbered, as the list below demonstrates.

**Sample program on an Ordered List which displaying list of items in default number**

<html>

<head>

<title>Using &li;ol&gt;tag</title>

</head>

<body>

<h2>An ordered HTML list</h2>

<ol>

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ol>

</body>

</html>

**Ordered HTML List - The Type Attribute**

The **type** attribute of the <ol> tag, defines the type of the list item marker:

|  |  |
| --- | --- |
| **Type** | **Description** |
| type="1" | The list items will be numbered with numbers (default) |
| type="A" | The list items will be numbered with uppercase letters |
| type="a" | The list items will be numbered with lowercase letters |
| type="I" | The list items will be numbered with uppercase roman numbers |
| type="i" | The list items will be numbered with lowercase roman numbers |
|  |  |

**Sample program on Ordered List using type attribute**

<html>

<head>

<title>Using type attribute in List</title>

</head>

<body>

<h2>Ordered List with Letters</h2>

<ol type="A">

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ol>

</body>

</html>

**Unordered Lists**: An Unordered list is a list of items. The items in the list are marked with bullets,An unordered list starts with the **<UL>** tag. Each list item starts with the **<LI>** tag. The bullet itself comes in three flavors: squares, discs, and circles. The default bullet displayed by most web browsers in the traditional full disc.

**Sample program on an Unordered List which displaying list of items in default bullet**

<html>

<head>

<title>Using &lt;UL&gt; Tag</title>

</head>

<body>

<h2>An unordered HTML list</h2>

<ul>

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ul>

</body>

</html>

**Unordered HTML List - Choose List Item Marker**

:

|  |  |
| --- | --- |
| **Value** | **Description** |
| disc | Sets the list item marker to a bullet (default) |
| circle | Sets the list item marker to a circle |
| square | Sets the list item marker to a square |
| none | The list items will not be marked |

**Sample program on an UnOrdered List using type attribute**

<html>

<head>

<title>Using Type attribute in an Unordered List</title>

</head>

<body>

<h2>Unordered List with Disc Bullets</h2>

<ul type=”disc">

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ul>

</body>

</html>

**Definition List**: A definition list is not a list of items. This is a list of terms and explanation of the terms. A definition list starts with the **<dl>**tag. Each definition list term starts with the **<dt>** tag, Each definition list definition starts with the **<dd>**tag. These lists displace the term word just above the definition itself for a unique look.

* **<dl>:Defines the start of the list.**
* **<dt>Definition term.**
* **<dd>Definition Describe**

**Sample program on Definition List**

<html>

<head>

<title>Using Definition List</title>

</head>

<body>

<h2>A Description List</h2>

<dl>

<dt>HTML</dt>

<dd>Hypertext Markup Language</dd>

<dt>DBMS</dt>

<dd>DataBase Mnagement System</dd>

</dl>

</body>

</html>

**EDITING TEXT:**

When working on a document with others, it helps if you can see changes that another person has made. Even when working on your own documents, it can be helpful to keep track of changes you make. There are two elements specifically designed for revising and editing text.

1. The **<INS>** element for when you want to add text(usually shown underlined in a browser).
2. The **<DEL>**element for when you want to delete some text(usually shown crossed out in a browser).

<html>

<head>

<title>Editing Text Example</title>

</head>

<body>

<p>My Favorite Color Is<DEL>Red</DEL><INS>Blue</INS></p>

</body>

</html>

**CORE ELEMENTS AND ATTRIBUTES/DOCUMENT TAGS**:

There are four main elements every HTML document should have. They define the overall structure of an HTML document. These elements define the major sections of the HTML document.

1. <HTML>
2. <HEAD>
3. <TITLE>
4. <BODY>

These core elements are also known as **Document tags** in HTML.

**i. <HTML>:** The <HTML> element is the main container or containing element for the whole HTML document. It represents the root of an HTML document. It simply tells the browser that this is an HTML document. Every HTML document should have one <HTML> tag and each document should end with a closing</HTML> tag.

Following two elements appear as direct children of an <HTML> element:

* <HEAD>
* <BODY>

**ii. <HEAD>:** The <HEAD> element is just a container for all other header elements. It simply defines the header section of the HTML document. It should be the first thing to appear after the opening <HTML> tag. Each <HEAD> element should container a <TITLE> tag indicating the title of the document.

Although it may also contain any combination of the following tags, in any order:

1. **<LINK>:** The <LINK> tag is used to link to an external file, such as a style sheet or JavaScript file.
2. **<STYLE>:** The <STYLE> tag is used to include CSS rules inside the HTML document.
3. **<SCRIPT>:** The <SCRIPT> tag is used to include JavaScript or VBScript code inside the HTML document.
4. **<META>:** The <META> tag includes information about the document such as keywords and a description, which are particularly helpful for research applications.

**iii. <TITLE>:** The <TITLE> element is used to set a title for every HTML document. It defines a title in the browser’s title bar. This element is a child of the <HEAD> element. The <TITLE> element should contain only the text for the title and it may not contain any other elements.

**iv <BODY>:** The <BODY> element appears after the <HEAD> element. It defines the body section of HTML document i.e. it simply defines the document’s body. The <BODY>element contain all the contents of an HTML document, such as text, hyperlinks, images, tables, lists, etc.

**Attributes:**

* **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. Ex:”#RRGGBB”,”#34FF78”.
* **BACKGROUND:**  Specifies the name of the GIF file that will be used as the background of the document. This GIF is used to give a background. Also the colors for background.
* **TEXT:** Changes the body text colors from its default value to the color specified with this attribute.
* **ALINK, VLINK,LINK:** Specifies the color of Active Link ,Visited Link and the color for the link which is not visited.(**NOTE:** Alink, Vlink,…are the attributes of body to change the color of another anchor tag text).
* **BGPROPERTIES:** Indicates the horizontal scroll bar for the web page values(Scroll/Fixed)
* **BOTTOM MARIN:** Specifies the empty space in the bottom of the web page.
* **MARGIN HEIGHT?WIDTH:** Gives the height and the width of the document towards right left top bottom.
* **RIGHT/LEFT MARGIN:** Margin distance form the border of the web page and the text given towards right and left any integer values can be given.
* **SCROLL:** Specifies the vertical scroll bar and the values are Yes/No.

**Example to implement body tag and its attributes.**

<html>

<head>

<title>Sample Program On Body Tag and Different Attributes</title>

</head>

<body bgcolor="blue" text="white">

HAVE A NICE DAY.

</body>

</html**>**

**ATRRIBUTE GROUPS:** In HTML, attributes are placed in the opening tag of an element and provide some extra information about the element that carries them. All attributes consists of a name and a value; the name reflects a property of the element the attribute is describing, and the value is a value for that property.

**For Example**: <BODY TEXT=”BLUE”>

In the above example, <BODY> element carries an attributes, whose name is TEXT, which you can use to set the color of the text in a web document. Here, the value set to TEXT attribute is “BLUE”. This means the color of the text in a web document will be BLUE.

There are mainly two groups of attributes that many of the HTML elements can carry:

1. **Core Attributes**: The four core attributes that can be used on the majority of HTML elements are:
2. id
3. title
4. class
5. style
6. The **id** Attribute of an HTML tag can be used to uniquely identify any element within an HTML page. There are two primary reasons that you might want to use an **id** attribute on an element:

If an element carries an id attributes as a unique identifier, it is possible to identify just that element and its content.

If you have two elements of the same name within a web, you can use the id attribute to distinguish between elements that have the same name.

**Example**: <p id=”HTML”>This paragraph explains what is HTML</p>

<p id=”XHTML”>This paragraph explains what is XHTML</p>

1. **The title Attribute**:

The title attribute gives a suggested title for the element.The syntax for the title attribute is as follows

title=”string”

The behavior of this attribute will depend upon the element that carries it , although it is often displayed as a tooltip when cursor comes over the element or while the element is loading.

**Example**:

<html>

<head>

<title>The title attribute</title>

</head>

<body>

<h3 title="Hello Everyone">Titled Heading Tag</h3>

</body>

</html>

1. **The class Attribute**:

The **class** attribute is used to specify that an element belongs to a class of elements.

**For Example**: You might have a document that contains many paragraphs, and a few of those paragraphs might contain a summary of key points ,in which case you could add a class attribute whose value is summary to the relevant<P> element, to differentiate those paragraphs from the rest in the document.

<P class=”summary”>Summary goes here</P>

The **syntax** of the class attribute is as follows:

**class=”classname”**

The value of the class attribute may also be a space-separated list of class names.

**For Example**: class=”className1 className2 className3”

The class attribute is commonly used with CSS, so you will learn more about the use of the class attribute when you will learn Cascading StyleSheet(CSS).

1. **The Style Attribute**:

The style attribute allows you to specify Cascading Style Sheet (CSS) rules within the element.

<html>

<head>

<title>The Style Attribute</title>

</head>

<body>

<p style="font-family: Bold; color: pink;">Have a nice day</p>

</body>

</html>

1. **Internationalization Attributes:**

There are three internationalization attributes that help users write pages for different languages and character sets, and they are available to most HTML/XHTML elements.

1. dir
2. lang
3. xml:lang
4. **The dir Attribute**

The dir attribute allows you to indicate to the browser the direction in which the text should flow; left to right or right to left. When you want to indicate the directionality of a whole document (or the majority of the document),it should be used with the <HTML> element rather than the <BODY> element for two reasons:

* Its use on the <HTML> element has better support in web browsers.
* It will apply to the header elements as well as to the body elements.

The dir attribute can also be used on elements within the body of the document if you want to change the direction of a small portion of the document.

The dir attribute can take one of two values, as you can see in the following table

|  |  |
| --- | --- |
| **Value** | **Meaning** |
| ltr | Left to right(the default value) |
| rtl | Right to left |

<html>

<head>

<title>Display Directions</title>

</head>

<body>

Left-To-Right directed text

<p dir="rtl">Right-To-Left directed text</p>

</body>

</html>

1. **The lang Attribute**:

The **lang** attribute allows you to indicate the main language used in a document. The lang attribute was designed to offer language-specific display to users, although it has little effect in main browsers. The real benefit of using the lang attribute is with search engine(s), which can tell the user about the language the document is authored in.

