

Unit-II

Chapter

4

## Introduction to HTML

HTML is the building block for web pages. Hyper Text Markup Language is the central core on which all Web pages are built. HTML was invented by Tim Berners-Lee. HTML is a language for describing web pages. HTML stands for Hyper Text Markup Language.

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

HTML is not a programming language, it is a markup language. A markup language is a set of markup tags. The purpose of the tags are to describe page content. HTML is the language interpreted by browser. Web pages are also called HTML documents. HTML is a set of special codes that can be embedded in text to add formatting and linking information. HTML is specified as TAGS in an HTML documents.

Basically an HTML document is a plain text file that contains text and nothing else. When a browser opens an HTML file, the browser will look for HTML codes in the text and interprets them in to Graphical layout. As HTML documents are text files they can be written in even the simplest text editor.

There are many popular HTML editor. Notepad is the default Windows text editor. On most Windows systems, click your Start button and choose Programs then Accessories. HTML is just a series of tags that are integrated into a text document. They are give directions silently to the browser that what to do, and what props to use.

HTML tags are usually English words or abbreviations (such as "p" for paragraph), but they are distinguished from the regular text because they are placed in small angle brackets (<>).

The definition of HTML is **HyperText Markup Language**.

- *HyperText* is the method by which you move around on the web — by clicking on special text called **hyperlinks** which bring you to the next page. The fact that it is *hyper* just means it is not linear — i.e. you can go to any place on the Internet by clicking on links — there is no set order to do so.
- *Markup* is what **HTML tags** do to the text inside them. They mark it as a certain type of text (*italicised* text, for example).
- HTML is a *Language*, as it has code-words and syntax like any other language.

### **1. HTML FEATURES**

- HTML tags are not case sensitive, <b> means the same as <B>..
- Complete code in <html></html> tag.
- Browser is an interpreter for the HTML not compiled.
- Complete source code is transferred to client from the server so there is no security of code.

- Client can view the complete code by right clicking on the web page and choose view source.
- We can use any text editor to write HTML code such as notepad or Notepad or Dreamweaver.
- Saved with extension .htm or .html.
- To execute html file double click on the html file or write its complete path in browser address bar.

HTML tags are used to mark-up HTML elements.

- 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

## 1.2 HTML Document

HTML is a formal that tells a computer to display a web page. HTML file is also known HTML document. The documents themselves are plain text files with special "tags" or codes that a web browser uses to interpret and display information on your computer screen. Html file are saved with .html or .htm extention.

<html> tag tells your browser that this is the start of an html document. The last tag in your document is </html>. This tag tells your browser that this is the end of the html document.

All tags are resides within the <HTML> tags so <HTML> tag is container of all tags or it is parent tags of all tags. <html> tags ensures that the beginning or the end of the document. Inside the <html> tag and its end tag are the document's head and body.

An HTML document has two distinct parts-a **head** and a **body**.

**1.2.1 The head Part:**  
Head of the HTML document is where you enter the title of the page. The head portion is enclosed by the <head> and </head> tags. It contains information about a page that is visible on the page itself, such as the title. The <head> tag serves to encapsulate other header tags. Place it at the beginning of your document, just after the <html> tag and before the <body> or <frameset> tag. It is good programming practice to include <head> in all documents, since they promote readability and support document automation.

The text between the <head> tag and the </head> tag is header information. Header information is not displayed in the browser window. The text between the <title> tags is the title of your document. The <title> tag is used to uniquely identify each document and is also displayed in the title bar of the browser window.

The <title> tag is the only thing required within the <head> tag. Since the <head> tag itself and even the <html> tag can safely be omitted, the <title> tag could be the first line within a HTML document. Beyond that, most browsers will even supply a generic title for documents lacking a <title> tag, such as the document's filename, so you don't even have to supply a title. But it not a good programming habit to not use <title> tag.

Browsers do not specifically format title text, and they ignore anything other than text inside the title start and end tags. For instance, they will ignore any images or links to other documents to create the head portion of HTML document and to give the document a title, type the following in your text editor:

```
<HEAD>
<TITLE>My First Page</TITLE>
</HEAD>
```

This tells a Web browser what information in the title tag is the name of document in the document in the browser window.

## 1.2.2 The body Part:

Body is where all most all available, text, headlines, graphics, and all Web goodies. The body - enclosed by <body> and </body> - is where the action is. The <body> tag has a number of attributes that control the color and background of your document. The text between the <body> tags is the text that will be displayed in your browser.

Various browsers have extended the tag to give even greater control over your document's appearance.

Anything between the <body> tag and its ending counterpart, </body>, is called *body content*. The simplest document might have only a sequence of text paragraphs within the <body> tag. More complex documents might include heavily formatted text, graphical figures, tables, and a variety of special effects.

Like the <html> and <head> tags, it is also good habit that you include the <body> tags in HTML documents, too, to make them more easily readable and maintainable. Everything that appears on the page is contained within these tag. To add the body section, start after the </HEAD> tag, and enter the following:

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

Between these two tags, you'll eventually enter the rest of the text and graphics for your Web page.

Above the first <HEAD> tag, enter the following:

```
<HTML>
After the last </BODY> tag, type the following:
```

The BODY tag defines the main body of an HTML document. As we discussed earlier also. Here we explains some attributes also. Its attributes enable the HTML author to define colors for text and links, as well as background colors or patterns for the document. In addition, as you have already learned, there are two event handlers, onLoad and onUnload, that can be used in the BODY tag. Details of events methods is discussed in later chapter. The following is a list of available attributes for the BODY tag:

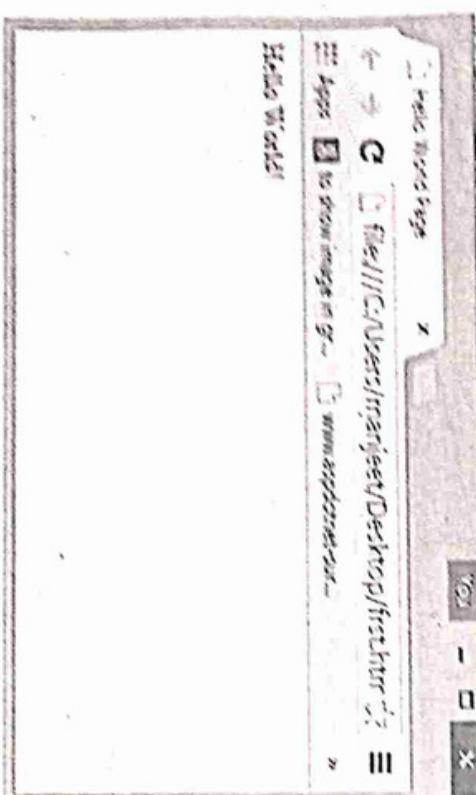
- BACKGROUND-Specifies the URL of a background image.
- BGCOLOR-Specifies a background color for the document as a hexadecimal RGB triplet or a color name.
- FORCOLOR-Specifies the foreground (and text) color as a hexadecimal triplet or color name.
- LINK-Specifies the color for links as a hexadecimal triplet.
- ALINK-Specifies the color for an active link (when the user has the mouse clicked on a link until the user releases the mouse button) as a hexadecimal triplet.
- VLINK-Specifies the color for a followed link as a hexadecimal triplet.

If we are using both properties BACKGROUND (for background image) and BGCOLOR (for background color), then priority is given to BACKGROUND (image) and image will appear in page. If image is in correct or wrong image name in that case (alternatively) BGCOLOR (background color) is reflected.

