

# **WWW & HTML**

## **BE VII – Computer & Electronics**

## Before you begin...

A **web browser** or **Internet browser** is a software application for retrieving, presenting, and traversing information resources on the World Wide Web. An *information resource* is identified by a Uniform Resource Identifier (URI) and may be a web page, image, video, or other piece of content. Hyperlinks present in resources enable users to easily navigate their browsers to related resources.

*In order of release:*

---

- [Netscape Navigator](#) and [Netscape Communicator](#), October 13, 1994
  - [Internet Explorer 1](#), August 16, 1995
  - [Opera](#), 1996
  - [Mozilla Navigator](#), June 5, 2002
  - [Safari](#), January 7, 2003
  - [Mozilla Firefox](#), November 9, 2004
  - [Google Chrome](#), September 2, 2008
- 

### Web Servers

Web servers are computers that deliver (*serves up*) Web pages. Every Web server has an IP address and possibly a domain name. Web servers are computers on the Internet that host websites, serving pages to viewers upon request. This service is referred to as web hosting. Every web server has a unique address so that other computers connected to the Internet know where to find it. The Internet Protocol (IP) address looks something like this: 69.93.141.146. This address maps to a more human friendly address, such as <http://www.sarojpandey.com.np>. Web hosts rent out space on their web servers to people or businesses to set up their own websites. The web server allocates a unique website address to each website it hosts.

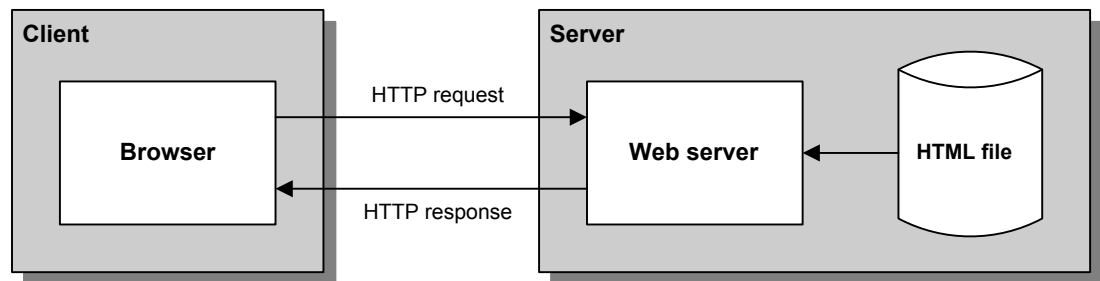
---

**For example**, if you enter the URL <http://www.sarojpandey.com.np/index.html> in your browser, this sends a request to the Web server whose domain name is *sarojpandey.com.np*. The server then fetches the page named *index.html* and sends it to your browser.

Any computer can be turned into a Web server by installing server software and connecting the machine to the Internet. Two leading Web servers are Apache, the most widely-installed Web server, and Microsoft's Internet Information Server (IIS).

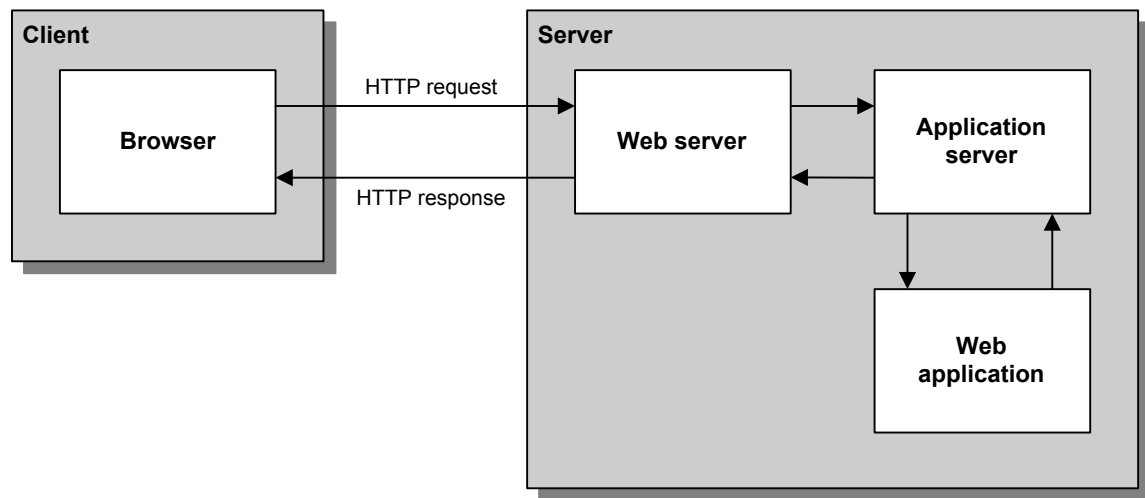
---

## Static web page



- ◆ A *static web page* is an HTML document that is the same each time it's viewed. It doesn't change in response to user input.
- ◆ Static web pages are usually simple HTML files that are stored on the web server with a file extension of .htm or .html. When a browser requests a static web page, the web server retrieves the file from disk and sends it back to the browser.
- ◆ A web browser requests a page from a web server by sending the server an HTTP message known as an *HTTP request*.
- ◆ The HTTP request includes, among other things, the name of the HTML file being requested and the Internet address of both the browser and the web server.
- ◆ A user working with a browser can initiate an HTTP request in several ways. One way is to type the address of a web page, called a *URL*, or *Uniform Resource Locator*, into the browser's address area and press Enter. Another way is to click a link that refers to a web page.
- ◆ A web server replies to an HTTP request by sending a message known as an *HTTP response* back to the browser.
- ◆ The HTTP response contains the addresses of the browser and the server as well as the HTML document that's being returned.