When used with the <html>element, the lang attribute applies to the whole document, although it can be used on other elements, in that case it just applies to the content of those elements.

The value of the lang attribute are ISO-639 standard two-character language codes. The following table displays a list of language codes for most of the main language in use today.

|  |  |
| --- | --- |
| **Language Code** | **Meaning** |
| ar | Arabic |
| en | English |
| en-us | U.S.English |
| zh | Chinese |

<html>

<head>

<title>English Language Page</title>

</head>

<body>

This page is using English Language

</body>

</html>

1. **The xml:lang Attribute**: The **xml:lang** attribute is the XHTML replacement for the lang attribute. It is an attribute that is available in all languages that are written in XML(eXtensible Markup Language), that’s why it is prefixed by the characters **xml:**

The value of the xml:lang attribute should be an ISO-639 country code like those listed in the previous section.

**UNIT-I**

**Chapter II**

**LINKS AND NAVIGATION**

**Hyperlinks in html:**

Most important concept of HTML programming is Hyperlinks. Web pages can contain links that takes directly to other pages and even specific parts of a given page. These links are known as hyperlinks. Hyperlinks allow visitors to navigate between Web sites by clicking on words, phrases, and images. Thus one can create hyperlinks using text or images available on any web page using html hyperlinks. Not only creating static web page but connecting multiple pages with the clickable hyperlinked text is considered as hyperlinks. A Anchor tag(<a>…. </a>is used to establish the hyperlinks in HTML.

A hyperlink (or link) is a word, group of words, or image that you can click on to jump to a new document or a new section within the current document. When you move the cursor over a link in a web page, the arrow will turn into a little hand.

In most internet documents, ” unvisited” hyperlinks will be blue in color. After the hyperlink has been activated or “visited”, its color should be changed to purple. On most browsers, these are the two default colors of “unvisited” and “visited” hyperlinks.

A Anchor tag(<a>……</a>)is used to establish the hyperlinks in HTML.

**Creating Links with the <A> Element /Anchor Tag:** A link is specified using the <a>element. This element is called anchor tag as well. Anything between the opening<a>tag and the closing</a>tag becomes part of the link and a user can click that part to reach to the linked document which is the hypertext. The term “anchor” is used because it indicates the static positioning of a hyperlink.

**Simple Syntax**: <a href=”document URL” attr\_name=”attr\_value”…more attributes>Link text</a>

Here, the **href** attribute specifies the destination of a link. The “Link text” doesn’t have to be text. It can be an image or any other HTML element.

**URL** stands for **“Uniform Resource Locator”**. A URL is the location of a specific website or file on the internet. It is a specific character string that constitutes o reference to a resource on the Internet.

**For Example:**

In the following URL:

<http://www.yahoo.com/docs/index.html>

* + - * **http:**communication protocol.
      * [**www.yahoo.com:** host](http://www.yahoo.com:%20host) name.
      * /**docs/index.html:** path and file name for the resource to be located.

**Note**: When you move the cursor over a link in a web page, the arrow will turn into little hand.

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

* An unvisited link is underlined and blue.
* A visited links is underlined and purple.
* An active link is underlined and red.

**<A> Attributes**:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| href | URL | Specifies the URL of the page the link goes to. |
| name | Section\_name | Specifies the name of an anchor or section |
| target | \_blank  \_parent  \_self  \_top  Framename | Specifies where to open the linked document. |
| shape | Default  Rect  Circle  Poly | Specifies the shape of a link.(Not supported in HTML5.) |
| cords | Coordinates | Specifies the coordinates of a link.(Not suppoted in HTML5) |

**Types of Paths**:

**Paths:** Path is the location of the file which is linkedto the text given between

<a></a>element. The location may be local or external local address can be specified using absolute/relative reference.

**Relative Path URL/Internal path)**

A relative path hyperlink uses a kind of shortand to specify a URL for a resource you are pointing to.

**Use the following guidelines with relative links in your HTML pages**.:

* Create relative links between resources in the same domain (web server).
* Because both resources are in the same domain, you may skip domain information from the URL.

A relative link is similar to telling someone that he or she needs to go the Western Mall. If the person already knows where the Western Mall is, he or she doesn’t need additional directions. Web browsers behave the same way in case or relative link.

If you use relative links on your site, your links still work if you change servers or Domain names. You can take advantage of relative URLs when you create a link between pages on the same website.

**For example**: If you want to make a link from <http://www.rimtpb.org/home.html> to <http://www.rimptb.org/about.html,you> can use this simplified relative URL in an anchor element on home.html:

know more <a href=”about.html”>About</a><RIMT.

When a browser sees a link without a domain name, the browser assumes that the link is relative and uses the domain and path from the linking page to find the linked page. The above example works only if home.html and about.html are in the same directory.

As your site grows more complex and you organize your files into various folders, you can still use relative links. However, you must provide additional information in the relative URL to help the browser find files that don’t reside in the same directory as the file from which you are linking.

Use ../(two periods and a slash) before the filename to indicate that the browser should move up one level in the directory structure.

The markup for this directory navigation process looks like this:

<a href=”../docs/home.html”>Documentation Home Page </a>

**The notation in this anchor element instructs the browser to take these steps**:

1. Move up one folder from the folder the linking document is stored in.
2. Fins a folder called docs.
3. Inside that folder, find a file called home.html.

**Format**:<a href=”contents.html”>Click hear to view the content</a>This is the hours page.

**Note**: As the above link is given in the file stored in same folder relative reference is used

**<!---Sample program on Relative Path-->**

<html>

<head>

<title>Relative Path Example</title>

</head>

<body>

<h2>The following text is a relative link to page on the same website.</h2>

<a href=”index.html”>Home Page</a>

</body>

</html>

**Absolute Path URL/External path**:

An absolute hyperlink uses a complete URL to connect browsers to a web page or online resource. Links that use a complete URL to point to a resource are called absolute because they provide a complete, stand-alone path to another web resource. When you link to a page on someone else’s website, the web browser needs every bit of information in the URL to find that page. The browser starts with the domain in the URL and works its way through the path to a specific file.

**Format**:<a href=”path/filename”>Text>/a>

**Ex:** <a href=”c:\My-site\support\contents.html”>click here to view the content</a>

This is catalog page.

**Note:** As the contents.html is not stored directly in products folder absolute reference is implemented.

**<!---Sample of on using absolute path-->**

<html>

<head>

<title>Absolute Path Example</title>

</head>

<body>

<h2>The following text is an absolute link to a page on the World Wide Web</h2>

<a href=<http://www.microsoft.com/>>

MICROSOFT

</a>

</body>

</html>

**Uniform Resource Locator(URL)**:

URL is an address/file specifier. It locates the documents requested through hyperlinks. Similarly, it may be used to point to a query, image or a command.

The **URL**(Uniform Resource Locator)is the global address of documents and the other resources on the web e.g .http://www.who.int.The first part of the address indicates which protocol to use e.g. http. The second part of the address identifies the domain name or the internet address where the information is located.

**Format**: Scheme:/host-domain/path/dataname

[Http://www.openuniversity.ng/admission/index.html](http://www.openuniversity.ng/admission/index.html)

**Attributes**:

* **Href**: Specifies the URL of the target of a hyperlink. Its value is any valid document URL, absolute or relative, including a fragment identifier or a JavaScript code fragment.
* **Target**: Specify where to display the contents of a selected hyperlink. If set to “-blank” then a new window will be opened to display the loaded page, if set to “-top” or”-parent” then same window will be used to display the loaded document, if set to “\_self” then loads the new page in current window. By default its “\_self”.
* **Name&id**: Attributes places a label within a document. When that label is used in a link to that document, it is the equivalent of telling the browser to goto that label.
* **Event**: Attributes like on Click, onMouseOver etc. are used to trigger any Javascript to VBscript code.
* **Title**: The title attribute holds text that will be displayed in tool tips when the mouse resets over the hyperlink. Set to an alphanumeric string.
* **Accesskey**: Attribute provides a keyboard shortcut that can be used to active a link. For Example, you could make the T key an access key so that when the user presses either the alt or ctrl key on this keyboard(depending on his operating system)along with the T key, link gets activated.

To change the default colors of the hypertext as link, active links and visited links text the link, alink and vlink attributes of <body>tag are used. But it is recommended to use CSS to set colors of links, visited links and active links.

**Using body attributes:**

<body alink=”#ff0000” link=”#900B09” vlink=”#900B09”>

**TYPES OF HYPERLINKS**:

Hyper link is the process of linking multiple documents/pages to a selected text/image these hyperlinks are implemented in multiple methods as linking to a selected page using a text or image, or connecting to the default mail box to send a email to the selected person, or to create internal bookmarks in the selected webpage.

Some of these links are

* Text hyperlink
* Image hyperlink
* Mailto

**Creating Text Hyperlinks**:

The standard form of a hyperlink element looks like this:

Find the answers at<a href=<http://w3c.org>>W3C</a>

Use the <a> tag to set up a hyperlink and using the HREF attribute to set the target of the hyperlink to http://W3C.org (which is the home page for the world wide web consortium).This is the URL that the browser will navigate to when the user clicks the link. `The text inside the <a> element will appear in the hyperlink style for the page, which is underlined and usually in a different color from the surrounding text like this.

Find the answers at W3C.

All the user has to do is to click the W3C hyperlink to be transported to the W3C Web site.

**Here’s a full Example, also linking to W3C**.

<html>

<head>

<title>Using &lt;TextHyperlinks&gt;</title>

</head>

<body>

<center>

<h1>Creating a Hyperlink</h1>

Here is a Web site to check out:

<a href="http:\\w3c.org">W3C</a>

</center>

</body>

</html>

**Creating Image Hyperlinks**: You can make images into hyperlinks using the **<IMG>** element inside an <A> element, which can spice up your Web pages .