#### Example: Creating an HTML Template

Let's take what you know and create a template. By saving this text file with .htm or .html extension using File → Save → to save it as your new Web page.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello World Page</title>
  </head>
  <body>
    Hello World!
  </body>
</html>
```



**Paired tags** are also known as **container tags**. Container tags always have the following form: **<TAG>** start being formatted or defined **</TAG>**

Paired tags have both opening and closing like **<html>** and **</head>**. Those tag which have only opening but not need to close them are known as singular tags like **<br>** tag. Singular tags are also known as Empty tags.

In HTML there are both logical tags and physical tags. Logical tags are designed to describe the enclosed text's meaning. An example of a logical tag is the **<strong></strong>** tag. By placing text between these tags you are telling the browser that the text has some greater importance. By default all browsers make the text appear bold when in between the **<strong>** and **</strong>** tags. Physical tags on the other hand provide specific instructions on how to display the text they enclose. Examples of physical tags include:

- <b>**: Makes the text bold.

- <i>**: Makes the text usually one size bigger than what's around it.

- <u>**: Makes text italic.

Physical tags used to add style to HTML pages because style sheets were not around, though the original intention of HTML was to not have physical tags. Rather than use physical tags to style your HTML pages, you should use style sheets.

## 2.1 Tag Attributes

Attributes Tags can have attributes. Attributes can provide additional information about the HTML elements on your page. The **<tag>** tells the browser to do something, while the attribute tells the browser how to do it with extra feature. The start tag (opening) may contain additional information, as in the preceding example. Such information is called an attribute. Attributes usually consist of 2 parts:

- An attribute name
- An attribute value

Note that the attribute name precedes an equal sign (=). Following the equal sign is the desired value for this attribute. The value must always be enclosed in quotation marks.

A few attributes can only have one value. They are Boolean attributes and may be shortened by only specifying the attribute name or leaving the attribute value empty. The value must be written within quotation marks. Both single quotes (' ) and double quotes (" ) are allowed. Many developers prefer to always use quotes to make the code less ambiguous to the eye and to avoid mistakes and to increase the readability of code.

Attribute values should always be enclosed in quotes. Double style quotes are the most common, but single style quotes are also allowed. In some rare situations, like when the attribute value itself contains quotes, it is necessary to use single quotes:

For instance, if we add the bgcolor attribute, we can tell the browser that the background color of your page should be blue.

Example: **<body bgcolor="blue">**.

## 2. HTML Tags

HTML tags are used to mark-up HTML elements and are surrounded by the two characters < and > are called angle brackets. Tags normally in pairs like <b> and </b> are known as paired tags.

as the first line. The doctype declaration is not an HTML tag, it is an instruction to the web browser about what version of HTML the page is written in.

The doctype has a long and intricate history, but for now need to know is that this doctype tells the browser to interpret the HTML and CSS code according to W3C standards.

#### The Comment Tag

HTML comments that are not displayed when the page is rendered in a browser. This is useful for explaining a section of markup, leaving notes for other people who might work on the page, or for leaving reminders for yourself. HTML comments are enclosed in symbols as follows:

<!-- This is comment text -->

Web browser ignore anything the tag contains. That can be text, hypertext links, image links, even small scripts and programs.

The comment tag ignores returns. Generally, use the comment tag to mark a point in a particular HTML document where need to remember to update some text, or to explain a particularly confusing part of your page.

### 2.3 FORMATTING HTML TAGS

As mentioned before, there are logical styles that describe what the text should be and physical styles which actually provide physical formatting. It is recommended to use the logical tags and use style sheets to style the text in those tags.

Logical Tags	Physical Tags	Tag	Description
<abbr>			Defines an abbreviation
<address>			Defines an address element
<code>			Defines computer code text
<blockquote>			Defines a long quotation
<del>			Defines text
<dfn>			Defines a <i>definition</i> term
<ins>			Defines inserted text
<kbd>			Defines keyboard text
<pre>			Defines preformatted text
<q>			Defines a short quotation
<samp>			Defines sample computer code
<strong>			Defines <b>strong</b> text
<var>			Defines a <i>variable</i>
<b>			Defines <b>bold</b> text
<big>			Defines <b>big</b> text
<i>			Defines <i>italic</i> text
<small>			Defines <i>small</i> text
<sup>			Defines <sup>superscripted</sup> text
<sub>			Defines <sub>subscripted</sub> text
<tt>			Defines <i>teletype</i> text
<u>			Deprecated. Use styles instead

### INTRODUCTION TO HTML

Character tags like <strong> and <em> produce the same physical display as <b> and <i> but are more uniformly supported across different browsers.

#### 2.4 Explicit Styles Tag

Explicit tags are also often called physical tags, since they very specifically tell the Web browser how you want the text to physically appear. The browser is given no choice in the matter. The basic explicit tags are containers that let the user mark text as bold, italic, or underlined.

##### Tags Meaning

<B>, </B> Bold text

<I>, </I> Italic text

<U>, </U> Underlined text

With these tags, the browser has no choice—it must either display the text as defined or, if it can't do that, then it must add no emphasis to the text.

### 2.5 Implicit HTML Tags

Implicit styles are often called logical styles, since they allow the browser some freedom in how it will display the text. These tags, like the header tags, are generally relative to one another, depending on the browser being used to view them. Some of the common implicit (logical) tags

<STRONG>

Strong emphasis Bold text

#### 2.6 PROGRAMMER'S HTML TAGS

Programmer Tags which use in to make easier to present computer-programming codes. Those tags are listed below.

<CODE>, <CODE> : Programming lines Monospaced.

<KBD>, <KBD>: Keyboard text Monospaced

<SAMP>, <SAMP>: Sample output Monospaced

<VAR>, <VAR>: Variable Italic

Using these tags is a great way to internally document your HTML pages. This will help you later when you return to the document to update it or fix errors—especially as the document becomes more complex.

#### 2.7 The <PRE> (preformatted text)

The <PRE> (preformatted text) tag is designed to allow us to keep the exact spacing and returns that what put between the on and off tags.

Using the <P> tag, except that it will be in a mono spaced font.

<PRE> tags is use to create extra lines in a document without typing any text between them. <PRE> tag doesn't allow the browser screen to wrap text automatically; users need to expand their browser window if we use particular long lines within a <PRE> container. So its programmer's responsibility to makes lines of text are short so that all browsers can view them without scrolling.

## 2.8 Heading Tag `<h1 ... h6>`

Header tags are used to provide heading to the text. There are six header tags `h1, h2, h3, h4, h5` and `h6`. All header tags are paired tags. Ranging from level 1 to level 6, headers allow you to create different levels of emphasized headlines to help you organize your documents. All header tags makes the enclosing text bold and also increase the font size of enclosing text. The size range is decrease from `h1` to `h6`. So as number is increases from 1 to 6, size of heading font is decreases.

All header tags do one more thing that is break the line without using `<br>` tag like `<p>` tag.

Headings are defined with the `<h1>` to `<h6>` tags. `<h1>` defines the largest heading while `<h6>` defines the smallest.

HTML automatically adds an extra blank line before and after a heading. A useful heading attribute is `align`.

```
<h5 align="left">I can align headings <h5>
<h5 align="center">This is a centered heading <h5>
<h5 align="right">This is a heading aligned to the right <h5>
```

## 2.9 Line Breaks `<br>` tag:

The `<br>` tag is used when you want to start a new line, but don't want to start a new paragraph. The `<br>` tag forces a line break wherever you place it. It is similar to single spacing in a document.

The `<br>` tag has no closing tag. So `<br>` tag is also a singular tag or empty tag. Singular should be write like `<br />`. It is new rule for new browsers. It is not must but it should be follow.

## 2.10 Named Character References:

Some characters have a special meaning in HTML, like the less than sign (`<`) that defines the start of an HTML tag. If we want the browser to actually display these characters we must insert character entities in place of the actual characters themselves. Named character reference is also called entities.

Entity are used to print characters that have a special meaning in HTML. For example, HTML interprets the less-than and greater-than symbols as tag delimiters. If you want to display a greater-than symbol in the text, you can use a named character reference.

A character entity has **three parts**:

- An ampersand (&)
- An entity name or an entity number
- And finally a semicolon (;).

The & means we are beginning a special character, the ; means ending a special character and the letters in between are sort of an abbreviation for what it's for.

To display a less than sign in an HTML document we must write: `&lt;` 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, while the support for entity numbers is very good in almost all browsers.

**Note:** Entities are case sensitive.

### Special character codes

| Symbol | Meaning             | Character  |
|--------|---------------------|--|
| <      | Less-than symbol    | &lt;   |
| >      | Greater-than symbol | &gt;   |
| &      | Ampersand ("and")   | &amp;  |
| "      | Quotation marks     | &quot;   |
| ,      | apostrophe          | &apos; (not work in IE)  |
| ,      | Non-breaking space  | (Inserts an extra space. Often used to create indentations in a paragraph, or create additional spaces between words.) |

Example including all above tags:

```
<html>
<head>
<title>My First Page</title>
</head>
<body bgcolor="pink">
<b>This is BOLD</b>
<strong>This is STRONG</strong><br/>
<i>This is ITALIC</i><br/>
<u>This is UNDERLINE</u><br/>
<strike>This is STRIKE</strike><br/>
<s>This is Strikeout TEXT</s><br/>
<code>This is code</code><br/>
<var>This is var</var><br />
<hl>This is HEADING 1</hl>
<h2>This is HEADING 2</h2>
<h3>This is HEADING 3</h3>
<h4>This is HEADING 4</h4>
<h5>This is HEADING 5</h5>
<h6>This is HEADING 6</h6>
<!-- This is comment -->
A text with multiple spaces but will display only one space:Hello!! World<br/>
A text with multiple spaces and will display multiple spaces:-
Hello!!&nbsp;&nbsp;&nbsp;&nbsp;World<br/>
```

The tag for bold is &lt;b&gt;</b>

To print space we use &nbsp;&nbsp;<br>

This is quotation mark &quot;<br>

<pre>

Matrix      Computers

Jupiter

</pre>

</body>

</html>



### This is HEADING 1

This is HEADING 2

This is HEADING 3

This is HEADING 4

More options

A text with multiple spaces but will display only one space -Hello! World

A text with multiple spaces and will display multiple spaces -Hello! World

The tag for bold ->b>

To print space we use &nbsp;

To print a new line ->\n<

Save...

Print...

### 3. PARAGRAPH Tag <p>

<p>

a paragraph. In order to give the appearance of paragraphs, then, you have to use the paragraph container tag.

Paragraphs are defined with the <p> tag. Think of a paragraph as a block of text. A browser ignores any indentations or blank lines in the source text. Without <p> elements, the document becomes one large paragraph. HTML automatically adds an extra blank line before and after a paragraph <p> tag. You can use the align attribute with a paragraph tag as well.

</p>

</head>

<title>My Second Page</title>

</head>

<body bgcolor="cyan" text="red">

<p>This is paragraph1. This is paragraph1.

This is paragraph1. This is paragraph1.

<p>

<p align="right">This is paragraph2. This is paragraph2.

This is paragraph2. This is paragraph2.

<p>

<p align="center">This is paragraph3. This is paragraph3.

This is paragraph3. This is paragraph3. This is paragraph3.

</p>

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

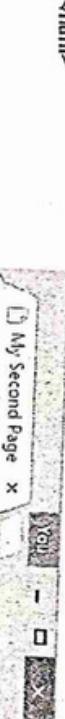
This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.

This is paragraph4. This is paragraph4. This is paragraph4.



## 4. Horizontal tag Graphical elements

### Horizontal Ruler<HR>

This tag is used to draw a horizontal line across the page. To add a horizontal rule to your page, insert <hr> tag at the position where you want the line to appear. In HTML, <hr> is an empty tag, requiring closing tag. By default, these lines include a 3-D shadow effect, which can be removed. In addition, lines can be set to various sizes, widths and alignments.

### 4.1 Horizontal rule ATTRIBUTES

The <hr> element has a width attribute that controls how far the line extends across the screen. By default, the value of the width attribute is 100 percent. If you want the line to extend across only a percent of the window, you would write the tag as follows:

<hr width="50%">

#### :Horizontal rule (<hr>) attributes

Attribute Name	Accepted Values
<b>align</b>	<ul style="list-style-type: none"> <li>• "left"</li> <li>• "right"</li> <li>• "center"</li> </ul> <p>If no alignment is specified, the default value for this attribute is center.</p>

## 5. HTML Fonts

The <font> tag in HTML is deprecated. The World Wide Web Consortium (W3C) has removed the <font> tag from its recommendations. In future versions of HTML, style sheets (CSS) will be used to define the layout and display properties of HTML elements.

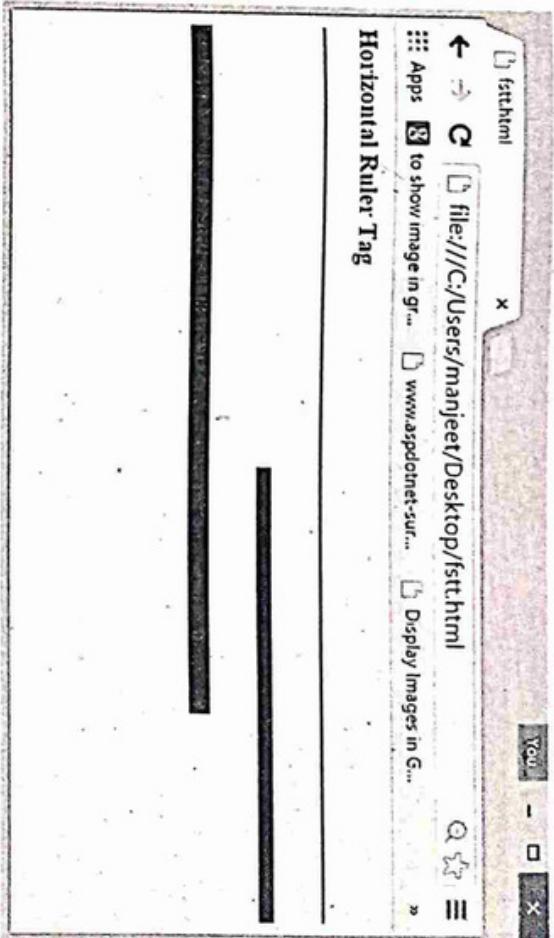
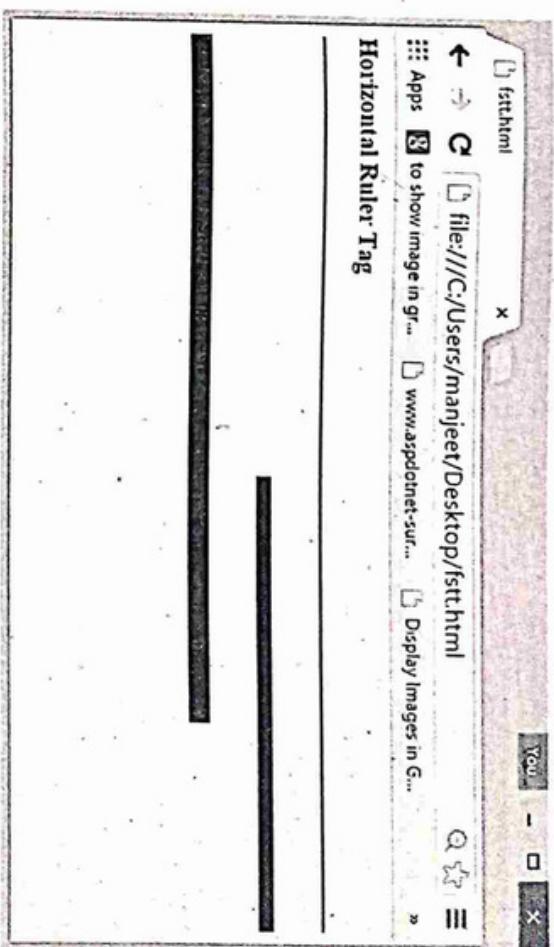
The <font> Tag Should NOT be used.

The <font> tag supports three attributes:

- SIZE
- COLOR
- FACE

```
<html><body><hr color="red"><br/>
<hr align="right" width=50% size=10 color="red"><br/>
<hr align="left" width=50px size=15 color="red"><br/>
</body></html>
```

Align attribute value can be provided in pixels also to have fix size line.



#### Example:

```
<font face="arial" size=5 color="#cc77aa>Matrix Computers</font><br>
```

## 6. HTML Backgrounds

### Backgrounds

The `<body>` tag has two attributes where you can specify backgrounds. The background can be a color or an image.

### Bgcolor

The `bgcolor` attribute specifies a background-color for an HTML page. The value of this attribute can be a hexdecimal number, an RGB value, or a color name:

```
<body bgcolor="#000000"><body bgcolor="rgb(0,0,0)"><body bgcolor="black">
```

The lines above all set the background-color to black.

### Background

The background attribute can also specify a background-image for an HTML page. The value of this attribute is the URL of the image you want to use. If the image is smaller than the browser window, the image will repeat itself until it fills the entire browser window.

```
<body background="clouds.gif"><body background="http://profdevtrain.austincce.edu/html/graphics/clouds.gif">
```

The URL can be relative (as in the first line above) or absolute (as in the second line above).

If you want to use a background image, you should keep in mind:

- Will the background image increase the loading time too much?
- Will the background image look good with other images on the page?
- Will the background image look good with the text colors on the page?
- Will the background image look good when it is repeated on the page?
- Will the background image take away the focus from the text?

### Background images and Background colors in body

You can specify both images and background images in a Web page. In fact, it can be advantageous to specify both, in case a background image becomes unavailable for some reason.

If you use the `bgcolor` and `background` attributes in the same `<body>` tag, then only the attribute given last in the tag will be rendered. If you use a style sheet and specify both image and color as a background, then the background image will always render first. If the image cannot be found, a background color will then appear. All values specified in style sheets will override anything specified in the HTML or XHTML itself.

## 7. HTML Colors

### Color Values

All colors are made by combination ratio of red, blue and green color. Colors are defined using a hexadeciml notation for the combination of red, green, and blue color values (RGB). The lowest value that can be given to one light source is 0 (hex #00). The highest value is 255 (hex #FF). This table shows the result of combining red, green, and blue. First ratio is for red color and second ratio is for green color and last ratio is for blue color.

"#000000" In this combination all three color ratio is at lowest point. Means Brightness of all three colors is lowest so dullest color is produced by this combination that is black.

"#FFFFFF" In this combination all three color ratio is at highest point. Means Brightness of all three colors is maximum so brightest color is produced by this combination that is white.

"#00FF00" In this combination red and blue color ratio is at lowest point and green is at highest. So green color is produced by this combination.

A few years ago, when most computers supported only 256 different colors. The combination of Red, Green and Blue values from 0 to 255 gives a total of more than 16 million different colors to play with ( $256 \times 256 \times 256$ ).

## 8. HTML Lists

List are of two types ordered list and unordered list. HTML provides two tags to create list.

Ordered Lists are numbered (ordered) and unordered list bulletted (unordered) lists.

For numbered/ordered lists, the tag is `<OL>`, and for bulletted/unordered lists, the tag is `<UL>`. For either of these lists, a line item is designated with the empty tag `<LI>`. In the case of ordered lists, the `<LI>` tag inserts a number; for unordered lists, it inserts a bullet point.

### 8.1 UNORDERED LISTS `<ul>` Tag

An unordered list is a list of items marked with bullets (typically small black circles). An unordered list starts with the `<ul>` tag. Each list item starts with the `<li>` tag.

An unordered list is a collection of related items that have no special order or sequence. Inside a list item you can put paragraphs, line breaks, images, links, other lists, etc.

The `<ul>` tag signals to the browser that the following content, ending with the `</ul>` tag, is an unordered list of items. Inside, each item in the unordered list is identified by a leading `<li>` tag. `<li>` is child tag of `<ul>` tag.

#### 8.1.1. Attribute of `<ul>` Tag

##### • type

This attribute is used to define the type to specify which bullet symbol you required in an unordered list. This attribute may have a value of either disc, circle, or square.

With the advent of standard Cascading Style Sheets, the W3C has deprecated the `type` attribute. There is not start attribute in unordered list.

### 8.2 ORDERED LISTS

An ordered list is also a list of items. The list items are with numbers. An ordered list starts with the `<ol>` tag. Each list item starts with the `<li>` tag.

Use an ordered list when the sequence of the list items is required. `<ol>` tag is used for ordered list. The typical browser formats the contents of an ordered list just like an unordered list, except that the items are numbered instead of bulletted. The numbering starts at one and is incremented by one for each successive ordered list element tagged with `<li>`. `<li>` is child tag of `<ol>` tag.

### 8.2.1 Attribute of <ol> tag

**Start**) The start attribute for the <ol> tag lets you change the starting value of list. If we do not start numbering with start attribute, then by default numbering start from one.

To start numbering a list at 7, for example:

```
<ol start="7">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Coca Cola</li>
  <li>Java</li>
</ol>
```

**Type attribute** By default, lists sets number ordered list items with a sequence of Arabic numerals.

If needed bring able to start the sequence of some number other than 1, you can use the type attribute with the <ol> tag to change the numbering style itself. The different values of type attribute are given below:

```
<ol type="A">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

for numbering with capital letters,

for numbering with lowercase letters,

for lowercase Roman numerals,

for common Arabic numerals, it default value also.

```
<ol>
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```



```
<ol type="I">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="a">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="i">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="l">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="o">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="A">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="a">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="I">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="i">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="a">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="l">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="o">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="A">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="a">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="I">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="i">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="a">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="l">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="o">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="A">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="a">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="I">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="i">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="a">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="l">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="o">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="A">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="a">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="I">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="i">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="a">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

```
<ol type="l">
  <li>Water</li>
  <li>Coffee</li>
  <li>Coca Cola</li>
  <li>Java</li>
```

The <li> tag is child tag of both <ol> tag and <ul> tag. The <li> tag defines an item in a list. It's the universal tag for list items in ordered (<ol>) and unordered (<ul>) lists.