## Dynamic web page



- ◆ A *dynamic web page* is an HTML document that's generated by a web application. Often, the web page changes according to information that's sent to the web application by the browser.
- ◆ When a web server receives a request for a dynamic web page, the server passes the request to an *application server*.
- ◆ The application server executes the web application, which generates an HTML document. This document is returned to the application server, which passes it back to the web server. The web server, in turn, sends the document back to the browser.
- ◆ After the page is displayed, the user can interact with it using its controls. Some of those controls let the user *post* the page back to the server, so it's processed again using the data the user entered.
- ◆ Each application mapping specifies which application should be run to process files with that extension.
- ◆ If the file extension isn't in the list of application mappings, the requested file is returned to the browser without any processing.
- ◆ The process that begins with the user requesting a web page and ends with the server sending a response back to the client is called a *round trip*.
- ◆ After a web application generates an HTML document, it ends. Then, unless the data the application contains is specifically saved, that data is lost.

## HTTP Overview

- HTTP, Hyper Text transfer Protocol is the standard Web transfer protocol.
- The HTTP is the language that Web clients and Web servers use to communicate with each other.
- It is essentially the backbone of the web.
- It is constantly evolving protocol with several versions in use and others are still under development.
- This protocol has two items: the set of requests from browsers to servers and the set of responses going back the other way.
- It is a stateless protocol and does not maintain any information from one transaction to the next, so the next transaction needs to start all over again
- The advantage is that an HTTP server can serve a lot more clients in a given period of time, since there's no additional overhead for tracking sessions from one connection to the next.
- Default Port used by HTTP is 80.

**Client-side scripting** is the class of computer programs on the web that are executed in *client-side*, by the user's web browser, instead of *server-side* (on the web server). This type of computer programming is an important part of the Dynamic HTML (DHTML) concept, enabling web pages to be scripted; that is, to have different and changing content depending on user input, environmental conditions (such as the time of day), or other variables. Web authors write client-side scripts in languages such as JavaScript (Client-side JavaScript) and VBScript.

Client-side scripts are embedded within an HTML document (hence known as an "embedded script"), but they may also be contained in a separate file, which is referenced by the document (or documents) that uses it (known as an "external script"). Upon request, the necessary files are sent to the user's computer by the web server (or servers) on whom they reside. The user's web browser executes the script, and then displays the document, including any visible output from the script. By viewing the file that contains the script, users may be able to see its source code.

**Server-side scripting** is a web server technology in which a user's request is fulfilled by running a script directly on the web server to generate dynamic web pages. It is usually used to provide interactive web sites that interface to databases or other data stores. This is different from client-side scripting where the viewing web browser, usually in JavaScript, runs scripts. The primary advantage to server-side scripting is the ability to highly customize the response based on the user's requirements or queries into data stores. Server side scripts are written in languages such as [Perl](#), [PHP](#), and [ASP.NET](#) etc.

SSS produce output in a format understandable by web browsers (usually HTML), which is then sent to the user's computer. The user cannot see the script's source code (unless the author publishes the code separately), and may not even be aware that a script was executed.

From security point of view, server-side scripts are never visible to the browser as these scripts are executed on the server and emit HTML corresponding to user's input to the page.

***Programs that run on a user's local computer without ever sending or receiving data over a network are not considered clients, and so the operations of such programs would not be considered client-side operations.***

**Comparison:** Client-side scripts have greater access to the information and functions available on the user's browser, whereas server-side scripts have greater access to the information and functions available on the server. Server-side scripts require that their languages interpreter be installed on the server, and produce the same output regardless of the client's browser, operating system, or other system details. Client-side scripts do not require additional software on the server; however, they do require that the user's web browser understands the scripting language in which they are written.

---

## FTP

**File Transfer Protocol**, the protocol for exchanging files over the Internet. FTP works in the same way as HTTP. FTP uses the Internet's TCP/IP protocols to enable data transfer. FTP is most commonly used to download a file from a server using the Internet or to upload a file to a server (e.g., uploading a Web page file to a server).

- File Transfer Protocol (FTP) is a method of transferring files from a client to a server or vice versa.
- Files are transferred over the Internet using TCP/IP protocol.
- FTP is old-time protocol that maintains two simultaneous connections.
- The first connection uses the telnet remote login protocol to log the client into an account and process commands via the *protocol interpreter*.
- The second connection is used for the *data transfer process*.
- Whereas the first connection is maintained throughout the FTP session, the second connection is opened and closed for each file transfer.
- The FTP protocol also enables an FTP client to establish connections with two servers and to act as the third-party agent in transferring files between the two servers.
- FTP servers rarely change, but new FTP clients appear on a regular basis.

- Although FTP is a command-line oriented protocol, the new generation of FTP clients hides this orientation under a GUI environment.
- A client makes a TCP connection to the server's port 21. This connection, called the *control connection*, remains open for the duration of the session, with a second connection, called the *data connection*, either opened by the server from its port 20 to a negotiated client port or opened by the client from an arbitrary port to a negotiated server port as required to transfer file data.

## What is HTML???

- HTML or **HyperText Markup Language** is designed to specify the logical organization of a document, with important hypertext extensions.
- HTML instructions divide the text of a document into blocks called *elements*.
- These can be divided into two broad categories:
  - Those that define how the **BODY** of the document is to be displayed by the browser, and
  - Those that define information about the document, such as the **title** or relationships to other documents.
- The detailed rules for HTML (the names of the tags/elements, how they can be used) are defined using another language known as the SGML (**Standard Generalized Markup Language**).
- HTML is a set of special codes that can be embedded in text to add formatting and linking information.
- HTML is the language interpreted by a Browser.
- The HTML file must have an extension “.htm” or “.html”.
- Any text editor can be used to create HTML file.