**Sample Program Using Image Hyper link**

<html>

<head>

<title>Using &lt;ImageHyperlink&gt;</title>

</head>

<body>

<center>

<h1>Creating A Graphical Hyperlink</h1>

<a href="http://w3c.org">

<img width=277 height=114 src="C:\Users\Public\Pictures\Sample Pictures\Penguins.jpg" alt="click me to go to W3C!">

</a>

</center>

</body>

</html>

**Creating Email Links:**

Creating an email link is simple. If you want somebody to mail you about your site a good way to do it is place an email link with a subject already in place for them.

**HTML Code**:

<a href=”maito:email@tizag.com?subject=Feedback”>[Email@tizag.com</a](about:blank)>

**Email Links:**

[Email@tizag.com](mailto:Email@tizag.com)

**<!---Sample Program Using Maito/Email Hyper link-->**

<html>

<head>

<title>Using &lt;EmailHyperlink&gt;</title>

</head>

<body>

<center>

<h1>Creating A Email Hyperlink</h1>

<a href=”maito:email@tizag.com?subject=Feedback”>Email@tizag.com</a>

</center>

</body>

</html>

**CUTTOMIZING LINKS**:

In HTML, You can customize lint to:

* Open linked documents in new windows.
* Specific locations within the same webpage.
* A spot on a different web page within the same website.
* A spot on a different web page on other websites.
* Non-HTML pages, such as Portable Document Format (PDF) files, compressed files, Word processing documents etc.

**Open Linked documents in New Windows:**

The web works because you can link pages on your website to pages on other people’s websites by using a simple anchor element(<A>).When you link to someone else’s site, though, you send users away from your own site.

To keep users on your site ,HTML can open the linked page in a new window or in a new tab inside the same browser window. The simple addition of the target attribute to an anchor element opens that link in a new window or in anew tab instead of opening it in the current window.

<html>

<head>

<title>Open Links in new windows</title>

</head>

<body>

<h2>The following link will be explain in a new window or in a new tab inside the same browser window.

</h2>

<a href="http://www.microsoft.com/" target="\_blank">

MICROSOFT

</a>

</body>

</html>

In the above example, the use of target attribute in <A> element (target=”\_blank”) will simply tell the browser to do the following:

* Keep the linking page open in the current window.
* Open the linked page in a new window or tab.

**Link to Specific Locations within the same webpage**

In HTML, Hyperlinks or simply links can users navigate a single web page from one section to another section. This type of link is called intradocument hyperlink or named document link.

**An intradocument hyperlink uses a URL like this:**

<a href=”#top”>Back to top</a>

Here, the pound sign(#)indicates that you are pointing to a spot on the same page, not on another page.

<html>

<head>

<title>Linking within the same webpage</title>

</head>

<body>

<a href="http://www.google.com/" target="#bottom">E-Commerce</a>

<br><br><br><br><br>

<br><br><br><br><br>

<br><br><br><br><br>

<br><br><br><br><br>

<h2><a name="bottom"></a>E-Commerce</h2>

<p>E-Commerce or Electronics Commerce is a methodology of modern business which addresses the need of business   
organizations, vendors and customers to reduce cost and improve the quality of goods and services while increasing the   
speed of delievery.

</p>

</body>

</html>

**Link to a Spot on a different web page within the same website**:

You can combine intradocument and interdocument links to send visitors to a spot on a different web page on your site. Thus, to link to a spot named contacts on a page named index.html on your site, use this markup:

<a href=”index.html#contacts”>Contact Details</a>

**Link to a spot on a different web page on other websites**:

If you know that a page on another website has spots marked, you can use an absolute URL to point to a particular spot on that page, like this:

<a href=<http://www.rimtpd.org/index.html#contacts>>Contact Details</a>

**Link to Non-HTML resources**:

You can also link to such resource that are not even HTML pages at all, such as e-mail addresses, pictures, text files, Portable Document Format (PDF) files, compressed files, word processing documents etc. ,like this:

<a href=”abc.txt”>Text File</a>

<a href=”pc.doc”>Document File</a>

<a href=”pk.zip”>Compressed File</a>

<a href=”ecom.pdf”>PDF File</a>

<a href=”sunrise.jpg”>Image File</a>

<a href=<mailto:way2punit@gmail.com>>E-mail Address</a>

**ADVANCED E-MAIL LINKS**:

To create a link to an email address, you need to use the following syntax with the <a> element:

<a href=<mailto:way2punit@gmail.com>>E-Mail Address</a>

Here, the value of the href attribute starts with the keyword mailto, followed by a colon(:),and then the e-mail address you want the mail sent to .As with any other link, the content of the <a>element is the visible part of the link shown in the browser window.

You can also specify some other parts of the message, such as the subject, body, and e-mail address that should be CC(Carbon Copy) or BCC(Blind Carbon Copy)on the message.

To control other properties of the e-mail, you place a question mark(?) after the address and then use name/value pairs to specify the additional properties. The name and the value are separated by an equal sign(=).

For example; To make the subject line of the e-mail hello, you would add the subject property name followed by an equals sign, and then the term Hello, like so:

<A href=<mailto:way2punit@gmail.com?subject=>hello”>

You can specify more than one property by separating the name/value pairs with an ampersand (& ) symbol. Here, you can see that the subject and a CC address have been added in:

<a href=<mailto:way2punit@gmail.com?subject=Hello&cc=way2pardip@gmail.com>>

**The following table describes a list of properties you can add with your e-mail**:

|  |  |
| --- | --- |
| **Property** | **Description** |
| subject | Adds a subject line to the e-mail; you can add this to encourage the user to use a subject line that makes it easier to recognize where the mail has come from |
| body | Adds a message into the body of the e-mail, although you should be aware that users would be able to alter this message. |
| cc | Sends a carbon copy of the mail to the cc address; the value must be a valid e-mail address. |
| bcc | Secretly sends a carbon copy of the mail to the BCC address without any recipient seeing any other recipients ;the value must be a valid e-mail address. |

**UNIT-I**

**Chapter-3**

**Images, Audio and Video**

**Role of Images on the Web:**

Images are very important to beautify as well as to depict many concepts on your web page. Images can improve the design and the appearance of a web page. Web images can be used to illustrate an idea, show strictly visual information, provide navigation, and serve as decoration. The saying “a picture is worth a thousand words” is very true when illustrating an idea. Images can be used to show procedures, product applications, design styles, and a variety of other concepts. An image that provides more information than would be conveyed with plain text is truly valuable.

For Example: A website that teaches people to dance. While it might be possible to explain the various dance steps in writing, a diagram is far easier to understand.

So, as a web designer you should have clear understanding on how to use images in your web pages. Effectively used, images can help drive more visitors to your website, promote social sharing and ultimately help drive business goals such as sales:

Images are more important than the text placed on the website for few reasons:

* People prefer images over text. Websites with more images get more views.
* Images are the first thing a user sees on the website. Images enhance the content of the website.
* Images provide easier navigation.
* Images are worth a thousand words. One image can define the mood, content, audience, and tone of the website.
* Images are an important part of digital marketing. Images sell better than text.

**Adding Images Using the <IMG> Element**:

When an image is ready for the web, you need to use the correct markup to add it to your web page. You can add or insert an image in your web page by using HTML **<IMG>** element.

The **<IMG>** tag is an empty tag, which means that it contains attributes only, and has no closing tag. To display an image on a web page, you need to use the **SRC** attribute of **<IMG>** tag. **SRC** stands for **“source**”. The value of the **SRC** attribute is the **URL** of the image you want to display.

**Syntax of using <IMG> tag is:**

<IMG SRC=”URL”>

The **URL** points to the location where the image is stored.

The browser displays the image where the <IMG> tag occurs in the document. If you put an image tag between two paragraphs, the browser shows the first paragraph, then the image, and then the second paragraph.

**<IMG> Attributes:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| src | URL | Specifies the URL or location of an image, where the image is stored. |
| align | Top  Bottom  Middle  Left  Right | Specifies the alignment of an image according to surrounding elements. |
| alt | Text | Specifies an alternate text for an image,if the image cannot be displayed. |
| title | Text | Specifies title text as pop-up tips. |
| border | Pixels | Specifies the width of the border around an image. |
| height | Pixels | Specifies the height of an image. |
| width | Pixels | Specifies the width of an image. |
| hspace | Pixels | Specifies the white space on left and right side of an image. |
| vspace | Pixels | Specifies the white space on top and bottom of an image. |
| usemap | #mapname | Specifies an image as a client-side image-map. |

**Example:**

<html>

<head>

<title>Images In HTML</title>

</head>

<body>

<img src="abc.jpg" border="5" height="250" width="300" alt="Good Evening"

title="abcpic" align="right" vspace="20" hspace="30">

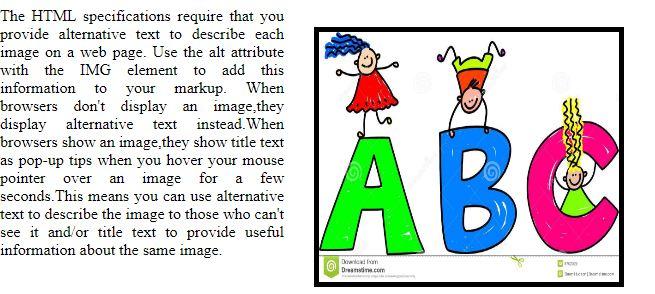
<p align="justify">

The HTML specifications require that you provide alternative text to describe each image on a web page. Use the alt attribute with the IMG element to add this information to your markup. When browsers don't display an image, they display alternative text instead. When browsers show an image, they show title text as pop-up tips when you hover your mouse pointer over an image for a few seconds. This means you can use alternative text to describe the image to those who can't see it and/or title text to provide useful information about the same image.

</p>

</body>

</html>

**Output:** ****

**USING IMAGES AS LINKS:**

Web pages often use images for navigation. They are more appealing than plain-text links. To create images that generate a link, you substitute an <IMG> element in place of text to which you would anchor your link.