<li> tag also have type attribute which works same as in <ol> tag. There is another attribute value which assign the value to list item.

The value attribute changes the numbers of a specific list item and all of the list items that follow it. Since the ordered list is the only type with sequentially numbered items, the value attribute is valid only when used within an <li> tag inside an ordered list.

```
<ol type="A">
  <li>Changing the numbering type</li>
  <li>Uppercase Roman numerals</li>
  <li>Lowercase Roman numerals</li>
  <li>ASP.NET</li>
</ol>
<ul type="a">
  <li>C</li>
  <li>C++</li>
  <li>Java</li>
</ul>
<ul type="i">
  <li>Liquid
    <ul type="l">
      <li>Tea</li>
      <li>Water</li>
      <li>Java</li>
    </ul>
  <li>Fruits
    <ul type="l">
      <li>Banana</li>
      <li>Apple</li>
      <li>Mango</li>
    </ul>
  </li>
</ul>
```

```
<li>Water</li>
<li>Coffee</li>
<li>Coca Cola</li>
<li>Java</li>
```

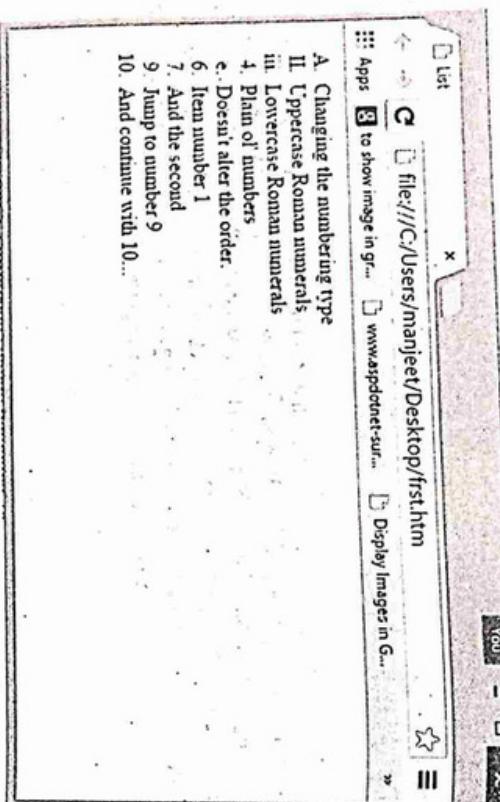
## INTRODUCTION TO HTML

<DT>Eggs

<DT>Cereal

</DL>

Use &lt;DL&> and &lt;DD&> Tags



## 8.4 Definition Lists

The final list tag is the definition list, which is designed to allow for two levels of list items, originally conceived to be the defined term and its definition. Definition lists consist of two parts: a **term** and a **description**.

The definition List tags are:

- The container tag <DL> (**definition list**).
- And two empty tags:

<DT> (**definition term**): The <DT> tag is use to fit on a single line of your Web page although it will wrap to the beginning of the next line if necessary.

<DD> (**definition**): The <DD> tag will accept a full paragraph of text, continuously indented beneath the <DT> term.

The definition list offers some additional flexibility over the standard lists, giving you more choices.

### Definition Lists

**Example:**

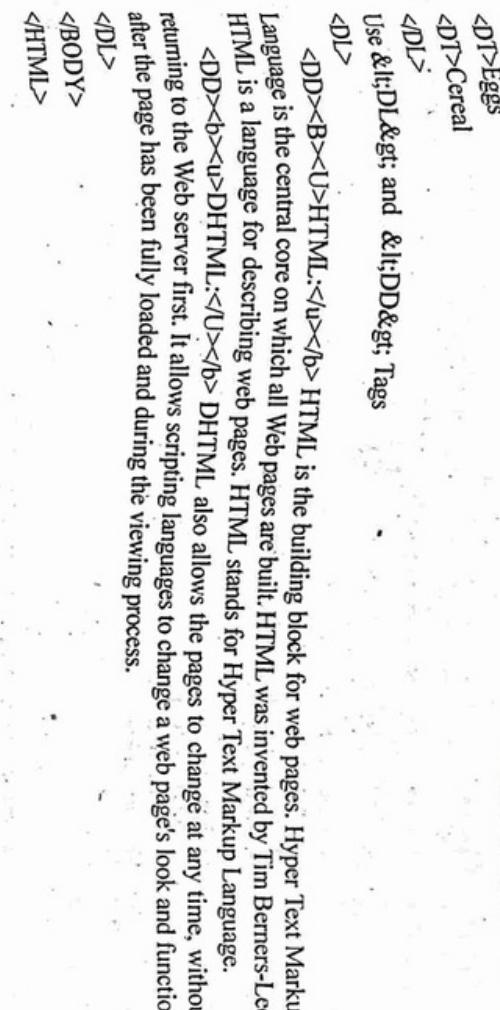
```

<HTML>
<HEAD>
<TITLE>List</TITLE>
</HEAD>
<BODY bgcolor="pink">
  <DL>
    <DT>Milk</DT>
    <DT>Honey</DT>
    <DT>Eggs</DT>
    <DT>Cereal</DT>
  </DL>
</BODY>

```

**HTML:** HTML is the building block for web pages. Hyper Text Markup Language is the central core on which all Web pages are built. HTML was invented by Tim Berners-Lee. HTML is a language for describing web pages. HTML stands for Hyper Text Markup Language.

**DHTML:** DHTML also allows the pages to change at any time, without returning to the Web server first. It allows scripting languages to change a web page's look and function after the page has been fully loaded and during the viewing process.



## 9. HTML Links

HTML uses the `<a>` anchor tag to create a link to another document or web page. Links are the most important part of the Web. It provide ability to Web to link from one document to another specific site.

`<A HREF="URL">Text describing link</A>`

The words Text describing link would appear underlined and in another color to indicate that clicking that text initiates the hypertext link.

There are two types of links, **absolute** and **relative**. All link tags are container tags.

An absolute link that tells exactly where a browser should go by following all the necessary paths. They are used when you link from one Web site to another. Relative links (sometimes called local links) just tell the shortcuts. The following is an example of a relative link:

`<A HREF="products.html">Our Product Information</A>`

Here product.html file is available in that folder from where we call this file.

If the HTML document to which you want a link is located elsewhere on the Internet, you need absolute URL, a complete URL as the following:

`<A HREF="http://www.bignerdranch.com/products.html">Our Product Information</A>`

In either case, things end up looking the same in a browser.

### 9.1 Attribute

#### 9.1.1 HrefAttribute:

The `<a>` tag is used to create an anchor to link from, the href attribute is used to tell the address of the document or page we are linking to, and the words between the open and close of the anchor tag will be displayed as a hyperlink.

External link are created by using of href attribute of anchor `<a>` tag. External links are those links by clicking that control goes to another page.

An anchor can point to any type of resource on the Web like an HTML page, an image, a sound file, a movie, etc. The syntax of creating an anchor:

`<a href="#">Text to be displayed</a>`

#### 9.1.2 Target Attribute

The target attribute, whether the linked document will be opened. Means targeted document will open in same browser window or in different window. By default, the link will open in the current window. Target attribute can have following values:

`target="_blank"` : Open in a new blank window.

`target="_self"` : Open in current window.

`target="_parent"` : Open in current window.

#### 9.1.3 Name Attribute

The name attribute is used to create a named anchor and to create internal link. Internal link means when we click on hyperlink then control transfer to same page rather than another page.

This can be done by using name attribute of `<a>` tag. When using named anchors we can create links that can jump directly to a specific section on a page, instead of letting the scroll around to find. A named anchor doesn't change the appearance of the text unless you set styles for that anchor.

The syntax of a named anchor:

`<a name="top">Text to be displayed</a>`

To link directly to the top section, add a # sign and the name of the anchor to the end of a URL, like this:

```
<a href="#top">Top of Page 5</a>
```

Example:

```
<html>
<head>
<title>My First Page</title>
</head>
<body bgcolor="cyan" text="red" link="blue" alink="yellow" style="color:purple">
<a href="xyz">Top</a><br>
<a href="second.html" name="xyz" target="_self">Link Color</a><br>
<a href="second.html" target="_self">Click to start next page</a><br>
<a href="second.html" target="_self">Active Link Color</a><br>
<a href="first.html" target="_self">Visited Link Color</a><br>
<a href="second.html" target="_self">Link Color</a><br>
<a href="#">Go Top</a><br>
</body>
</html>
```



## 10. <img> tag for IMAGE in Web PAGES

The <img> tag displays a graphic image on your page. The key attribute that is required in this tag is src (abbreviation for source). You use the src attribute to specify the name and, if necessary, the location of your image file.