HTML stands for **Hyper Text Markup Language**. HTML is a presentation language. We use HTML language to display information according to our need. An HTML file is a text file containing small **markup tags**. The markup tags tell the Web browser **how to display** the page. An HTML file must have an **.htm** or **.html** file extension. An HTML file can be created using a **simple text editor**.

HTML is very popular language used on web because of its interoperability. HTML language is platform independent i.e. HTML files can be opened on any platform. HTML files can be written using a simple text editor like notepad which is present in all the operating system.

HTML language is used to create web pages. Web pages can be viewed using application software called a **Web browser**. Popular browsers are **Internet Explorer, Mozilla Firefox and Netscape Navigator**. A web browser parses the HTML file containing markups (html tags) and displays the information with the proper format as specified in the HTML document. HTML tags are also called mark-up. HTML tags are surrounded by the two characters < and >. The surrounding characters are called angle brackets. HTML tags normally come in pairs like <b> and </b>. The first tag in a pair is the start tag, the second tag is the end tag. The text between the start and end tags is the element content. HTML tags are not case sensitive; <b> means the same as <B>.



**Structure of an HTML document is shown below:**

**<html>**

**<head>**

**<title>** Title of page **</title>**

**</head>**

**<body>**

This is the place where the information to be displayed in a web page is written.

**</body>**

**</html>**

Every html document must start with <html> tag. It shows the starting of HTML document. <head> tag contains information like the title of the document and other information which describes about the content of the document.

BODY part of a HTML document contains the information and its format to be displayed by the browser. HEAD part of a HTML document contains the information that is not displayed on the browser window. It defines information `about' the document, such as the title or relationships to other documents.

<body> tag is the place where we write all the information that is to be displayed in the web browser. It also contains other tags which defines how information are to be displayed in the web browser. <body> tag shows the starting of the body tag and </body> tag shows the ending of the body tag. </html> shows the ending of the HTML document. Every ending tag must have a forward slash as shown in </html> tag.

HTML discards whitespaces. HTML only considers a single space as a space. The browser automatically discards rest of the whitespace. Hence, we can use as much whitespace as we want while creating our HTML document. This makes html document easy to read or edit.

## **Versions of HTML**

### **HTML 2.0**

- It set the standard for core HTML features based upon current practice in 1994.

### **HTML 3.2**

- W3C's first Recommendation for HTML which represented the consensus on HTML features for 1996.
- **HTML 3.2** added widely-deployed features such as tables, applets, text-flow around images, superscripts and subscripts, while providing backwards compatibility with the existing HTML

2.0 standard.

### **HTML 4.0**

- First released as a W3C Recommendation on 18 December 1997.
- A second release was issued on 24 April 1998 with changes limited to editorial corrections.
- This specification has now been superseded by HTML 4.01.

### **HTML 4.01**

- HTML 4.01 is the current official standard.
- It includes support for most of the proprietary extensions, plus support for extra features (Internationalized documents, support for Cascading Style Sheets, extra TABLE, FORM, and JavaScript enhancements), that are not universally supported.
- This is the last version of HTML.
- After this XHTML was released which stands for eXtensible HyperText Markup Language.

### **HTML 5.0**

- This is the new version of HTML with many exciting new features. This version is still under development.

### **HTML Elements/Tags**

- The HTML instructions, along with the text to which the instructions apply, are called HTML *elements*.
- The HTML instructions are themselves called *tags*, and look like <element\_name> -- that is, they are simply the element name surrounded by left and right angle brackets.
- The content in the web-page is written after the starting tag, and closed with the end tag.
- E.g: <element\_name> text to be written HERE </element\_name>
- The end tag has slash character in front of it.
- HTML tags are not case sensitive; <b> means same as <B>.

### **Empty Elements**

- Some elements are *empty* -- that is, they do not affect a block of the document in some way.
- These elements do not require an ending *tag*.

- An example is the <HR> element, which draws a horizontal line across the page.

## HTML Tag Attributes

- Many elements can have arguments that pass parameters to the interpreter handling the element.
- These arguments are called *attributes* of the element.
- An attribute is a customizable option for a tag.
- In other words, attributes are used to define the properties of a tag.
  - Example: <p align = "left"> Trial Example </p>.
  - In the above example the align attribute allows you to specify how text in a paragraph is arranged on the page.
- Not all tags support attributes.
- Some tags support multiple attributes, and the attributes are listed one after another in the start tag, separated by space.
- Attributes are always set to the opening tag.

## HTML Tags Lists

TITLE	TAG	DESCRIPTION
<b>Basic Elements</b>		
Document Type	<HTML> </HTML>	document root element, beginning and end of file
Title	<TITLE> </TITLE>	document title, must be in header
Header	<HEAD> </HEAD>	descriptive info, such as title
Body	<BODY> </BODY>	bulk of the page, notes body of document
<b>Formatting</b>		
Bold	<B> </B> or <strong></strong>	bold text style
Italic	<I> </I>	italic text style
Underline	<U> </U>	underlined text (not widely implemented)
Strikeout	<STRIKE> </STRIKE>	strike-through text (not widely implemented)