**The following example demonstrates how to use an image as a link.**

**Example:**

<html>

<head>

<title>Image as a Link</title>

</head>

<body>

<p><h3>Image Link</h3>

<a href="http://www.music.pz10.com">

<img src="E:\vineela\smile.png" alt="Music" title="Musical Notes" width="250" height="200">

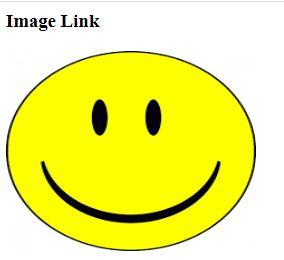
</a>

</p>

</body>

</html>

**Output:**

****

In the above output window, once you will click on the embedded image, link will be explored.

**IMAGE MAPPING:**

When you use an <IMG> element with an anchor element <A> to create a linking image. You can attach only one link to that image. To create a larger image that connects links to different regions or areas on the page, you need an image map. Image maps are used to define regions within a larger image as links.

So, the technique of making one image link to several pages, depending on where the image is clicked is called **image mapping**. You simply specify which areas of the image should link to where.

**The <MAP> tag is used to define an image-map**.

While creating an image map, following points should be considered:

1. First of all, the image that is going to form the map is inserted into web page using the <IMG> element as normal, except it carries an extra attribute called **usemap**.
2. The value of the **usemap** attribute is the value of the name attribute on the <MAP> element, preceded by a pound or hash sign.
3. The <MAP> element actually creates the map for the image and usually follows directly after the <IMG> element.
4. <MAP> element acts as a container for the <AREA> elements that actually define the clickable hotspots.
5. The <MAP> element carries only one attribute, the name attribute, which is the name that identifies the map.
6. The <AREA> element specifies the shape and the coordinates that define the boundaries of each clickable hotspot.

**Note: The <AREA> element is always nested inside a <MAP> tag.**

**<AREA> Attributes:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| shape | rect  circle  poly | Specifies the shape of the area. |
| cords | coordinates\_values | Specifies the coordinates of the area. |
| alt | text | Specifies an alternate text for the area.Required if the href attribute is present. |
| title | text | Specifies title text as pop-up-tips. |
| href | URL | Specifies the hyperlink target for the area. |
| target | blank  \_parent  \_self  \_top  framename | Specifies where to open the target URL. |

**Example**:

<html>

<body>

<h2>Image Maps</h2>

<p>Click on the computer, the phone, or the cup of coffee to go to a new page and read more about the topic:</p>

<img src="workplace.jpg" alt="Workplace" usemap="#workmap" width="400" height="379">

<map name="workmap">

<area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm">

<area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">

<area shape="circle" coords="337,300,44" alt="Cup of coffee" href="coffee.htm">

</map>

</body>

</html>

**Choosing The Right Image Format:**

Images and graphics can really bring your website to life, but it is important to learn how to prepare images for the Web. Otherwise, they can significantly increase the time it takes for a page to load.

When writing sites on your desktop or laptop computer, you may not realize how long a page will take to load; files that are sitting on your computer will load a lot faster than they would if they were on the internet. Therefore, choosing the right image format and saving your images correctly will ensure that when you put your site on the web for people to see, it will not be unnecessarily slow.

There are a variety of file formats used to encode digital images. The most popular Web based images are: **GIF(Graphics Interchange Format),** as designated by the .gif extension, **JPEG(Joint Photographic Experts Group),** as indicated by the .jpg or .jpeg file extension and **PNG(Portable Network Graphics),**as indicated by the .png file extension.

Each format has unique characteristics that make it fitting for particular uses. Typically it is the way the file format handles compression and the number of colors in the image. The more compressed the image, the smaller the file size. The downside is that image quality will suffer. There are two types of compression, lossy and lossless. Lossy compression reduces the size of the image by discarding image detail. Lossless compresses the image without discarding any detail, but the file sizes are larger. The number of different colors that make up an image can vary greatly. The simplest may only have one of two colors (a good example of these would be logos) while the more complex (like digital photos) contain millions. Certain file types, like. GIF, are better suited for images with less color while others, like .JPEG, are well suited for images with millions of colors.

The basic features of the main image formats are explained below in slightly more detail:

1. **.GIF**:

* GIF stands for “**Graphics Interchange Format**”, a bitmap image format introduced in 1987.
* GIF files are a format commonly used for graphics/pictures presented on websites.
* GIF supports up to 256 colors only, so if you try to save an image created with millions of colors as a GIF, you lose image quality.
* GIF is the best format for less-complex, non-photographic images, such as **line** **art, clip art, or icons**.
* GIF files are compressed using “lossless” compression technique, meaning image quality is not sacrificed as file size reduces.
* GIF files can also contain animations.
* GIF files are great for small icons and animated images, but they lack the color range to be used for high-quality photos.
* On the down side, GIFs are usually larger files, not as compressed as a JPEG, which calls for slow load times and large transfer rates.

1. **.JPG/JPEG:**

* JPG files, also known as JPEG files, stands for “Joint Photographic Experts Group”, a common file format for digital photos and other digital graphics.
* JPEG files are compressed using “lossy” compression technique, meaning image quality is lost as file size decreases.
* The JPEG file format supports 34-bit color (millions of colors).
* They are the most common file type for images taken with digital cameras, and widely used for photos and other graphics used on websites.
* JPEG images are not limited to a certain amount of color, like GIF images are. Therefore, the JPEG format is best for Compressing photographic images. So if you see a large, colorful image on the Web, it is most likely a JPEG file.

1. **.PNG:**

* PNG stands for “Portable Network Graphics”, a graphics file format that supports lossless data compression.
* PNG was created to combine the best aspects of GIF and JPEG. PNG has the same compression as GIF but supports 24-bit color (and even 32-bit color) like JPEG does.
* PNG allows the image data to be filtered before it is compressed. The purpose of filtering is to improve the compressibility of the data.
* PNG file format allows storage of images with greater color depth and other important information.
* Like a GIF, a PNG file is compressed in lossless fashion (meaning all image information is restored when the file is decompressed during viewing).
* An image in a PNG file can be 10 to 30% more compressed than in a GIF format.

**ADDING FLASH, VIDEO, AND AUDIO TO YOUR WEB PAGES:**

Flash is an authoring tool that designers and developers use to create presentations, applications, and other content that enables user interaction. You can make media-rich flash applications by including pictures, sound, video, and special effects. Flash is extremely well suited to creating content for delivery over the internet because its files are very small.

Flash can generate animations. These include banner ads, online greeting cards, cartoons and so on. Many website designers use Flash to design user interfaces. The interfaces include simple navigation bars as well as much more complex interfaces.

**ADDING FLASH TO WEB PAGES:**

To embed the flash movie into an HTML page, you should go back to your Flash program and do the following steps:

1. Choose File 🡪Open. Open a Flash movie you have created.
2. Choose File 🡪Export Movie.
3. Name the file “sample.swf”.(.SWF is the format that you place in a web page). Choose the location where the file is to be stored and click ok.
4. Open the HTML page where you want to insert your Flash movie and insert the following code:

<object width=”550” height=”400”>

<param name=”movie” value=”sample.swf”>

<embed src=”sample.swf” width=”550” height=”400”>

</embed>

</object>

This is the minimum code you need to embed a Flash movie in a web browser. In the above code, there is both an <embed> tag and an <object> tag. This is because the <object> tag is recognized by internet Explorer, and Netscape Navigator recognizes the <embed> tag and ignores the <object> tag.

Note: A broken icon will appear on the Web page if the user does not have the Flash plug-in-installed.

1. Finally, run the HTML file in your web browser to check the embedded flash movie.

**ADDING VIDEO TO WEB PAGES:**

One of the easiest ways to feature video on your website is to upload your video to a site such as YouTube, and then embed the YouTube player in your page using the code You Tube provides. If you don’t want to have your video on You tube , then look at some alternatives.

When you want to play a video on the web, there are two things you need to consider:

1. **File Format**: You need to choose the format of the video, and there are lots of different formats to choose from, including AVI, Flash Video, MPEG, QuickTime, and Windows Media. Which format you choose can influence the size of the file that users have to download and the quality of the picture they get to watch.

**Here is a short description of various video file formats:**

* **FLV:** FLV or Flash video is by far the most popular format for showing videos on the Web, largely because the Flash Player plug-in is installed on more computers than any of the other plugins needed to show video.
* **MPEG:** MPEG is a format that takes its name from the group who developed it, the **Motion Picture Experts Group**. Most computers will have a player capable of playing MPEG movies because they can be played in windows Media Player or QuickTime Player.
* **WMV:** WAV or Windows Media Video is part of the Microsoft Windows Media suite, and requires Windows Media Player to play it-Which comes with new PCs but requires users on other platforms to download a plug-in.
* **MOV:** MOV files are based on QuickTime which was initially developed for Mac. It can be played on most other operating systems but again requires users to download a plug-in.
* **MP4:** MP4 makes very small file sizes, but users need to install QuickTime to view it.
* **AVI**: AVI (Audio/Video Interleaved) was originally developed as a rival to QuickTime for windows. It is known as a container format because it contains a video compressed with another codec. This means that your computer may play some AVI files, and not others.
* **Real Video**: Real Video was one of the first formats used for showing video online. It requires a plug-in called RealPlayer to be installed on the computer. It is free but can be tricky to install. It is rapidly falling in popularity and is generally being replaced by FLV.

1. **Plug-in needed to play that type of file:**

Different formats require different plugins to play the movie. Some plugins are more popular than others and are pre-installed on maximum of computers.

**Example:** include: Flash player, QuickTime Player, RealPlayer, and Windows Media Player.

To tell a web browser what kind of video you are going to play, you can use the **type** attribute of the **<object>**element. Its **value** is something known as a MIME type(for example, the MIME type of a QuickTime movie is video/x-mov ).