The <img> tag is an empty tag, so it requires no closing tag. In XHTML, however, you must close the tag. <img> tag can use in either of two ways in XHTML:

```
<imgsrc="imagefile.gif"></img>
```

```
<imgsrc="imagefile.gif"/>
```

Images can be big or small; they can function as links; they can be used to launch script actions; and they can be used as image maps. Although scripting is not discussed in this course, you will be introduced to clickable image maps in a later lesson. In this lesson, you will focus on using images purely as graphical enhancements.

Image file formats: The three universally supported Web image formats are:

- Graphics Interchange Format (GIF).
- Joint Photographic Experts Group (JPEG).
- Portable Network Graphics (PNG).

All image file formats are discussed in earlier chapter (IN Unit I).

Format	Transparency	Interlacing	Compression	Animation
GIF	Yes	Yes	Yes	
JPEG (standard)	No	No	Yes	No
PNG	Yes	Yes	Yes	Yes

There is no implied line or paragraph break before or after the <img> tag, so images can be truly "in line" with text and other content.

### 10.1 The src attribute

The src attribute for the <img> tag is compulsory. Its value is the image file's URL, either absolute or relative to the document referencing the image.

```
<imgsrc="image_folder/imagefile.gif">
```

This is main attribute that is required for <img> tag. This tag is src (abbreviation for source) is must. src attribute is use to specify the name and, the location of your image file.

### 10.2 The alt attribute

It is a good coding practice by containing the alt attribute with a corresponding value. The alt attribute specifies alternative text to appear while the graphic is loading. This alternative text will also display if the image fails to load or if the user has configured his or her browser not to display images.

The syntax for using the alt attribute is as follows:

```
<imgsrc="image.gif" alt="This is alternative text. Type anything here.">
```

### 10.3 Align attribute:

Align attribute is use to aligning images relative to text. For attractive web page it is must know how to position images relative to text on a page. The syntax for the align attribute is as follows:

```
<imgsrc="imagefile.gif" align="alignment value"/>
```

```
align="Right"  
align="left"  
align="center"  
align="justify"
```

Default value of align attribute is left.

Values for align attribute

Align Attribute Value	Description
"bottom"	The default alignment. The bottom of the image is aligned with the baseline of adjoining text.
"middle"	A vertical — not horizontal — alignment option. This value aligns the middle of the image to the baseline of adjoining text.
"top"	Aligns the top of the image with the top of adjoining text.
"left"	Floating the image to the left of the text paragraph into which the <img> tag is inserted. The top of the image will align with the left and top of the adjoining text.
"right"	Floating the image to the right of the text paragraph into which the <img> tag is inserted. The top of the image will align with the right and top of the adjoining text.

Additional <img> attributes include:

- hspace - aligns the image horizontally, moving it to the right.
- vspace - aligns the image vertically, moving it down from the top
- height - resizes the image's original height dimension
- width - resizes the image's original width dimension

### 10.4 Height, width and border attribute

The syntax for specifying image height and width is as follows:

```
<imgsrc="imagename.gif" height="HeightInPixels" width="WidthInPixels"/>
```

**Border attribute:** Use the border attribute and a pixel-width thickness value to remove (border=0) or widen that image border. Beware that this attribute, too, is deprecated in HTML, but continues to be well supported by the modern browsers.

## 10.5 dynsrc loopattribute

Use the `dynsrc` attribute extension in the `<img>` tag to reference an AVI movie for inline display by Internet Explorer. Its required value is the URL of the movie file, enclosed in quotation marks. For example, this text displays the tag and attribute for an AVI movie file entitled intro.avi:

```

```

The `loop` attribute:

Internet Explorer normally plays a movie clip from beginning to end once after download. The `loop` attribute for the movie `<img>` tag lets to the clip play repeatedly for an integer number of times set by the attribute's value, or forever if the value is infinite.

The following intro.avi movie clip will play from beginning to end, then restart at the beginning and play through to the end four more times:

```

```

## 11. MARQUEE TAG :

`Marquee` tag is used to move your image or text on browser window. This makes the web page dynamic and make the appearance of page very attractive, without using any complex javascript coding. `<marquee>` tag have many attributes which makes and control the movement of text or image good looking.

### 11.1 ATTRIBUTE OF <MARQUEE>:

**Behavior:** This attribute define the way of moving. The following value can be assign to this attribute:

- behavior="scroll"
- behavior="alternate"

**Direction:** This attribute define the direction of marquee content. The following value can be assign to this attribute:

- Left : Means make movement of marquee content towards left.
- Right: Means make movement of marquee content towards right.
- Up: Means make movement of marquee content towards upward direction.
- Down: Means make movement of marquee content towards downward direction.

#### Scroll amount attribute:

Scrollamount attribute is use to define the speed of moving content of marquee tag. Value of its attribute is any integer number. As the number value is increases the speed of moving content of marquee is directly increased.

```
<marquee behavior="scroll" direction="left" scrollamount="4">Moving text</marquee>
```

In above examples later one is move with higher speed.

Example:

```
<html>
```

## INTRODUCTION TO HTML

```
<head><title>Marquee</title>
```

```
</head><body>
```

```
<marquee behavior="scroll" direction="left"
```

```
onMouseOver="this.stop()" onMouseOut="this.start()"
```

```
scrollamount="4"
```

```
onMouseDown="this.setAttribute('scrollamount',20)"><b>Moving text</b>
```

```
</marquee>
```

```
<marquee behavior="alternate" direction="right"
```

```
onMouseOver="this.stop()" onMouseOut="this.start()"
```

```
scrollamount="8"
```

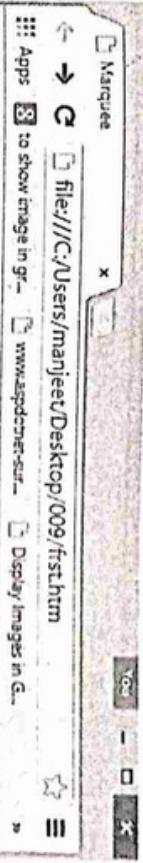
```

```

```
</marquee>
```

```
</body>
```

```
</html>
```



## 12. HTML Table Tags

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, lists, paragraphs, forms, horizontal rules, tables, etc. And the letter tr stands for table row.

#### Table Tags

Tag	Description
<table>	Defines a table
<th>	Defines a table header
<tr>	Defines a table row
<td>	Defines a table cell
<caption>	Defines a table caption
<colgroup>	Defines groups of table columns
<col>	Defines the attribute values for one or more columns in a table
<thead>	Define header portion of table
<tbody>	Define body part of table
<tfoot>	Define footer part of table.

A wide variety of tables can be created with only five tags: the <table> tag, which encapsulates a table and its elements in the document's body; content; the <tr> tag, which defines a table row; the <th> and <td> tags, which define the table's headers and data cells; and the <caption> tag, which defines a title or caption for the table.

By ord these core tags, you may also define and control whole sections of tables, including adding running headers and footers, with the <colgroup>, <col>, <thead>, and <tfoot> tags. Each tag has one or more required and optional attributes, some of which affect not only the tag itself but also related tags.

#### 12.1 The <table> Tag

Tables work like HTML lists tags, in that you must use the table container tag to hold together a group of tags that define each individual row. The main container is the <TABLE> tag, which uses enclosing tags for table rows (<TR>) and table data (<TD>). Most tables will also use an element for table headers (<TH>), which is generally used as the title text for rows and columns. <TH> enclosed text is by default bold.

Tables take the following format:

```
<TABLE>
<CAPTION>Caption text for table</CAPTION>
<TR><TH>column1</TH><TH>column2</TH><TH>column3</TH>
<TR><TD>row1data1</TD><TD>row1data2</TD><TD>row1data3</TD>
<TR><TD>row2data1</TD><TD>row2data2</TD><TD>row2data3</TD>
</TABLE>
```

The <TABLE> Tag attribute

**12.1.1 Align and col attribute**  
**ALIGN :** The ALIGN attribute is used to determine where the table will appear relative to the browser window. Valid values are:  
**ALIGN=LEFT**

**COLS.** The COLS attribute specifies the number of columns in your table, allowing the browser to draw the table as it downloads.

#### 12.1.2 Cellpadding and Cellspacing

**CELLSPACING.** The CELLSPACING attribute tells the browser how much space to include between the walls of the table and between individual cells. Value is a number in pixels. Cellspacing is the pixel width between the individual data cells in the table (The thickness of the lines making the table grid). The default is zero. If the border is set at 0, the cellspacing lines will be invisible.

```
<table border="1" cellspacing="5">
```

**CELLPADDING.** The CELLPADDING attribute tells the browser how much space to give data between the cell contents and the cell border. The default for this property is also zero. This feature is not used often, but sometimes comes in handy when you have your borders turned on and you want the contents to be away from the border a bit for easy viewing. Cellpadding is invisible, even with the border property turned on. Cellpadding can be handled in a style sheet.

```
<table border="1" cellpadding="5">
```

#### 12.1.3 Border and width attribute Attribute

**Border:** To display a table with borders, you will use the border attribute.

```
<table border="1"><tr><td>Row 1, cell 1</td><td>Row 1, cell 2</td></tr></table>
```

#### Table Width

The width attribute can be used to define the width of your table. It can be defined as a fixed width or a relative width. A fixed table width is one where the width of the table is specified in pixels. For example, this code, <table width="550">, will produce a table that is 550 pixels wide. A relative table width is specified as a percentage of the width of the visitor's viewing window. Hence this code, <table width="80%">, will produce a table that occupies 80 percent of the screen.