Strikeout	<S> </S>	strike-through text (not widely implemented)
Subscript	<SUB> </SUB>	subscript numbers like footnotes
Superscript	<SUP> </SUP>	superscript numbers like cross - reference numbers
Pre formatted	<PRE> </PRE>	pre formatted text (display text spacing as-is)
Center	<CENTER> </CENTER>	centers text and images
Blinking	<BLINK> </BLINK>	blinking text, Netscape only
Font Size	<FONT SIZE=?> </FONT>	local font size(ranges from 1-7)
Change Font Size	<FONT SIZE="+ -?"> </FONT>	controls font size rendered
Font Color	<FONT COLOR="#\$\$\$\$\$"> </FONT>	controls font color rendered
Select Font	<FONT FACE="****"> </FONT>	the style of the text, such as Times New Roman
Marquee	<MARQUEE> </MARQUEE>	scrolling text (IE only)
<b>Links</b>		
Link Something	<A HREF="URL"> </A>	links text or graphic to another URL
Link to Location	<A HREF="URL#****"> </A>	links text or graphic an anchor in an other document
Link to Location in Current Page	<A HREF="#****"> </A>	links text or graphic an anchor in current document
Target Window	<A HREF="URL" TARGET="****"> </A>	links text or graphic to a URL in a new browser widow
Action on Click	<A HREF="URL" ONCLICK="****"> </A>	takes effect when user clicks on the item (Javascript)
Mouseover Action	<A HREF="URL" ONMOUSEOVER="****"> </A>	takes effect when user moves pointer over item
Link to Email	<A HREF="mailto:@"> </A>	creates blank e-mail to indicated address with visitor's default e-mail client
<b>Graphics and Sound</b>		
Display Image	<IMG SRC="URL">	displays image from the indicated URL

Alignment	<IMG SRC="URL" ALIGN=TOP BOTTOM MIDDLE LEFT RIGHT>	aligns the image
Dimensions	<IMG SRC="URL" WIDTH=? HEIGHT=?>	the dimensions, in pixels, of the image
Border	<IMG SRC="URL" BORDER=?>	border, in pixels, around the image
<b>Dividers</b>		
Paragraph	<P> </P>	paragraph (closing tag often unnecessary)
Align Text	<P ALIGN=LEFT CENTER RIGHT> </P>	aligns paragraph
Justify Text	<P ALIGN=JUSTIFY> </P>	justify's paragraph's text
Line Break	 	a single carriage return
Horizontal Rule	<HR>	horizontal line
Alignment	<HR ALIGN=LEFT RIGHT CENTER>	alignment of horizontal line
Thickness	<HR SIZE=?>	thickness, in pixels, of horizontal line
Width	<HR WIDTH=?>	width, in pixels, of horizontal line
Width Percent	<HR WIDTH="%">	width(as a percentage of page width), in pixels, of horizontal line
Solid Line	<HR NOSHADE>	horizontal line without the 3D cutout look
No Break	<NOBR> </NOBR>	prevents line breaks
<b>Structural Elements</b>		
Heading	<H?> </H?>	document header, the ? defines 6 levels (#'s 1-6)
Strong Emphasis	<STRONG> </STRONG>	strongly emphasized text, usually displayed as bold
Address	<ADDRESS> </ADDRESS>	author information
Large Font Size	<BIG> </BIG>	uses a large text size
Small Font Size	<SMALL> </SMALL>	use a small text size
<b>Backgrounds</b>		

Tiled Background	<BODY BACKGROUND= "URL">	causes the image to tile as the background of the page
Watermark	<BODY BGPROPERTIES= "FIXED">	Static image which remains in the same location as visitors scroll.
Background Color	<BODY BGCOLOR= "#\$\$\$\$\$\$">	solid background color of the page
Text Color	<BODY TEXT="#\$\$\$\$\$\$">	color of the text throughout the page
Link Color	<BODY LINK="#\$\$\$\$\$\$">	color of all links throughout the page
Visited Link	<BODY VLINK="#\$\$\$\$\$\$">	color of all links that have already been clicked on by visitor
Active Link	<BODY ALINK="#\$\$\$\$\$\$">	color of link while being selected
<b><u>Lists</u></b>		
Unordered List	<UL> </UL>	list with bulleted items
List Item	<LI> </LI>	indicates an item on the list
Bullet Type	<UL TYPE=DISC CIRCLE SQUARE>	shape of bullet for the whole list
Bullet Type	<LI TYPE=DISC CIRCLE SQUARE>	shape of bullet for specific list item
Ordered List	<OL> <LI> </OL>	numbered list
Numbering Type	<OL TYPE=A a I i 1>	type of numbering for the whole list
Numbering Type	<LI TYPE=A a I i 1>	type of numbering for specific list item
Starting Number	<OL START=?>	starting number for list
Starting Number	<LI VALUE=?>	starting number for this & subsequent items
Definition List	<DL> </DL>	a list of definitions
Definition Term	<DT> </DT>	definition term
Definition	<DD> </DD>	definition of a term
Menu List	<MENU> </MENU>	display menu type list
Directory List	<DIR> </DIR>	directory link

<b>Tables</b>		
Define Table	<TABLE> </TABLE>	signals the beginning of a table
Table Alignment	<TABLE ALIGN= LEFT RIGHT CENTER>	aligns the table within the browser window
Table Border	<TABLE BORDER=?> </TABLE>	border of table, you can set the value (aka width)
Cell Spacing	<TABLE CELSPACING=?>	places specific amount of space between the individual cells within a table
Cell Padding	<TABLE CELLPADDING=?>	places specific amount of space between the cells border and its contents
Desired Width	<TABLE WIDTH=?>	width of table in pixels
Width Percent	<TABLE WIDTH=%>	width of table in percentage of page
Table Color	<TABLE BGCOLOR="#\$\$\$\$\$\$"> </TABLE>	overall background color of table
Border Color	<TABLE BORDERCOLOR="#\$\$\$\$\$\$"> </TABLE>	the color of the table border
Table Row	<TR> </TR>	table row
Alignment	<TR ALIGN= LEFT  RIGHT  CENTER  MIDDLE  BOTTOM>	alignment of the table row
Table Cell	<TD> </TD>	specific table cell, must appear within table rows
Alignment	<TD ALIGN= LEFT RIGHT CENTER VALIGN= TOP MIDDLE BOTTOM>	alignment of the table cell
Columns to Span	<TD COLSPAN=?>	identifies the the number of columns the cell should span
Rows to Span	<TD ROWSPAN=?>	identifies the the number of rows the cell should span
Desired Width	<TD WIDTH=?>	width of cell in pixels
Width Percent	<TD WIDTH="%">	width of cell as percentage of table
Cell Color	<TD BGCOLOR="#\$\$\$\$\$\$">	background color of table cell