**The following table provides a detail of the most common video formats on the web, the file extensions they use, the MIME type, and default players on PC:**

|  |  |  |  |
| --- | --- | --- | --- |
| **File format** | **File Extension** | **MIME Type** | **Default Player PC** |
| Flash Video player | .swf | Application/x-shockwave-flash | Flash Player(often pre installed) |
| MPEG | .mpg or .mpeg | Video/mpeg | (Usually)  Windows  Media Player |
| Windows Media Video | .wmv | Video/x-ms-wmv | Windows  Media Player  Pre-installed |
| Quick Time Movie | .mov | Video/quicktime | User must download plug-in |
| AVI | .avi | Video/x-msvideo | Windows  Media Player  Pre-installed |
| Real Media | .rm | Application/vnd.rm-realmedia | Real Player(often needs to be downloaded) |

**Adding Audio to Web Pages:**

As with videos, when you want to add audio to your website you need to consider:

1. **File Format:** Different types of audio files are saved using different compression techniques, which can affect the quality of the recording and file size.
2. **Plug-in needed to play that type of file**: As with video, visitors to your site need the right kind of player installed to play your chosen format. Luckily, there is a selection of file formats that you can play on most computers, so this is less of an issue than it is with video.

**The most popular audio file types are:**

* **MP3**: This is like the sound part of the MPEG video format. MP3 is one of the most popular formats for music recordings, and combines good compression with fairly good quality. Files are stored with the .mp3 file extension and will play in most browsers. The MIME type for MP3s is audio/mpeg.
* **WAVE**: This format was developed by IBM and Microsoft, and again plays in most web browsers. WAVE files are stored with the .wav file extension. Typically, WAVE files are not compressed, so their quality is much better than MP3(but users need good-quality speakers in order to hear the difference). The disadvantages with the uncompressed nature of the WAVE files is that the file sizes tend to be far larger than MP3s. The MIME type for WAVE files is audio/x-wav.

**Three other formats that are used on the web, all of which requires their own plug-ins, are:**

* **Window Media Audio**: Window Media Audio requires the Windows Media plug-in.
* **QuickTime:** QuickTime requires the QuickTime plug-in.
* **Real Audio**: Real Audio requires the Real Media plug-in.

**Adding Audio Using Default Media Players**:

In some web browsers, you do not need to download any extra plugins to play an MP3 or WAVE file.

**For Example**, you can just use the <object> element to include the audio file in the page, and the browser will automatically determine the type of plug-in needed to play your file by checking the value you have given for the type attribute(which is used to specify the MIME type of the file that you want to play).

**Here is an example of adding an MP3 file to a web page:**

<object width="300" height="42" type="audio/mpeg" data="my\_music.mp3">

<param name="src" value="my\_music.mp3"/>

<param name="autoplay" value="true"/>

<param name="autoplay" value="1"/>

<embed src="my\_music.mp3" width="300" height="42"></embed>

</object>

In the above example, the <param> elements tell the player to automatically start playing when the audio file loads. The first, autoplay, is used by QuickTime Player, while auto start is understood by Windows Media and Real Player.

**UNIT-I**

**Chapter-4**

**Tables:**

In HTML, tables make it easy to lay out data, text, and even images in a grid. Tables are just like spreadsheets and they are made up or rows and columns. Table model allows authors to arrange data text, preformatted text, images, links, forms, form fields, other table etc into rows and columns of cells. Using tables to divide the page into different sections is an extremely powerful tool. Table encloses elements like <CAPTION>,<TR>,<TH>,<TD>,<COLSPAN>,<COL><THEAD>,<TBODY> and <TFOOT>.

Tables are defined with the <table> tag. A table is divided into rows(with the< tr > tag),and each row is divided into data cells(with the <td> tag).The letters ‘td’ stands for “table data”, which is the content of a data cell. A data cell can contain text, images, list, paragraphs, forms, horizontal rules, tables, etc. If you do not specify a border attributes the table will be displayed without any border.

**Purpose**: Creates a table in a web page.

**Attributes**:

* **ALIGN**: Specifies the horizontal alignment of the table in the browser window. Set to **LEFT, CENTER, or RIGHT.**
* **BACKGROUND:** Specifies the url of a background image to be used as a background for the table. All cell contents are displayed over this image. Note that if the image is smaller than the table, it is tiled to fill the table. Set to a URL>
* **BGCOLOR**: Sets the back ground color of the table cells,
* **BORDER**: Sets the border width; set to a pixel width .If you set this attribute to 0,no border appears.
* **BORDERCOLOR**: Sets the external border color for the entire table. Set to an RGB triplet color value or a predefined color name.
* **CELLPADDING**: Sets the spacing between cell walls and cell contents. Set to a pixel size.
* **CELLSPACING**: Gives the distance between cells .Sets to a pixel size.
* **HEIGHT**: Gives the height of the whole table, in pixels.
* **WIDTH**: Sets the width of the table ; set to a pixel value or a percentage of the display are(add a percent sign[%] to such values.
* **HSPACE**: Sets the horizontal padding for the whole table, in pixels.
* **VSPACE**: Sets the vertical padding for the whole table in pixels
* **TITLE**: Holds additional information.

**Table Code**:

<TABLE>

………..

…………

</TABLE>

The following table list tags available in HTML

|  |  |
| --- | --- |
| **Tag** | **Description** |
| Table | Defines a table |
| Th | Defines a table header |
| Tr | Defines a table row |
| Td | Defines a table data |
| Caption | Defines a table caption |
| Cellpadding | This controls the distance between the data in a cell and the boundaries of the cell |
| Cellspacing | This controls the distance between cells |
| Colspan | This is useful when one row of the table needs to be certain number of columns wide |
| Rowspan | This allows a cell a take up more than row. It can be set by giving a numeric value. |
| Thead | Defines a table head. |
| Tbody | Defines a table body. |
| Tfoot | Defines table footer. |

**Example:**

<html>

<head>

<title>Table With Borders</title>

</head>

<body>

<h4>TABLE WITH A NORMAL BORDER</H4>

<table border="1">

<tr>

<td>First</td>

<td>Row</td>

</tr>

<tr>

<td>Second</td>

<td>Row</td>

</tr>

</table>

<h4>TABLE WITH A THICK BORDER:</H4>

<table border="8">

<tr>

<td>First</td>

<td>Row</td>

</tr>

<tr>

<td>Second</td>

<td>Row</td>

</tr>

</table>

<h4>TABLE WITH A VERY THICK BORDER:</h4>

<table border="15">

<tr>

<td>First</td>

<td>Row</td>

</tr>

<tr>

<td>Second</td>

<td>Row</td>

</tr>

</table>

</body>

</html>

**<TR> Creating A Table Row**:

**Purpose**: Creates a row in a table. Encloses **<TH>** and **<TD>** elements

**Attributes**:

* **ALIGN**: Specifies the horizontal alignment of the text in this table row. Set to **LEFT, CENTER, or RIGHT.**
* **BGCOLOR:** Sets the back ground color of the table cells.
* **BORDERCOLOR**: Sets the external border color for the rows.
* **VALIGN:** Sets the vertical alignment of the data in this row. Set to **TOP,MIDDLE,BOTTOM or BASELINE**.
* **TITLE**: Holds additional information

**Table Code**:

<TABLE>

<TR>

………..

…………

</TR>

………

………

</TABLE>

For each row in your table, there is a **<TR>** element; but how does the browser know how many columns you want to use? In the simple Table Model, it simply checks how many **<TH>** or**<TD>**elements you put into a row**.<TH>** elements creates table headings and **<TD>** elements create the table data that appears in the cell of a table.

**<TH>:Creating Table Headings**: Heading information in a table is defined with the <TH> tag. All major web browsers display the text in the <TH> element as bold and centered.

**Purpose:** Creates a table heading; just like table data, but usually bold and centered vertically and horizontally.

**Attributes**:

* **ALIGN**: Specifies the horizontal alignment of the text in this table row. Set to **LEFT, CENTER, or RIGHT.**
* **BACKGROUND:** Specifies the background image for the table cell. All cell contents are displayed over this image.
* **BGCOLOR**: Sets the background color of the table cells.
* **BORDERCOLOR**: Sets the external border color for the cell.
* **COLSPAN**: Indicates how many cell columns of the table this cell should span. Set to a positive integer.
* **ROWSPAN**: Indicates how many rows of the table this cell should span. Set to a positive integer.
* **HEIGHT**: Sets the height of the cell. Set to pixel value.
* **WIDTH:** Sets the width of the cell. Set to pixel value or percentage of the display area.
* **VALIGN:** Sets the vertical alignment of the data in this cell. Set to **TOP,MIDDLE,BOTTOM or BASELINE**

**Table Code**:

<TABLE>

<TR>

<TH>TIC</TH>

<TH>TAC</TH>

<TH>TOE</TH>

</TR>

………

………

</TABLE

**Example:**

<html>

<head>

<title>Headings in a Table</title>

</head>

<body>

<table border="3">

<tr>

<th> NAME</th>

<th>JOB</th>

<th>SALARY</th>

</tr>

<tr>

<td>Amit Kumar</td>

<td>President</td>

<td>100000</td>

</tr>

<tr>

<td>Raman Gupta</td>

<td>General Manager</td>

<td>75000</td>

</tr>

<tr>

<td>Rahul Sharma</td>

<td>Salesman</td>

<td>25000</td>

</tr>

</table>

</body>

</html>

**Table With a Caption**:

It is a good idea to add a title or caption to a table that describes the table data. To add a caption to the table,<CAPTION> tag is used. There could be only one caption per table. The <CAPTION>tag must be inserted immediately after the <TABLE> tag. The <CAPTION> tag is supported in all major browsers.

The following example demonstrates how to create a table with a caption.