The WIDTH attribute sets the relative or absolute width of your table in the browser window. Values can be percentages, as in WIDTH="50%", or absolute values. With absolute values, a suffix is used as px for pixel or in for inches.

The browsers automatically wrap text lines to fill the allotted table cell space. The nowrap attribute, when included in the <table> tag, stops that normal word wrapping in all rows in the table. With nowrap, the browser assembles the contents of the cell onto a single line, unless you insert a <br> or <p> tag, which then forces a break so that the contents continue on a new line inside the table cell.

## 12.1.4 The colspan and row span attribute

### The colspan attribute

It's common to have a table header that describes several columns beneath it.

Use the colspan attribute in a table header or data tag (td or th) to extend a table cell across two or more columns in its row. Set the value of the colspan attribute to an integer value equal to the number of columns you want the header or data cell to span.

For example:<td colspan="3">

### The rowspan attribute

Just as the colspan attribute layers a table cell across several columns, the rowspan attribute stretches a cell down two or more rows in the table.

Include the rowspan attribute in the <th> or <td> tag of the uppermost row of the table where you want the cell to begin and set its value equal to the number of rows you want it to span. The cell then occupies the same space as the current row and an appropriate number of cells below that row. The browser flows the contents of the cell to occupy the entire extended space.

For example:<td rowspan="3">

Like the colspan attribute, the browser ignores over extended rowspan attributes and extends the current cell only down rows. The browsers do not add empty rows to a table to fill a rowspan below the last defined row in a table.

## 12.1.5 The bgcolor, valign and background attributes

The valign attribute sets the default vertical alignment of data in cells for the entire table. Acceptable values for the valign attribute in <table> are top, bottom, middle, or baseline; the default vertical position is the center of the cell.

You can change the background color - this time for an individual data cell. This attribute's values either an RGB hexdecimal color value or a standard color name as explained earlier.

## 12.2 The <caption> Tag

A table commonly needs a caption to explain its contents, so the popular browsers provide a table caption tag. The <caption> tag and its contents put immediately after the <table> tag, but it can be placed nearly anywhere inside the table and between the row tags. The caption may contain any body content much like a cell within a table.

The <caption> tag is a container

The <caption> tag has one attribute, ALIGN. ALIGN="TOP" and ALIGN="BOTTOM" are encouraged. By default, text is also aligned to center (horizontally). By TOP and BOTTOM.

For Example : <CAPTION ALIGN="BOTTOM">Table Food</CAPTION>

## 12.3 The <thead> Tag

Use the <thead> tag to define a set of table header rows. The <thead> tag may appear only once per table and is placed at the beginning, just after the <table> tag. Within the <thead> tag, you may place one or more <tr> tags defining the rows within the table header.

## INTRODUCTION TO HTML

### 12.4 The <tfoot> Tag

Use the <tfoot> tag to define a footer for a table. The <tfoot> tag may appear only once, just before the end of a table. Like <thead>, it may contain one or more <tr> tags that let you define those rows. The rows define in any part of table but always appear in footer section as these rows (tr) are define within <tfoot> tag.

### 12.5 The <tbody> Tag

Use the <tbody> tag to divide your table into discrete sections. The <tbody> tag collects one or more rows into a group within a table. It is perfectly acceptable to have no <tbody> tags within a table, although where you might include one, you probably will have two or more <tbody> tags within a table. So identified, you can give each <tbody> group different rule line sizes above and below the section. Within a <tbody> tag, only table rows may be defined using the <tr> tag. And, by definition, a <tbody> section of a table stands alone. For example, you may not span from one <tbody> into another.

### 12.6 The <colgroup> Tag

The <colgroup> tag defines a column group. You can use the <colgroup> tag in two ways: as a single definition of several identical columns, or as a container for several dissimilar columns. You can put one or more <colgroup> tags within a <table> tag. The ending </colgroup> tag is rarely used in HTML.

**Attribute :** Use the span attribute with the <colgroup> tag to achieve the first type of column grouping. The value of the span attribute is the integer number of columns affected by the <colgroup> tag.

For example, a table with six columns - four in the first group and two in the other - would appear in the source code as:

```
<colgroup span="4">
<colgroup span="2">
```

### 12.7 Table Rows <tr>

Table rows (<tr>) can accept one attribute you should concern yourself with - ALIGN. The ALIGN attribute is used to determine how text will appear (horizontally) in each of the rows data cells. For example:

```
<TR ALIGN="CENTER"><TH>Date</TH><TH>Chocolate</TH><TH>Vanilla</TH>
<TR ALIGN="LEFT"><TH>1970</TH><TD>50%</TD><TD>50%</TD>
<TR ALIGN="RIGHT"><TH>1980</TH><TD>76%</TD><TD>24%</TD>
```

### 12.8 Table Data <td> and Table Header <th>

The difference between the two is that <TH> emphasizes (boldfaces) the text and <TD> does not. Now, technically, the <TH> is a tag that the browser interprets as a header and thus displays text in a way that's distinct from the <TD> tag. In practice, that generally means it's turned bold.

Aside from accepting nearly any type of HTML markup tags within them, both tags can accept four attributes. These are ALIGN, VALIGN, COLSPAN, and ROWSPAN. If you were to add all of these attributes, a typical <TH> (or <TD>) tag would be formatted like the following:

72

- ALIGN="direction" VALIGN="direction" COLSPAN="number" ROWSPAN="italics"  
COLSPAN is used to align the data within the cell horizontally, accepting values of LEFT, RIGHT, and CENTER. Note that ALIGN is redundant when used with the ALIGN attribute of TD unless it is used to override the <TR ALIGN=> setting.
- VALIGN is used to align the data vertically within cells. Possible values are TOP, BOTTOM, and CENTER.
- COLSPAN and ROWSPAN are used to force a cell to span more than one column or row respectively. An example of this might be:

**Example of table using all tags:**

```
<html>
<head></head>
<body>
<table border="1" background="img01.jpg" width="200px">
<caption align="right">List of students</caption>
<col span="2" style="background-color:pink;">
<thead><tr>
<th>Roll</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr> <td align="center" colspan="2" style="background-color:yellow;">Max95.5</td>
</tr>
<tr>
<td align="center" colspan="2" style="background-color:lightgreen;">102</td>
<td align="center" colspan="2" style="background-color:lightblue;">Ajay</td>
<td align="center" colspan="2" style="background-color:lightgreen;">75.5</td>
</tr>
<tr>
<td align="center" colspan="2" style="background-color:lightblue;">Rai</td>
<td align="center" colspan="2" style="background-color:lightgreen;">102</td>
<td align="center" colspan="2" style="background-color:lightblue;">Rai</td>
</tr>
</tbody>
</table>
```

```
<td>85.5</td>
</tr>
<tr>
<td>103</td>
<td>Suresh</td>
<td>65.5</td>
</tr>
<tr>
<td>104</td>
<td>Taru</td>
<td>95.5</td>
</tr>
<tr>
<td>105</td>
<td>Manu</td>
<td>85.5</td>
</tr>
```

← → C ⌂ Apps ⌂ to show image in g... ⌂ www.appspotketku... ⌂ Backup Image in G...

List of students

Roll	Name	Per
101	Ajay	75.5
102	Ravi	85.5
103	Suresh	65.5
104	Tanu	75.5
105	Manu	85.5
<b>Max</b>		<b>95.5</b>

**Exercise****Unit-II****Chapter****5****HTML Frames Forms and Web Development****1. HTML <FRAMESET> and <FRAME> Tag**

For creating frame <FRAMESET> tag is used. This container is required for frames-style pages but it also replaces the <BODY> tag completely on these pages. On a typical page, <FRAMESET> is added like this:

```
<HTML>
```

```
<HEAD>
```

```
...HEAD markup...
```

```
<HEAD>
```

```
<FRAMESET>
```

```
...Frames and other HTML markup...
```

```
<FRAMESET>
```

```
<HTML>
```

The <FRAMESET> tag can accept two attributes: ROWS and COLS. Both attributes accept numerical values (size in pixels), percentages, or a combination of both. The value (\*) can also be used to suggest that a particular row or column should take up the rest of the page.

The number of values suggests the number of rows or columns you give the attribute. These attributes take the following format:

```
<FRAMESET ROWS="numbers,percentages,*" COLS="numbers,percentages,*">
```

An example like the following would create two rows; one 50 pixels long and another row is the rest of the page:

```
<FRAMESET ROWS="50,*">
```

The following example to create page with two column, one on the leftmost 25 percent of the screen and one on the other 75 percent:

```
<FRAMESET COLS="25%,75%">
```

OR

```
<FRAMESET COLS="25%,* ">
```

In order to create rows within columns and vice-versa you can nest <FRAMESET> statements. The following will create a page with two columns:

- Very Short Questions**
- Q1. What does the HTML stands for?
  - Q2. What is HTML tags?
  - Q3. Name any interpreter of HTML language.
  - Q4. Explain head any body tags.
  - Q5. Explain structure of HTML document.
  - Q6. Write extension of HTML document.
  - Q7. What are paired and singular tags?
  - Q8. What is attribute?
  - Q9. Write about HTML comments.
  - Q10. Tag <a> is stands for?
  - Q11. What do you mean by Cellpadding and Cellsspacing?