Header Cell	<TH> </TH>	table cell for header information (bold & centered)
Alignment	<TH ALIGN= LEFT  RIGHT  CENTER  MIDDLE BOTTOM>	alignment of the header cell
Table Body	<TBODY>	identifies the specific body section of the table
Table Footer	<TFOOT> </TFOOT>	separates group of cells to serve as footer material for the table (must come before <THEAD>)
Table Header	<THEAD> </THEAD>	separates group of cells to serve as header material for the table
Table Caption	<CAPTION> </CAPTION>	caption for a table
Alignment	<CAPTION ALIGN=TOP BOTTOM LEFT RIGHT>	alignment for the caption of a table
<b>Frames</b>		
Frame Document	<FRAMESET> </FRAMESET>	creates layouts of frames (instead of <BODY>)
Row Heights	<FRAMESET ROWS=,,,> </FRAMESET>	comma separated list of size of each row within the frameset (pixels or %)
Column Widths	<FRAMESET COLS=,,,> </FRAMESET>	comma separated list of size of each column within the frameset (pixels or %)
Borders	<FRAMESET FRAMEBORDER="yes no"> </FRAMESET>	identifies if a frame has a visible border or not
Border Width	<FRAMESET BORDER=?> </FRAMESET>	width of frame border if visible
Border Color	<FRAMESET BORDERCOLOR="#\$\$\$\$\$"> </FRAMESET>	color of frame border if visible
Frame Spacing	<FRAMESET FRAMESPACING=?> </FRAMESET>	number of pixels of reserved space between frames
Define Frame	<FRAME>	specific contents of an individual frame
Display Document	<FRAME SRC="URL">	identifies the initial contents of the frame



Frame Name	<FRAME NAME="*" _blank _self _parent _top>	assigns a name to the current frame
Margin Width	<FRAME MARGINWIDTH=?>	distance between content and frame's left and right margins
Margin Height	<FRAME MARGINHEIGHT=?>	distance between content and frame's top and bottom margins
Scroll bar	<FRAME SCROLLING="YES NO AUTO">	controls how the window is or isn't scrolled
Not Re-sizable	<FRAME NORESIZE>	prohibits the document viewer from changing dimensions of the frame
Borders	<FRAME FRAMEBORDER="yes no">	controls whether frame has a border
Border Color	<FRAME BORDERCOLOR="#\$\$\$\$\$">	color of border of frame

## HTML Lists

- HTML provides three type of lists.
- They are listed below:
  - 1. Ordered List:**
    - A list of multi-line paragraphs, listed separately and ordered numerically in some way.
    - The list items are marked with numbers.
    - <OL ...> creates an ordered list.
    - "Ordered" means that the order of the items in the list is important.
    - By default, the number starts with 1,2,3.....
    - An ordered list starts with the <ol> tag.
    - Each list item starts with the <li> tag.

- Example:

```
<ol>
<li>Coffee</li>
<li>Milk</li>
</ol>
```

Here is how it looks in a browser:

1. Coffee
2. Milk

## **2. Unordered List:**

- A list of multi-line paragraphs, listed separately and usually marked by a bullet or similar symbol (Unordered List)
- <UL ...> creates an unordered list.
- The *unordered* part means that the items in the list are not in any particular order.
- The list items are marked with bullets (typically small black circles).
- An unordered list starts with the <ul> tag.
- Each list item starts with the <li> tag.

- Example:

```
<ul>
<li>Coffee</li>
<li>Milk</li>
</ul>
```

Here is how it looks in a browser:

- Coffee
- Milk

## **3. 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.

- Example:

```
<dl>
<dt>Coffee</dt>
<dd>Black hot drink</dd>
```

```
<dt>Milk</dt>
<dd>White cold drink</dd>
</dl>
```

Here is how it looks in a browser:

Coffee

Black hot drink

Milk

White cold drink

## Frames

- Frames allow displaying more than one web-page in a single browser at a same instance of time.
- HTML tags `<frameset>.....</frameset>` is used to divide a browser screen into two or more HTML recognizable unique regions.
- Each unique region is called frame.
- Each frame can be loaded with a different document and hence, allow multiple HTML documents to be seen concurrently.

The disadvantages of using frames are:

- The web developer must keep track of more HTML documents
- It is difficult to print the entire page.

The major advantages of using frames are:

- It can be given an individual URL, so it can load information independent of the other frames on the page;
- It can be given a NAME, allowing it to be targeted by other URLs, and;
- It can resize dynamically if the user changes the window's size. (Resizing can also be disabled, ensuring a constant frame size.)

## The Frameset Tag

- The `<frameset>` tag defines how to divide the window into frames.
- Each `<frameset>` defines a set of rows **or** columns.
- The `<frameset>` tags require one of the following two attributes depending on whether the

screen has to be divided into rows or columns.

### **The two attributes are:**

#### 1. Rows:

- This attribute is used to divide the screen into multiple rows.
- The each row can be set with different values depending on the required size of the row.

#### 2. Cols:

- This attribute is used to divide the screen into multiple columns.
- The values for both Rows and Cols can be:
  - A number in pixels (***Not commonly used.***)
  - Expressed as a percentage of the screen resolution.
  - The symbol \*, which indicates the remaining space.

### Example of <frameset> tag:

```
<frameset rows="33%,33%,*">
```

divides the browser screen into 3 equal horizontal sections.

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

divides the browser screen into 2 different vertical sections.