**Example:**

<html>

<head>

<title>Table With A Caption Table</title>

</head>

<body>

<table border="3">

<caption><b>My Caption</b></caption>

<tr>

<td>100</td>

<td>200</td>

<td>300</td>

</tr>

<tr>

<td>400</td>

<td>500</td>

<td>600</td>

</tr>

</table>

</body>

</html**>**

**Sample Program on Table With Attributes:**

<html>

<head>

<title>Headings in a Table</title>

</head>

<body>

<table border="10" align="center" height="60%" width="65%" bordercolor="green" bgcolor="yellow" cellspacing="10" cellpadding="30">

<caption><B>TABLES IN HTML</B></html>

<tr>

<th> NAME</th>

<th>JOB</th>

<th>SALARY</th>

</tr>

<tr>

<td>Amit Kumar</td>

<td>President</td>

<td>100000</td>

</tr>

<tr>

<td>Raman Gupta</td>

<td>General Manager</td>

<td>75000</td>

</tr>

<tr>

<td>Rahul Sharma</td>

<td>Salesman</td>

<td>25000</td>

</tr>

</table>

</body>

</html>

**<TD>:Creating Table Data**:

**Purpose**: Specifies the data for a table cell. Used inside the **<TR>** element.

**Attributes**:

* **ALIGN**: Specifies the horizontal alignment of the text in this table row. Set to **LEFT, CENTER, or RIGHT.**
* **BACKGROUND:** Specifies the background image for the table cell. All cell contents are displayed over this image.
* **BGCOLOR**: Sets the background color of the table cells.
* **BORDERCOLOR**: Sets the external border color for the cell.
* **COLSPAN**: Indicates how many cell columns of the table this cell should span. Set to a positive integer.
* **ROWSPAN**: Indicates how many rows of the table this cell should span. Set to a positive integer.
* **HEIGHT**: Sets the height of the cell. Set to pixel value.
* **WIDTH:** Sets the width of the cell. Set to pixel value or percentage of the display area.
* **VALIGN:** Sets the vertical alignment of the data in this cell. Set to **TOP,MIDDLE,BOTTOM or BASELINE**

**Table Code**:

<HTML>

<HEAD><TITLE>SAMPLE PROGRAM ON TABLE</TITLE>

</HEAD>

<BODY>

<TABLE BORDER=”2” CELLPADDING=”2” CELLSPACING=”2”>

<TR>

<TH>TIC</TH>

<TH>TAC</TH>

<TH>TOE</TH>

</TR>

<TR>

<TD>X</TD>

<TD>O</TD>

<TD>X</TD>

</TR>

<TR>

<TD>O</TD>

<TD>X</TD>

<TD>O</TD>

</TR>

<TR>

<TD>X</TD>

<TD>O</TD>

<TD>X</TD>

</TR>

</BODY>

</TABLE>

**Grouping Sections of a Table**:

HTML offers some techniques that allows you to group together cells, rows, and columns of a table.

* Use of rowspan and colspan attributes to make cells stretch over more than one row or column

The rowspan attribute of <TH> and <TD> elements specify the number of rows a cell should span.

The colspan attribute of <TH>and <TD> elements specify the number of columns a cell should span.

**<!---Sample Program on using rowspan and colspan attributes--->**

<html>

<head>

<title>Rowspan and Colspan Attributes</title>

</head>

<body>

<table border="5" align="center" cellspacing="5" cellpadding="5">

<caption>

<b><u>Use of Rowspan and colspan attributes</u></b><br><br>

</caption>

<tr>

<th colspan="2">Name</th>

<th rowspan="2">Class</th>

<th rowspan="2">RollNumber</th>

</tr>

<tr>

<th>First Name</th>

<th>Last Name</th>

</tr>

<tr>

<td>Ajay</td>

<td>Kumar</td>

<td>BCS</td>

<td>101</td>

</tr>

<tr>

<td>Rethu</td>

<td>Kumari</td>

<td>BSC</td>

<td>102</td>

</tr>

<tr>

<td>Nikhil</td>

<td>Raj</td>

<td>BCOM</td>

<td>103</td>

</tr>

</table>

</body>

</html>

* Split a table into three sections:a head,body.and foot

There are occasions when you may wish to distinguish between the body of a table (where most of the data is held) and the headings or may be even the footers.

**For example**. Think of a bank statement: you may have a table where the header contains column headings, the body contains a list of transactions, and the footer contains the balance in the account.

If the table is too long to show on a screen, then the header and footer might remains in view all the time, while the body of the table gains a scrollbar. Similarly, when printing a long table that spreads over more than one page, you might want the browser to print the head and foot of a table on each page.

The three elements for separating the head, body, and foot of a table are:

**<THEAD>:** To create separate table header.

**<TBODY>:** To indicate the main body of the table.

**<TFOOT>:** To create separate table footer.

**<!---Sample program on Thead, Tbody,Thead Tags--->**

<html>

<head>

<title>Splitiing a Table-Head,Body,Footer</title>

</head>

<body>

<table border="5">

<thead>

<tr>

<th>Transaction Date</th>

<th>Payment Type</th>

<th>Amount Paid</th>

<th>Balance</th>

</tr>

</thead>

<tfoot>

<tr>

<th colspan="2">Total</th>

<td>55,000</td>

<td>45,000</td>

</tr>

</tfoot>

<tbody>

<tr>

<td>12 Jun 19</td>

<td>Amazon.com</td>

<td>15,000</td>

<td>10,000</td>

</tr>

<tr>

<td>15 july 19</td>

<td>Visa Payment</td>

<td>30,000</td>

<td>20,000</td>

</tr>

<tr>

<td>19 july 19</td>

<td>Whole Foods</td>

<td>10,000</td>

<td>15,000</td>

</tr>

</tbody>

</table>

</body>

</htmL>

* **Group Columns Using the <COLGROUP> element**

If two or more columns are related, you can use the <COLGROUP>element to explain that those columns are grouped together.

For example, in the following table, there would be six columns. The first four columns are in the first column group, and the next two columns are in the second column group.

**Sample Program**

<html>

<head>

<title>Grouping Columns</title>

</head>

<body>

<table border="2" height="15%" width="40%">

<colgroup span="4">

<colgroup span="2">

<tr>

<td>1</td>

<td>2</td>

<td>3</td>

<td>4</td>

<td>5</td>

<td>6</td>

</tr>

</table>

</body>

</html>

When the <COLGROUP> element is used, it comes directly after the opening <TABLE> tag and carries a span attribute, which is used to indicate how many columns the group contains.

**NESTED TABLES**: Nesting of tables refers to Table within another Table.

In HTML, you can use one table inside another table. Not only tables, you can use almost all tags inside table data tag<TD> or table header tag<TH>.

Nesting tables can lead to complex tables layouts, inner table should begin and end in the same cell of the outer container table table. You can do the nesting of tables up to any number of levels.

<html>

<head>

<title>Nesting of Table</title>

</head>

<body>

<table border="10" align="center" height="60%" width="65%" bordercolor="green" bgcolor="yellow" cellspacing="5" cellpadding="5">

<caption><B>Nested Tables</b></caption>

<tr>

<th>Name</th>

<th>Job</th>

<th>Salary</th>

</tr>

<tr>

<td>Raju</td>

<td>

<table border="5" align="center">

<tr>

<td>President</td>

<td>Vice President</td>

</tr>

<tr>

<td>Manager</td>

<td>General Manager</td>

</tr>

</table>

</td>

<td>100000</td>

</tr>

<tr>

<td>Rethu</td>

<td>Executive</td>

<td>75000</td>

</tr>

<tr>

<td>Nikhil</td>

<td>Salesman</td>

<td>25000</td>

</tr>

</table>

</body>

</html>

**Accessing Tables**:

Tables can contain a huge volume of data and they provide a very helpful visual representation of data. With a little efforts or planning, you can make tables a lot easier for all to understand.

Here are some things you can do to ensure your tables are easy to understand:

* Add captions to your tables. The <CAPTION> element clearly associates a caption with a table, and the screen reader will read the caption to the user before they see the table so that they know what to expect. If the listener knows what to expect, it will be easier to understand the information.
* Always try to use the <TH>element to indicate a table heading.
* Always put headings in the first row and the first column.
* Avoid using nested tables, as this will make it harder for the user of a screen reader to follow.
* Avoid using rowspan and colspan attributes, which again make it harder for the user with a screen reader to follow.

**UNIT-I**

**Chapter-5**

**FORMS**

**Forms:** Forms are a mechanism that allows the user to type information into fields on a browser screen and submit to a web server, forms also allow to create interactive web pages.

Forms are used to collect different kinds of user input.

**For Example:** Forms are required when you want to collect some data (like name, email, address, phone number etc.) from the site visitors who visit your site.

**Form Controls**:

A form basically an area that can contain several elements, called form controls, which are used to collect user input. User interacts with the form through such named controls.

There are different types of form controls that you can use to collect data from the user.

Some of the commonly used form controls are discussed below:

1. **Text field/Textbox:** This control is used to get single-line text input. It is used for items that require only one line of user input, such as search boxes or e-mail addresses.

1. **Textarea**: This control is used to get multi-line text input. It is used when the user is required to give details that may be longer than a single sentence.

1. **Drop-down menus**: This control allows users to select one item from a drop-down list.

1. **Checkbox:** This control allows users to select several items from a list of possible options such as when you want to select all the skills you have from a given list.

1. **Radio Button:** Radio buttons are similar to checkboxes in that they can be either on or off, but the key difference is that when you have a group of radio buttons that share the same name, only one of them can be selected. Once one radio button has been selected, if the user clicks another option, the new option is selected and the old one deselected.

1. **Button**: Buttons are most commonly used to submit a form, although they are sometimes used to clear or reset a form.

1. **File Upload**: This control allows user to browse the system ans attach a file for uploading on the website.

1. **Label:** This control is used to associate a text label with the input text field.

**<FORM> Element:**

In HTML a form is defined with the <FORM> element. The <FORM> element can contain the sub tags implemented between the starting an closing of the root element.HTML <FORM> support with many elements which play vital role in creating and executing user defined structure for processing the information.

The following are the sub element of form tag which supports with different controls to accept and submit messages from HTML forms. All are case insensitive.