**Short Questions**

- Q1. Explain features of HTML language.
- Q2. Write note on named character references.
- Q3. What do you mean by list tags? Explain ordered and unordered list with example.
- Q4. Write note on definition list.
- Q5. Explain alink, link and vlink attributes of <body> tag.
- Q6. Why marquee tag is used in HTML document? Explain with code.
- Q7. What is use of <body>, <thead>, <tfoot> tag in HTML?

**Long Questions**

- Q1. Describe Formatting tags in details with proper example.
- Q2. Explain hyperlink in HTML with all attributes.
- Q3. Which tag is used for display image in your web document? Explain all attributes <img> tag with code.
- Q4. Explain table tag in HTML. Explain all supporting table tags with attributes and code.
- Q5. Write down the HTML code using basic HTML tag.

```
<FRAMESET COLS="25%,75%">
<FRAMESET ROWS="50%,50%">
```

```
</FRAMESET>
```

```
<FRAMESET ROWS="10%,90%">
```

```
</FRAMESET>
```

```
</FRAMESET>
```

The **<FRAME>** tag is used within the **<FRAMESET>** container to determine what will actually appear in a particular frame. Each **<FRAME>** tag is an empty tag and it's not unlike the **<LI>** tags. It is simply there, to determine what URL or name is associated with the particular frame it defines. It takes the following format:

```
<FRAMESET COLS="numbers">
```

```
<FRAME SRC="URL">
```

```
...
```

```
<FRAMESET>
```

## 1.1 <FRAME> Attribute

### 1.1.1 SRC attribute:

**SRC** attribute: is used to tell the frame what URL should be loaded in that frame. The value of the **src** attribute for the **<frame>** tag is the URL of the document that is to be displayed in the frame. The document referenced by the **src** attribute may be any valid document or any displayable object, including images and multimedia. For instance, the following would create two frame rows—one that loaded the URL index.html at the top of the Web page and one that loaded the URL help.html at the bottom of the page

```
<FRAMESET ROWS="50%,50%">
<FRAME SRC="index.html">
<FRAME SRC="help.html">
</FRAMESET>
```

NAME are appearance oriented. Let's deal with them first and come back to NAME in a moment.

MARGINWIDTH and MARGINHEIGHT are used to control the right/left margins and the top/bottom margins of the text and graphics within a frame, respectively. Each takes a numerical value in pixels. For example:

```
<FRAME SRC="text.html" MARGINWIDTH="5" MARGINHEIGHT="5">
</FRAMESET>
```

This creates a five-pixel border between the contents of text.html and the frame edges. SCROLLING can accept the values yes, no, and auto and is used to determine whether or not scroll bars will appear in the frame window. The default value is auto, and this is probably the best to use in most cases. Since users have all different screen resolutions and available browser window space, even short documents will sometimes need to be scrolled.

The NORESIZE attribute doesn't require a value assignment, and is used to keep the user from resizing a frame window. (Frame windows can be resized by dragging the frame with the mouse

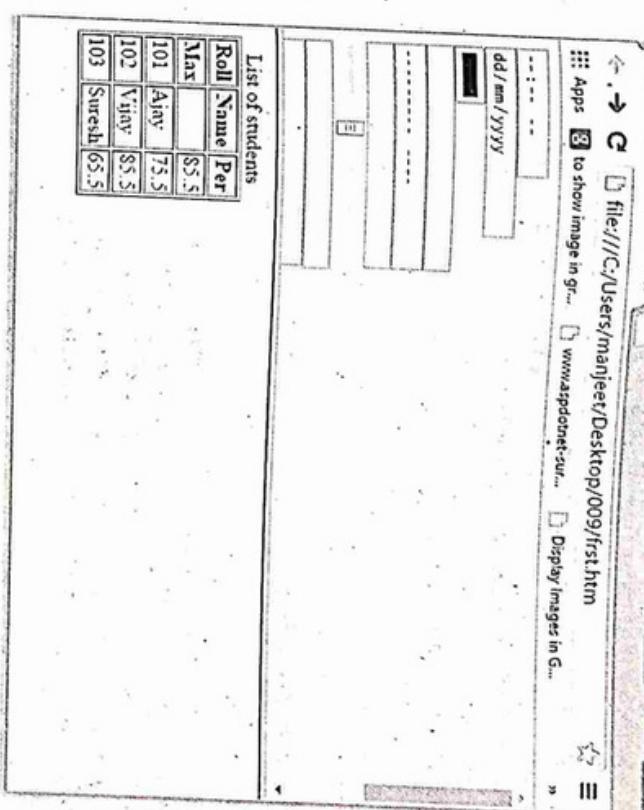
in the viewer window.)

An example of SCROLLING and NORESIZE would be:

```
<FRAME SRC="text.html" SCROLLING="yes" NORESIZE>
```

```
</FRAMESET>
```

parent exists, it acts like **\_self**.



### TARGET=\_top

The document is loaded in the top most frame of the current browser window.

### 1.1.3 The noresize attribute

Even though you may explicitly set frame dimensions with attributes in the `<frameset>` tag, user can manually alter the size of a column or row of frames. To suppress this behavior, add the `noresize` attribute to the frame tag in the row or column whose dimensions not want alter by user. It effectively freezes the relative proportions of all the frames.

The `noresize` attribute is especially useful for frames that contain fixed images serving as advertisements, abutton bar, or a logo. By fixing the size of the frame to contain image and setting the `noresize` attribute guarantee that the image is displayed in the intended manner and that the remainder of the browser window is always given over to the other frames in the document.

Example:

```
<frame src="frame_1.html" noresize="noresize">
```

### 1.1.4 The scrolling attribute

The browser displays vertical and horizontal scrollbars with frames whose contents are larger than the allotted window space. If there is sufficient space for the content, the scrollbars disappear.

Example:

```
<frame src="frame_1.html" noresize="noresize">
```

### 1.1.5 The marginheight, marginwidth and frameborder attributes

The browser normally places a small amount of space between the edge of a frame and its contents. These can change with the `marginheight` and `marginwidth` attributes, each including a value for the exact number in pixels to place around the frame's contents.

You cannot make a margin less than 1 pixel.

Example:

```
<frame src="frame_a.htm" marginwidth="50">
```

```
<frame src="frame_b.htm" marginwidth="30">
```

The `frame border` attribute is used to apply the border to the frame. Values this attribute is yes or 1 and no or 0 respectively enable or disable borders for the frame.

Example:

```
<frameset rows="20%, 80%" framespacing="20" border="20">
```

## 2. HTML FORM

HTML Forms are required when you want to collect some data from the site visitor. For example etc.

A form will take input from the site visitor and then will post it to a back-end application which will perform required processing on the

passed data based on defined business logic inside the application.

There are various form elements available like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.

The HTML `<form>` tag is used to create an HTML form and it has following syntax:

```
<form action="File_URL" method="GET/POST">
    Put form elements here.
</form>
```

### Attribute

### Description

Attribute	Description
action	Value of this attribute is URL of any file where the data of form to be submitted on click on submit button.
method	It defines the way or method to be used to upload data. The methods used are GET and POST methods.

## 2.1 Form Attributes

Apart from common attributes, following is a list of the most frequently used form attributes:

### Attribute Description

**action :** Value of this attribute is URL of any file where the data of form to be submit on click on submit button.

**method :** It defines the way or method to be used to upload data. The methods used are GET and POST methods.

**target:** Specify the target window or frame where the result of the script will be displayed. It takes values like \_blank, \_self, \_parent etc.

### 2.1 Method Attribute

The method attribute tells the server how to submit the form information in a file whose URL is mentioned in action attribute on submit button click. There are two methods, get and post. The default method is get.

**80****`method="get"`**

This sends the form information by including it on the URL. The get method sends the information to the server in one step. It is best used for forms where the amount of data submitted is small.

`<form method="get" action="">`

When to use the get method:

- Short forms, with only 1 or 2 input fields
- Forms with primarily select, radio, and checkbox fields
- When the form needs to be called from a link.
- When a form or page need to be bookmarked.

**`<form method="post">`**

This sends the form information in the HTTP environment. It is a little more secure, as your customers can't see the submitted data directly. The post method sends the data to the server in two steps. First the browser contacts the server and after the contact is made, it sends the information. It is best used when you have a lot of information to pass (either textarea tags, or just a lot of fields).

`<form method="post" action="">`

When to use the post method:

- Longer forms, more than 3 input fields
- Forms with large textarea fields
- Forms where security is more important.
- If we want make book mark to the page in that case post method is not better as post method does not allow to make book mark.

**Difference between GET and POST**

Features	GET	POST
History	Parameters remain in browser history because they are part of the URL	Parameters not remain in browser history because they are part of the URL
Bookmarked	Can be bookmarked	Cannot bookmarked
BACK button/re-submit behaviour	GET requests are re-executed but may not be re-submitted to server if the HTML is stored in the browser cache.	POST requests are not re-executed but may not be re-submitted to server if the HTML is stored in the browser cache.

Hacked	Easier to hack.	Chance of hacking is negligible.
Restrictions on form data type	Yes, only ASCII characters are allowed.	No, kind of characters are allowed.
Security	GET is less secure compared to POST because data sent is part of the URL. So it's saved in browser history and server logs in plaintext.	Highly secure as data is not visible in URL and not save in browser history.
Restrictions on form data length	Yes, since form data is in the URL and URL length is restricted. Only 2048 characters is allow to send safely.	No restrictions on length of data.
Usability	GET method should not be used when sending passwords or other sensitive information.	POST method is better to used when sending passwords or other sensitive information.
Visibility	GET method is visible to everyone as it will be displayed in the browser's address bar.	POST method is not visible to everyone as data is directly submit to body.
Cached	Can be cached.	Cannot cached.