### **The <frame> Tag**

- Once the screen is divided into rows and columns, each unique section can be loaded with different HTML documents.
- This is achieved by using the <frame> tag.
- The **<frame>** tag defines what HTML document to put into each frame.
- The attributes of the **<frame>** tag are:

Attributes	Description
SRC="url"	Indicates the url of the document to be loaded into the frame.
MARGINHEIGHT="n"	Specifies the amount of white space to be left at the top and bottom of the frame.

MARGINWIDTH="n"	Specifies the amount of white space to be left along the sides of the frame.
NAME="name"	Gives the same unique name so it can be targeted by other documents. The name given must begin with an alphanumeric character.
NORESIZE	Disables the frames resizing capability.
Scrolling	Controls the appearance of horizontal and vertical scrollbars in a frame. This takes the values YES/NO/AUTO.

### **Example of using <frame> tag:**

```
<frameset cols="25%,75%">
  <frame src="frame_a.htm">
  <frame src="frame_b.htm">
</frameset>
```

### **In the example above we have a frameset with two columns.**

- The first column is set to 25% of the width of the browser window.
- The second column is set to 75% of the width of the browser window.
- The HTML document "frame\_a.htm" is put into the first column, and the HTML document "frame\_b.htm" is put into the second column.

### **HTML Links**

- A link is a connection from one Web resource to another.
- A *link* has two ends -- called *anchors* -- and a direction.
- The link starts at the "source" anchor and points to the "destination" anchor, which may be any Web resource (e.g., an image, an HTML document, an element within an HTML document, etc.).
- The text or an image that provides such linkages is called hypertext, hyperlink, or hotspot.

### **What is Hyperlink?**

- A Hyperlink is a connection between an HTML element such as text, an image, or anything else on a page and other resource.

- That link might be to another web page, an external image, or an e-mail address.

### **Difference between Hyperlink and Normal HTML Text:**

- Appears in blue color.
  - The default color setting in a browser for hyperlink text or image.
  - The color can be set dynamically via HTML program if required.
- The Hyperlink text/image is underlined.
- When the mouse cursor is placed over it, the standard arrow shaped mouse cursor changes to the shape of a hand.

### **Changing the color of Links:**

- To change the link color there are three attributes that can be specified with the **<body>** tag.
- These are:
  - LINK (Normal)
  - ALINK (Active)
  - VLINK (Visited)

## **Types of Hyperlink**

### **There are three types of Hyperlinks:**

#### **1. Inter-page Hyperlink**

- In this type of link the control flows from one-page to another.

Example:

```
<a HREF="myExample.htm"> Click for Example </a>
```

*You can specify the relative as well as the absolute path of the file that you want to call.*

#### **2. Intra-page Hyperlink**

- Intra-page Hyperlink is a link within a same page.
- Sometimes, a jump is required to a different location in the same document.
- Since the jump has to be targeted to a specific location the two steps need to perform.
  - a) Identify the location with a name and
  - b) Jump to that location using the name.

Example:

`<a name = "top"> The HTML text is written here </a>`

`<a HREF="#top"> Goto Top </a>`

### 3. Email Hyperlink

- This type of Hyperlink is used especially to write e-mail.
- The link does not open any web-pages but opens the outlook express for writing mail.
- You can write the mail and send.

#### Steps:

a) First type any text like:

Email: info@kcc.edu.np

Surround the email address with the anchor tags i.e. <a>, but instead linking to the web page, use the 'mailto' command to link it to an e-mail program.

Email: `<a HREF="mailto: info@kcc.edu.np"> info@kcc.edu.np </a>`

b) Save the page and view it in browser.

### 4. External Links

You can also have external links like links, when clicking upon them you can jump to next web page.

In such scenario you have to give the path of web page like:

`<a HREF="http://www.google.com"> Goto Google </a>`

## HTML Forms

Forms are the most popular way to make web pages interactive. A form on a web page looks similar to a form on a sheet of paper which allows the user to enter requested information and submit it for further processing. A form can have different types of form elements for different purpose like textbox, list box, checkbox, radio buttons, dropdown menus, text area etc.

### 1. HTML Text Fields

**type** - Determines what kind of input field it will be. Possible choices are text, submit, and password.

**name** - Assigns a name to the given field so that you may reference it later.

**size** - Sets the horizontal width of the field. The unit of measurement is in blank spaces.

**maxlength** - Dictates the maximum number of characters that can be entered.

**HTML Code:**

```
<form method="post">  
  Name: <input type="text" size="10" maxlength="40" name="name"> <br />  
  Password: <input type="password" size="10" maxlength="10" name="password">  
</form>
```

*Do not use the password feature for security purposes. The data in the password field is not encrypted and is not secure in any way.*

## 2. Submit Buttons

**HTML Code:**

```
<form method="post">  
  Name: <input type="text" size="10" maxlength="40" name="name"> <br />  
  Password: <input type="password" size="10" maxlength="10" name="password"><br />  
  <input type="submit" value="Send">  
</form>
```

## 3. HTML Radio Buttons

Radio buttons are a popular form of interaction. You may have seen them on quizzes, questionnaires, and other web sites that give the user a multiple choice question. Below are a couple attributes you should know that relate to the radio button.

**value** - specifies what will be sent if the user chooses this radio button. Only one value will be sent for a given group of radio buttons.

**name** - defines which set of radio buttons that it is a part of.

**HTML Code:**

```
<form method="post">  
  What kind of shirt are you wearing? <br />  
  Shade:
```



```



```

#### 4. HTML Check Boxes

Check boxes allow for multiple items to be selected for a certain group of choices. The check box's name and value attributes behave the same as a radio button.

##### HTML Code:

```

<form method="post">
Select your favorite cartoon characters.


```

#### 5. HTML Drop down Lists (Known as Combo Box)

Drop down menu are created with the <select> and <option> tags. <select> is the list itself and each <option> is an available choice for the user.