* **<INPUT>**
* **<TEXTAREA>**
* **<BUTTON>**
* **<SELECT>**
* **<OPTION>**
* **<FIELDSET>**
* **<LABEL>**

* **<FORM>........</FORM> Element:**

**Purpose**: Forms are used to collect different kinds of user input.

**Attributes:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| name | form\_name | Specifies the name of a form |
| method | Get  Post | Specifies the HTTP method to use when the form is submitted. Possible values:   * get(the form data is appended to the URL when submitted). * post(the form data is not appended to the URL). |
| action | URL | Specifies URL of the script file, where to send the form-data when a form is submitted. |
| target | \_blank  \_self  \_parent  \_top | Specifies where to display the response that is received after submitting the form. |

**<INPUT> tag:**

The primary element of **HTML FORMS** which helps the user to create various controls using its type attribute. It is an independent tag(not a paired element) which don’t have a correspondent closing tag. It supports with 13 different form controls with different structure and presentation which helps to accept the data in various form form the user.

Each control has its own structure and presentation format some are the types under input element and some are separate elements. These controls are used to create interactive web pages.

When a form is submitted, the current value of each **INPUT** element within the **FORM** is sent to the server as name/value pairs.

**<input type=   1.text**

**2. password**

**3. checkbox**

**4. radio**

**5. reset**

**6. submit**

**7. button**

**8. file**

**9. image**

**Attributes:**

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Value** | **Description** |
| type | Button  checkbox  file  hidden  image  password  radio  reset  submit  text | Specifies the type of control. |
| name | field name | Assigns a name to the input control. |
| align | Left  right  top  middle  bottom | Specifies the alignment of an image input (only for type=”image”). |
| alt | Text | Specifies  an alternate text for images(only for type=”image”) |
| size | Number | Specifies the width,in characters,of an <INPUT> element. |
| src | URL | Specifies the URL of the image to use as a submit button(only for type=”image”). |
| value | Text | Specifies the value of an <INPUT> element. |
| maxlength | Number | Specifies the maximum  number of characters allowes in an <INPUT> element. |
| height | Pixels | Specifies the height of an <INPUT> element(only for type=”image”). |
| width | Pixels | Specifies the width of an <INPUT> element s(only for type=”image”). |

**NOTE:** The **<INPUT>** tag is an empty tag, it contains attributes only.

**<!--- Sample program on Form tags-🡪**

<html>

<head>

<title>Form Example</title>

</head>

<body>

<form name="form1" action="http://www.server.com//cgibin/program"    method="post">

Name:<input type="text" size="10" maxlength="40"  name="name"><br>

PassWord:<input type="password" size="10" maxlength="10" name="password"></br>

Hobbies:<input type="checkbox">Reading

<input type="checkbox">Music

<input type="Checkbox">Dancing<br>

Gender:<input type="radio" name="gender" value="male">Male

<input type="radio" name="gender" value="Femaile">Female<br>

<input type="image" src="siri.jpg" alt="submit" width="48" height="48">

<input type="button" value="send">

</form>

</body>

</html>

**Drop-Down List:**

A drop-down list is a selectable list. It displays a list of selectable options in a small input field that can be dropped down to reveal multiple options.The dropdown format is handy for saving space in  a form as the input field is only one line in height and it can be used to “hide” long lists of menu options.

**<SELECT>...........</SELECT> AND <OPTION>..........</OPTION> TAGS**:

 The **<SELECT>** tag is used to create a drop-down list. The **<OPTION>** tags inside the **<SELECT>** element defines the available options in the list.

**<SELECT> Attributes:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| Name | Name | Defines a name for the drop-down list. |
| Size | number | Defines the number of visible options in a drop-down list. |
| Multiple | multiple | Specifies that multiple options can be selected at once. |
| Disabled | disabled | Specifies that a drop-down list should be disabled. |

**<OPTION> Attributes:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| Label | Text | Specifies a shorter label for an option. |
| Value | Text | Specifies the value to be sent to a server. |
| Selected | selected | Specifies that an option should be pre-selected when the page loads. |
| Disabled | disabled | Specifies that an option should be disabled. |

**<OPTGROUP> .............</OPTGROUP>Element**:

 In order to create a group element so that the multiple blocks for multiple sub items can be created. Which is implemented with <optgroup></optgroup> tag a dependent element.

**<!---Sample on using optgroup tag--->**

<html>

<head>

<title>Example on optgroup tag</title>

</head>

<body>

<form>

Course:<select>

<optgroup label="BSC">

<option>MSCs</option>

<option>MPCs</option>

<option>MECs</option>

<option>BZc</option>

<option>MPC</option>

</optgroup>

<optgroup label="BCOM">

<option>Bcom General</option>

<option>Bcom Computers</option>

<option>Bcom Honors</option>

</optgroup>

<optgroup label="BA">

<option>PPP</option>

<option>EPP</option>

</optgroup>

</select>

</form>

</body>

</html>

**<TEXTAREA>...........</TEXTAREA> Tag:**

Textarea is a multiline text input control. A text area can hold an unlimited number of characters.

The <TEXTAREA> tag is used within a form to define a textarea. The size of a textarea can be specified by cols and rows attributes of <TEXTAREA>tag.

**<TEXTAREA> Attributes:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| name | text | Specifies a name for the text area |
| cols | number | Specifies the visible width of atext area |
| rows | number | Specifies the visible number of lines in  a text area. |
| disabled | disabled | Specifies that a text area should be disabled. |
| maxlength | number | Specifies the maximum number of characters allowed in the text area. |

**Example:** <textarea name="message"

  cols="80"

  rows="20"

  maxlength="2000"

  wrap>

Enter your text here:

</textarea>

**<BUTTON>........</BUTTON> tag:**

There are several ways to create a button in HTML, and they all map to the “push button” role. Buttons can be created using **<INPUT>** tag as well .Inside a **<BUTTON>** element you can put content, like text or images. This is the difference between **<BUTTON>** element and buttons created with the **<INPUT>** element.

There are 2 major category of Pushbutton Predefined and user defined where predefined is considered as Action buttons.

**User defined Button**: This helps the user to create his own control button and fix the action using script code. This can be implemented in 2 ways.

1. As the Type of input tag as <input type=”button” value=”login”>
2. As the element of form tag <button value=”login”>

**ACTION BUTTONS:** There are two types’ action buttons. They are submit and reset. When the user clicks on the submit button, the values that have been entered into the form are sent the program that process the form.

**SUBMIT**:

<input type=”submit” value=”submit”>

The arrangement to value is displayed inside the graphic corresponding to the submit button.

**RESET:** The reset button is to allow the user to clear all of the input that have thus far entered into the form. This is a quick for users to start over from the original default values.

<input type=”reset” value=”reset”>

**<BUTTON> Attributes:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| name | name | Specifies a name for the button |
| type | button  reset  submit | Specifies the type of button |
| value | text | Specifies an initial value for the button |
| disabled | disabled | Specifies that a button should be disabled. |

**For Example**:

<button name=”button1” type=”button” >ClickMe</button>

<button name=”button2” type=”button”>Clear</button>

<button name=”button3” type=”button”>Submit</button>

**CREATING BUTTONS USING IMAGES**:

In HTML, you can create submit buttons using images. Such buttons are called image buttons. Image buttons have the same effect as submit buttons, allowing a user to click an image to submit a form.

**Example:** <input type=”image” name=”img1” src=”Sunset.jpg” height=”50” width=”50”>

**FILE UPLOAD**:

A special type of form control supported in HTML is the file upload control. This control can be used to browse the user’s local system and attach a file for uploading on the website. The syntax of the file upload field is relatively simple as shown:

**Example:** <input type=”file” name=”upfile” id=”upfile”>

This may allow the users to include one or more files from their system in the form submission. The files could be text files, images files, or other data.

**<LABEL>..........</LABEL> Tag:**

The label is used to tell users the value that should be entered in the associated input filed.

The <LABEL> tag defines a label for an <INPUT> element. It is used to add a label to a form.

The FOR attributes of the <LABEL> tag should be equal to the ID attribute of the related element to bind them together.

**<!--- Sample Program on Label- ->**

<html>

<head>

<title>Labels in HTML forma</title>

</head>

<body>

<p>

<center>

<h2><u>Labels in Form</u></h2>

<p>Which soft drink would you like?</p>

<form>

<input type="radio" name="drink" id="pepsi">

<label for="pepsi">PEPSI</label>

<br>

<input type="radio" name="drink" id="COLA">

<label for="cola">COLA</label>

</form>

</center>

</p>

</body>

</html>

**<FIELDSET>&<LEGEND>:** These are used to draw the border for the selected form with a title in order to identify each from separately in a single web page.

**<FIELDSET>**draw the lined box. <**LEGEND>** set title.

You can pretty much take it from here, but let’s take a look at the code that created the first little section that surrounds the two text boxes:

**Example:**

<fieldset>

<legend><b>Step One:Personal Information</b></legend>

Name:<input type=”text” size=”20”><br>

Email:<input type=”text” size=”20”>

</fieldset>

**Example: Programme to illustrating form controls and many input types**.

<html>

<head>

<title>Using Form Controls</title>

</head>

<body bgcolor="coral" text="white">

<center>

<font size="5" face="matura mt script capitals" color="blue">

ONLINE STUDENT DETAILS

</font><br>

</center>

<form>

<fieldset>

<legend>Student Information</legend>

<h2>Student Name:<input type="text"><br><br>

Father's Name:    <input type="text"><br><br>

Address:<textarea rows=3 cols=20></textarea><br><br>

Date of Birth:DD-MM-YYYY

<input type="text" size=4>-

<input type="text" size=4>-

<input type="text" size=4><br><br>

Course Details

<select>

<optgroup label="B.COM">

<optionvalue="Comp">Computers</option>

<option value="Reg">Regulars</option>

<option value="Hon">Honors</option>

</optgroup>

<optgroup label="B.SC">

<option value="MPCS">MPCS</option>

<option value="MSCS">MSCS</option>

<option value="MECS">MECS</option>

</optgroup>

</select>

<br><br>

<font size=6 color="blue" family="playbill">

Registration Details

</font><br>

<hr size=3 color="red">

Login ID:<input type="text"><br>

Password:<input type="password"><br>

Conformation Password:<input type="password"><br>

<hr size=3 color="red">

Gender:

<input type="radio" name="r1">Male

<input type="radio" name="r2">Female<br><br>

Select your Favourite<br>

<input type="checkbox">Sports

<input type="checkbox">Dancing

<input type="checkbox">Singing

<input type="checkbox">painting

<input type="checkbox">Gamming

Any other??<input type="text"><br><br>

Profile Picture:

<input type="image" src="22.jpg" height="100" width="100"><br><br>

UR Resume<input type="file"><br><br>

Want to check Again<input type="button" value="Yes">

<input type="button" value="NO"><br><br>

<center>

<input type="SUBMIT"><input type="RESET">

</center>

</form>

</body>

</html>

**SENDING FORM DATA TO THE SERVER**:

Web is just a form of client/server interaction. Web Server and Web Browser (also known as Web Client) are the two main elements in all applications. The Web Browser sends requests and the Web Server listens to those requests and responds/accomplishes the required task. The Web Server and Web Browser communicate with each other through a common protocol (HTTP HyperTextTransferProtocol).