##### HTML Code:

```

<form method="post">
College Degree?
  <select name="degree">
    <option>Choose One</option>
    <option>Some High School</option>
    <option>High School Degree</option>
    <option>Some College</option>
    <option>Bachelor's Degree</option>

```

```

    <option>Doctorate</option>
<input type="submit" value="Email Yourself">
</select>
</form>

```

## TRY IT YOURSELF

### HTML Code:

```

<form method="post" action="mailto:youremail@email.com">
Musical Taste
<select multiple name="music" size="4">
    <option value="emo" selected>Emo</option>
    <option value="metal/rock" >Metal/Rock</option>
    <option value="hiphop" >Hip Hop</option>
    <option value="ska" >Ska</option>
    <option value="jazz" >Jazz</option>
    <option value="country" >Country</option>
    <option value="classical" >Classical</option>
    <option value="alternative" >Alternative</option>
    <option value="oldies" >Oldies</option>
    <option value="techno" >Techno</option>
</select>
<input type="submit" value="Email Yourself">
</form>

```

## 6. HTML Text Areas

Text areas serve as an input field for viewers to place their own comments onto. Forums and the like use text areas to post what you type onto their site using scripts. For this form, the text area is used as a way to write comments to somebody.

Rows and columns need to be specified as attributes to the <textarea> tag. Rows are roughly 12pixels high, the same as in word programs and the value of the columns reflects how many characters wide the text area will be. i.e. The example below shows a text area 5 rows tall and 20 characters wide.

### HTML Code:

```

<form method="post">
<textarea rows="5" cols="20" wrap="physical" name="comments">
Enter Comments Here
</textarea>
<input type="submit" value="Email Yourself">

```

</form>

*Note that any text placed between the opening and closing textarea tags will show up inside the text area when the browser views it.*

### Entities References that can be used in HTML:

In HTML we cannot directly use the special symbols so we use a technique called Entities References. With this we can keep any symbols in a web page. It takes a form: **&Entity\_Name;**

### Some of the mostly used symbols and their corresponding entities are as follows:

Result	Description	Entity Name
	non-breaking space	&nbsp;
<	less than	&lt;
>	greater than	&gt;
&	ampersand	&amp;
"	quotation mark	&quot;
'	apostrophe	&apos; (does not work in IE)
¢	cent	&cent;
£	pound	&pound;
¥	yen	&yen;
€	euro	&euro;
§	section	&sect;
©	copyright	&copy;
®	registered trademark	&reg;
×	multiplication	&times;
÷	division	&divide;

### Points for review:

- HTML stands for **Hyper Text Markup Language**.
- HTML file consist of tags also called markups to display information in an arranged way.
- Tags in HTML are predefined i.e. we cannot create our own tags.
- Web browser is application software that is used to view web pages created using HTML.

- HTML document is mainly divided into two parts: head and body.
- The current version of HTML that we are using now is HTML4.
- W3C stands for World Wide Web consortium is a body which looks after the standardization of HTML language.
- HTML file ends with .htm or .html extension.
- Every starting tag has its corresponding ending tag in HTML for e.g. `<b> .... </b>`.
- HTML also has empty tags like `<br />` which is used to break row.
- Comments are ignored by the browser.
- Tag can also have attributes, which are used to define the properties of a tag.  
For e.g. `<p align="center"> ... </p>`
- Character entities are used to display some special characters which cannot be typed from the keyboard. It is also used to display some of the characters which are forbidden to be written as element content.

## HTML Editors

HTML editors are tools used to edit the HTML content. The tool can be different in different platform (Operating System). In windows environment Notepad is the mostly used text editor. Wordpad can also be used.

TextEdit is the tool used in Mac OSX for editing text. We write HTML code with HTML editor and save that as .html or .htm so that we can view the output with any browser.

## HTML Convertors

HTML convertors are also tools, where the developer use click, drag-and-drop and other graphical technique to create the elements like table, form etc. and corresponding HTML code will be automatically writer by the tool. With this tool the developer do not need to know the detailed code.

Macromedia Dreamweaver is mostly used HTML convertor. Microsoft FrontPage can also be used for the same purpose.

## DHTML

Dynamic HTML is not really a new specification of HTML, but rather a new way of looking at and controlling the standard HTML codes and commands. When thinking of dynamic HTML, you need to remember the qualities of standard HTML, especially that once a page is loaded from the server, it will not change until another request comes to the server. Dynamic HTML gives you more control over the HTML elements and allows them to change at any time, without returning to the Web server.

### **There are four parts to DHTML:**

- Document Object Model (DOM)
- Scripts
- Cascading Style Sheets (CSS)
- HTML

#### **DOM (Will be studied in detail in JavaScript)**

The DOM is what allows you to access any part of your Web page to change it with DHTML. The DOM specifies every part of a Web page and using its consistent naming conventions you can access them and change their properties.

#### **Scripts**

Scripts written in either JavaScript or VBScript

#### **Cascading Style Sheets (CSS)**

CSS is used in DHTML to control the look and feel of the Web page. Style sheets define the colors and fonts of text, the background colors and images, and the placement of objects on the page. Using scripting and the DOM, you can change the style of various elements.

#### **HTML**

HTML 4.x is used to create the page itself and build the elements for the CSS and the DOM to work on. There is nothing special about HTML for DHTML - but having valid HTML is even more important.

#### **Features of DHTML**

These are most common features of DHTML:

- **Changing the tags and properties**
  - This is one of the most common uses of DHTML. It allows changing the qualities of an HTML tag depending on an event outside of the browser (such as a mouse click, time, or date, and so on). With this information can be preloaded but not displayed until the reader clicks on a specific link.
- **Real-time positioning**
  - When most people think of DHTML this is what they expect. Objects, images, and text moving around the Web page. This can allow you to play interactive games with your readers or animate portions of your screen.

## Practical Exercises:

Type the following codes in a notepad and save it as a html page to see the output. Here in this example we have used tables to divide the content of web page into different sections.

### Question 1:

```
<html>
<head>
<title>WEB PAGE TITLE GOES HERE</title>
</head>

<body style="margin: 0px; padding: 0px; font-family: 'Trebuchet MS', verdana;">
<table width="100%" style="height: 100%;" cellpadding="10" cellspacing="0"
border="0"><tr>
<!-- ===== LEFT COLUMN (MENU) ===== -->
<td width="20%" valign="top" bgcolor="#999f8e">
<a href="#">Menu link</a><br>
<a href="#">Menu link</a><br>
<a href="#">Menu link</a><br>
<a href="#">Menu link</a><br>
<a href="#">Menu link</a>
</td>
<!-- ===== RIGHT COLUMN (CONTENT) ===== -->
<td width="80%" valign="top" bgcolor="#d2d8c7">

<h1>Website Logo</h1>
<h2>Page heading</h2>
This is a basic two-column web page layout. The left column or the <i>menu
column</i> is a narrow band of space (usually between 15-25% of the page width)
and is reserved for a menu of hyperlinks leading to other pages on your
website. The table used to create this layout employs a single table row
containing two table cells.<br>
<br>
The right column or the <i>content column</i> takes up the lion's share of the
web page width and contains the actual content of each particular page. In a
basic two column layout like this, it is common to place the website logo at
the top of the content column on each page.</td></tr></table>
</body>
</html>
```

### Question 2:

```
<html>
<head>
<title>WEB PAGE TITLE GOES HERE</title>
</head>
```

*Note: This handout is for simple reference only. Do not completely depend on it.*

```

<body style="margin: 0px; padding: 0px; font-family: 'Trebuchet MS', verdana;">
<table width="100%" style="height: 100%;" cellpadding="10" cellspacing="0"
border="0">
<tr>
<!-- ===== HEADER SECTION ===== -->
<td colspan="2" style="height: 100px;" bgcolor="#777d6a"><h1>Website
Logo</h1></td></tr>

<tr>
<!-- ===== LEFT COLUMN (MENU) ===== -->
<td width="20%" valign="top" bgcolor="#999f8e">
<a href="#">Menu link</a><br>
<a href="#">Menu link</a><br>
<a href="#">Menu link</a><br>
<a href="#">Menu link</a><br>
<a href="#">Menu link</a>
</td>

<!-- ===== RIGHT COLUMN (CONTENT) ===== -->
<td width="80%" valign="top" bgcolor="#d2d8c7">
<h2>Page heading</h2>

Here's a two column layout with a header section that spans the width of both
columns. The first table row creates the header and contains a single table
cell which uses the colspan="2" attribute-value pair. The website logo
typically goes in the header section.<br>
<br>
The second table row contains two table cells which create the menu column
(left) and the content column (right). The colspan attribute is not set in
either so they default to colspan="1".</td></tr></table>
</body>
</html>

```

**Question 3:**

```

<html>
<head>
<title>WEB PAGE TITLE GOES HERE</title>
</head>
<body style="margin: 0px; padding: 0px; font-family: 'Trebuchet MS', verdana;">
<table width="100%" style="height: 100%;" cellpadding="10" cellspacing="0"
border="0">
<tr>
<!-- ===== HEADER SECTION ===== -->
<td colspan="2" style="height: 100px;" bgcolor="#777d6a"><h1>Website
Logo</h1></td></tr>

```

```

<tr>
<!-- ===== LEFT COLUMN (MENU) ===== -->
<td width="20%" valign="top" bgcolor="#999f8e">
<a href="#">Menu link</a><br>
<a href="#">Menu link</a><br>
<a href="#">Menu link</a><br>
<a href="#">Menu link</a><br>
<a href="#">Menu link</a>
</td>

<!-- ===== RIGHT COLUMN (CONTENT) ===== -->
<td width="80%" valign="top" bgcolor="#d2d8c7">
<h2>Page heading</h2>

Here's a two column layout with header and footer sections that span the width
of both columns. The first table row creates the header and contains a single
table cell which uses the colspan="2" attribute-value pair.<br>
<br>
The second table row contains two table cells which create the menu column
(left) and the content column (right). The colspan attribute is not set in
either so they default to colspan="1".<br>
<br>
The third table row creates the footer. Like the header, it contains a single
table cell which uses the colspan="2" attribute-value pair.</td></tr>

<!-- ===== FOOTER SECTION ===== -->
<tr><td colspan="2" align="center" height="20" bgcolor="#777d6a">Copyright
©</td></tr>
</table>
</body>
</html>

```

#### Question 4

```

<!-- save the file as frame.html-->
<frameset rows="75%, *" cols="*, 40%">
  <frame src="framea.html">
  <frame src="frameb.html">
  <frame src="framec.html">
  <frame src="framed.html">

  <noframes>
    <h1>Your browser does not supports frames</h1>
    Click the below link to continue your visit.
    <a href="noframes.html">no-frames</a>
  </noframes>

```



```
</frameset>  
</html>
```

```
<!-- save the file as framea.html-->  
<html>  
<head>  
<title> framea</title>  
</head>  
<body>  
<h1>this is framea</h1>  
</body>  
</html>
```

```
<!-- save the file as frameb.html-->  
<html>  
<head>  
<title> frameb</title>  
</head>  
<body>  
<h1>this is frameb</h1>  
</body>  
</html>
```

```
<!-- save the file as framec.html-->  
<html>  
<head>  
<title> framec</title>  
</head>  
<body>  
<h1>this is framec</h1>  
</body>  
</html>
```

```
<!-- save the file as framed.html-->  
<html>  
<head>  
<title> framed</title>  
</head>  
<body>  
<h1>this is framed</h1>  
</body>  
</html>
```

~