There are two methods that a web browser can use to send form data to the web server:

1. **GET Method**
2. **POST Method**

You have to specify which method should be used by adding the method attribute of the <FORM> element. If the <FORM> element does not carry a method attribute, then by default the GET method will be used.

1. **GET Method**: When you send form data to the server using the HTTP GET method, the form data is appended to the URL that is specified in the action attribute of the <FORM> element.

The form data is separated from the URL using a question mark (?).Following the question mark, you get the name/value pairs for each form control. Each name/value pair is separated by an ampersand (&).

One of the advantages of passing form data in a URL is that it an be bookmarked. If you look at searches performed on major search engines such as Google, they tend to use the get method so that the page can be bookmarked.

The GET method, however, has some disadvantages. Indeed, when sending sensitive data such as password shown here, or debit/credit card details, you should not use the GET method because the sensitive data becomes part of the URL and is in full view to everyone.

1. **POST Method**: When you send data from a form to the server using the HTTP POST method, the form data is sent transparently in what is known as the HTTP headers. This means POST method appends form-data inside the body of the HTTP request and data is not shown in the URL.

It you are sending sensitive information such as debit/credit card details, the data should be sent under a Secure Socket Layer (SSL), and they should be in encrypted form.

**GET METHOD VS POST METHOD**

|  |  |
| --- | --- |
| **GET Method** | **POST Method** |
| Information sent from a form with the GET method is **visible to everyone**. All variable names and values are displayed in the URL. | Information sent from a form with the POST method is **invisible to others.** All variable names/values are embedded within the body of the HTTP request. |
| GET method has limits on the amount of information to send. Upto 3000 characters can be send with the GET method. | POST method has no limits on the amount of information to send. |
| However, because the variables are displayed in the URL, it is possible to bookmark the page. | As the variables are not displayed in the URL, so it is not possible to bookmark the page. |
| GET method may be used for sending non-sensitive data. It should not be used for sending passwords or other sensitive information. | More preferable method for sending form data. |

**UNIT-I**

**Chapter-6**

**FRAMES**

**Frames:** HTML allows us to divide a web page or a single browser window into several pieces or panes, called frames.

Each frame can contain a separate HTML document. So, frames allow us to display more than one HTML document in the same browser window simultaneously. Each frame is independent of the others. A collection of frames in the browser window is known as a **FRAMESET.**

**There are few drawbacks also you should be aware of with frames.**

**They are:**

* The web developer must keep track of more HTML documents.
* It is difficult to print the entire page.
* Users often dislike them.
* It prevents linking challenges.
* All web browsers do not support frames.

**<FRAMESET> ......</FRAMESET>and <FRAME> Elements:**

To create a frameset document, first you need the <FRAMESET> element, which is used instead of the <BODY> element. The <FRA MESET> element defines how to divide the window into frames. Each frameset defines a set of rows or columns. The values of the rows/columns indicate the amount of screen area each row/column will occupy. Each frame is then represented by a <FRAME> tag. The <FRAME> tag defines which HTML document will initially open in each frame.

**CREATING VERTICAL FRAMESET**:

The following example demonstrates how to make a vertical frameset with three different frames.

<html>

<head>

<title>Vertical FrameSet Example</title>

</head>

<frameset cols="30%,40%,30%">

<frame name="f1">

<frame name="f2">

<frame name="f3">

</frameset>

</html>

**CREATING HORIZONTAL FRAMESET**:

The following example demonstrates how to make a horizontal frameset with three different frames.

<html>

<head>

<title>Horizontal Frameset Example</title>

</head>

<frameset rows="30%,40%,30%">

<frame name="f1">

<frame name="f2">

<frame name="f3">

</frameset>

</html>

**<NOFRAMES>.......</NOFRAMES>:**

Every web browser does not support frames. The <NOFRAMES> element is a special element for web browsers that do not support frames. However, if you add a <NOFRAMES> element containing some text for browsers that do not support frames, you will have to enclose the text in the <BODY> tag. The <NOFRA MES> element goes inside the <FRAMESET> element. It can contain all the HTML elements that you can find inside the <BODY> element of a normal HTML page.

**Example**:

<html>

<head>

<title>NoFrames Element</title>

</head>

<frameset rows="30%,40%,30%">

<frame name="f1">

<frame name="f2">

<frame name="f3">

<noframes>

<body>

Your Browser doesnot support frames.

</body>

</noframes>

</frameset>

</html>

**Note:** If your web browser supports frames, then the <NOFRAMES>> text remains invisible.

**<FRAMESET> .........</FRAMESET>**

**Attributes**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| cols | Columns size(in pixels or %) | Specifies the number of columns and their width (sizes) in either pixels or percentages. Default is 100%. |
| rows | Row size(in pixels or %) | Specifies the number of rows and their height (sizes) in either pixels or percentages. Default is 100%. |

**<FRAME>**

**Attributes:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| frameborder | 0 or 1 | Specifies whether or not to display border around the frame. |
| src | URL | Specifies the URL or location of the document to show in a frame. |
| name | Frame name | Specifies the name of the frame. |
| noresize | Noresize | Specifies that a frame is not resizable. |
| scrolling | Yes  No  Auto | Specifies whether or not to display scrollbars in a frame.   * Yes-specifies that scrollbars are to be added always. * No-specifies that no scrollbars will be provided. * Auto-specifies that the scrollbar will be added as and when required. |
| marginheight | Pixels | Specifies the top and bottom margins of a frame. |
| marginwidth | Pixels | Specifies the left and right margins of a frame. |

**Example using all attributes**

<html>

<head>

<title>Frames Attributes</title>

</head>

<frameset cols="40%,\*">

<frame name="f1" noresize="noresize" src="siri.jpg" frameborder="0"

marginheight="30" marginwidth="30"scrolling="auto">

<frame name="f2" noresize="noresize" scrolling="yes" frameborder="1">

</frameset>

<HTML>

**NAVIGATION FRAME/CREATING LINKS BETWEEN FRAMES**:

A navigation frame contains a list of links with the second frame as the target. The second frame will show the linked document. This is one of the most popular uses of frames. As each <FRAME> element can carry the name attribute to give each frame a name. This name is used in the links to indicate which frame the new page should load into.

**Example:**

**Step 1: Create a file “NF.html” with the following code**

<html>

<head>

<title>Navigation Frame Linking of frames</title>

</head>

<frameset cols="30%,\*">

<frame name="f1" src="Flink.html">

<frame name="f2">

</frameset>

</html>

**Step 2: Create a file “Flink.html” with the following code:**

<html>

<head>

<title>Linked File</title>

</head>

<body>

<a href="C:\vineela\HTML\_PROGRAMS\siri.jpg" target="f2">CLICK HERE</a>

</body>

</html>

Now, once you will execute the file “**NF.html**” ,and the see the output.

**SETTING A DEFAULT TARGET FRAME USING <BASE> ELEMENT**:

You can set a default target frame using the <BASE> element in any page that contains links that should open in another frame. The <BASE> element resides inside the <HEAD> element. The <BASE> element should carry an attribute called target, whose value is the name for the frame you want the content to be loaded into.

**Example:**

**Step1: creating a file “NFRM.html” with the following code:**

<html>

<head>

<title>Navigation Frame-Linking of Frames</title>

</head>

<frameset cols="30%,\*">

<frame name="f1" src="Flink.html">

<frame name="f2">

</frameset>

</html>

**Step 2:Create a file :Flink.html” with the following code:**

<html>

<head>

<title>Linked File</title>

<base target="f2">

<head>

<body>

<a href="C:\vineela\HTML\_PROGRAMS\siri.jpg">Click Here</a>

</body>

</html>

Now, once you will execute the file “NFRM.html”, and see the output.

When you will click on the link CLICK HERE placed in the first/left frame, linked document will be explored in the default target frame set by using the <BASE> element in the HTML page that contains links(Flink.html).

**NESTING OF FRAMES/NESTED FRAMESETS:**

Nesting of frames refers to a frame within another frame. A frame which contains other frames is called a container or outer frame. On the other hand, contained frames are called inner frame(s).Framesets may be nested to any level.

**Example:**

<html>

<head>

<title>Nesting of Frames in HTML</title>

</head>

<frameset rows="40%,\*">

<frameset cols="50%,50%">

<frame name="f11">

<frame name="f12">

</frameset>

<frameset cols="33%,33%,\*">

<frame name="f21">

<frame name="f22">

<frame name="f23">

</frameset>

</frameset>

</html>

**INLINE FRAMES:**

Frames can also be used within a single HTML page. These are known as inline frames.

You can define an inline frame with the <IFRAME> tag. The <IFRAME> tag is not used within a <FRAMESET> element. Instead, it appears anywhere in your document. The <IFRAME>tag defines a rectangular region within the document in which the browser displays a separate document, including scrollbars and borders.

Use the src attribute with <IFRAME> to specify the URL of the document that occupies the inline frame. All of the other optional attributes for the <IFRAME>tag, including name, frameborder, marginheight, marginwidth and scrolling behave exactly like the corresponding attributes for the <FRAME> tag.

**Example:**

<html>

<head>

<title>Inline Frame</title>

</head>

<body>

<iframe src="C:\vineela\HTML\_PROGRAMS\siri.jpg"></iframe>

</body>

</html>