Only ASCII data can be submitted by GET. Special care should be taken to encode and decode other types of characters. On the other hand, binary data, images and other files can all be submitted through `METHOD="POST"`.

All form data filled in is visible in the URL and it is also stored in the user's web browsing history of the browser by `GET` method which makes less secure. However, one advantage of form data being sent as part of the URL is that one can bookmark the URLs and directly use them and completely bypass the form-filling process.

**HTML Form Controls**

There are different types of form controls that you can use to collect data using HTML form:

- Text Input Controls
- Checkboxes Controls
- Radio Box Controls
- Select Box Controls
- File Select boxes

- Hidden Controls
- Clickable Buttons
- Submit and Reset Button

## 2.2 TEXT INPUT CONTROLS

There are three types of text input used on forms:

- **Single-line text input controls** - This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML `<input>` tag.
- **Password input controls** - This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML `<input>` tag.
- **Multi-line text input controls** - This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML `<textarea>` tag.

### 2.2.1 Single-line text input controls

This control is used for items that require only one line of user input. They are created using HTML `<input>` tag using text value of type attribute.

#### Example

Here is a basic example of a single-line text input used to take first name and last name:

```
<html>
<head>
<title>Text Input Control</title>
</head>
<body>
<form>
First name: <input type="text" name="first_name" />
<br>
Last name: <input type="text" name="last_name" />
</form>
</body>
</html>
```

This will produce following result:

Firstname: \_\_\_\_\_  
Last name: \_\_\_\_\_

**Attributes of `<input>` to create textfields**

Following is the list of attributes for `<input>` tag for creating text field.

Attribute	Description
type	Indicates the type of input control and for text input control it will be set to text.
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
value	This can be used to provide an initial value inside the control.
size	Allows to specify the width of the text-input control in terms of characters.
maxlength	Allows to specify the maximum number of characters a user can enter into the text box.

### 2.2.2 Password input controls

This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML `<input>` tag but type attribute is set to **password**.

#### Example

```
<html>
<head>
<title>Password Input Control</title>
</head>
<body>
<form>
User ID : <input type="text" name="user_id" />
<br>
Password: <input type="password" name="password" />
</form>
</body>
</html>
```

This will produce following result:

```
User ID : _____  
Password: _____
```

#### Attributes

Following is the list of attributes for <input> tag for creating password field.

Attribute	Description
type	Indicates the type of input control and for password input control it will be set to password.

name Used to give a name to the control which is sent to the server to be recognized and get the value.

value This can be used to provide an initial value inside the control.

size Allows to specify the width of the text-input control in terms of characters.

maxlength Allows to specify the maximum number of characters a user can enter into the text box.

#### 2.2.3 Multiple-Line Text Input Controls

This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.

Example:

```
<html>  
<head>  
<title>Multiple-Line Input Control</title>  
</head>  
<body>  
<form>
```

Description :<br /><textarea rows="5" cols="50" name="description">  
Enter description here...</textarea>

```
</form>  
</body>  
</html>
```

Attributes:  
Following is the list of attributes for <textarea> tag.

Attribute	Description
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
rows	Indicates the number of rows of text area box.
cols	Indicates the number of columns of text area box

#### 2.3 Checkbox Control

Checkboxes are used when more than one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to checkbox.

Example:

```

<html>
<head>
<title>Checkbox Control</title>
</head>
<body>
<form>
<input type="checkbox" name="maths" value="on">Maths
<input type="checkbox" name="physics" value="on"> Physics
</form>
</body>
</html>

```

This will produce following result:

Maths     Physics

Attributes

Attribute	Description
type	Indicates the type of input control and for checkbox input control it will be set to checkbox.
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
value	The value that will be used if the checkbox is selected.
checked	Set to checked if you want to select it by default.

## 2.4 Radio Button Control

Radio buttons are used when out of many options, just one option is required to be selected. They are also created using HTML &lt;input&gt; tag but type attribute is set to radio.

Example:

```

<html>
<head>
<title>Radio Box Control</title>
</head>
<body>
<form>
<input type="radio" name="subject" value="maths">Maths
<input type="radio" name="subject" value="physics"> Physics
</form>
</body>
</html>

```

This will produce following result:

Maths     Physics

Attributes

Attribute	Description
type	Indicates the type of input control and for checkbox input control it will be set to radio.
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
value	The value that will be used if the radio box is selected.
checked	Set to checked if you want to select it by default.

## 2.5 Dropdown List Control

A select box, also called drop down box which provides option to list down various options in the form of drop down list, from where a user can select one or more options. For dropdown list <select> and <option> tags are used <option> tag is child tag of <select> tag. Both <select> and <option> tags are paired tag.

88

**Example:**

```
<html>
<head>
<title>Select Box Control</title>
</head>
<body>
<form>
<select name="dropdown">
<option value="Maths" selected>Maths</option>
<option value="Physics">Physics</option>
</select>
</form>
</body>
</html>
```

This will produce following result:

Maths ▾

**Attributes :** Following is the list of important attributes of <select> tag:

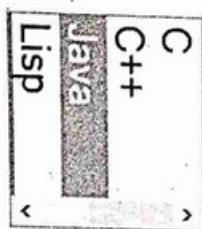
Attribute	Description
value	The value that will be used if an option in the select box box is selected.
selected	Specifies that this option should be the initially selected value when the page loads.
label	An alternative way of labeling options

**2.6. List box Control:**

The second example shows a SELECT element rendered as a list box. If more than one entry is active when the form is submitted, then more than one value is sent.listed as separate entries.

**Example:**

```
<SELECT NAME="language" MULTIPLE>
<OPTION VALUE="c">C
<OPTION VALUE="c++">C++
<OPTION VALUE="java" SELECTED>Java
<OPTION VALUE="lisp">Lisp
<OPTION VALUE="perl" SELECTED>Perl
<OPTION VALUE="smalltalk">Smalltalk
<SELECT>
```



Following is the list of important attributes of <option> tag:

If you want to allow a user to upload a file to your web site, you will need to use a file upload box, also known as a file select box. This is also created using the <input> element but type attribute is set to file.

Example  
Here is example HTML code for a form with one file upload box:

```
<html>
<head>
<title>File Upload Box</title>
</head>
<body>
<form>
<input type="file" name="fileupload" accept="image/*" />
</form>
</body>
</html>
```

**Browse...** No file selected.

Attributes :

Attribute	Description
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
accept	Specifies the types of files that the server accepts.

## 2.8 Button Controls

There are various ways in HTML to create clickable buttons. You can also create a clickable button using `<input>` tag by setting its type attribute to `button`. The type attribute can take the following values:

Example:

```
<html>
<head>
<title>File Upload Box</title>
</head>
<body>
<form>
<input type="submit" name="submit" value="Submit" />
<input type="reset" name="reset" value="Reset" />
<input type="button" name="ok" value="OK" />
<input type="image" name="imagebutton" src="/html/images/logo.png" />
</form>
</body>
</html>
```

This will produce following result:



Submit   Reset   OK

Image Button

### 2.8.1 Reset and Submit Buttons

The **Reset** button allows the user to clear the data they have entered in the form and start fresh. The **VALUE** is the words that you want to appear in the box. The reset button allow to clear the form elements without sending data to server.

The submit type button allow to send the form elements data to the file whose URL is mentioned in action attribute of `<form>` tag. Remember that there is only one working submit button per form.

Standard reset and submit buttons are as follows.

```
<INPUT type="reset" value="clear fields">
<INPUT type="submit" value="submit">
```

### Submit Image

Also use an image instead of a button to send the form. Note, you can only make a Submit button, not a Reset image button.

```
<INPUT type="image" src="submit.gif">
```

### 2.9 Hidden Form Controls

Hidden form controls are used to hide data inside the page which later on can be pushed to the server. This control hides inside the code and does not appear on the actual page. For example, following hidden form is being used to keep current page number. When a user will click next page then the value of hidden control will be sent to the web server and there it will decide which page has be displayed next based on the passed current page.

Example

```
<html>
<head>
<title>File Upload Box</title>
</head>
<body>
<form>
<input type="hidden" name="pagename" value="10"/>
<input type="submit" name="submit" value="Submit"/>
<input type="reset" name="reset" value="Reset"/>
</form>
</body>
</html>
```

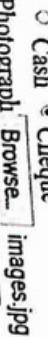
```
<form action="second.html" method="post">
<fieldset>
<legend align="center">Enter Student details</legend>
<b>Roll</b>
<input type="text" name="roll" value="" placeholder="Enter Roll"/><br>
<!--<Label for="name1">Name</Label>-->
Name
<input type="text" name="name1" value="" /><br>
<!--<Label for="per">Per</Label>-->
<b>Percentage</b>
<input type="text" name="per" value="" placeholder="Enter Name"/><br>
<label for="course">Course</label>
<select>
<optgroup label="Programming courses">
<option>C</option>
<option>C++</option>
<option>Java</option>
</optgroup>
<optgroup label="Basic courses">
<option>Office</option>
<option>Windows</option>
<option>Tally</option>
</optgroup>
</select>
<br>
<input type="checkbox" name="c1" value="Fast Track" /> Fast Track course
<input type="checkbox" name="c2" value="Normal Course" /> Normal Course..<br />
<input type="radio" name="payment" value="Cash" /> . Cash
<input type="radio" name="payment" value="Cheque" /> Cheque<br />
Photograph<input type="file" /><br />
Password<input type="password" name="pwd" /><br />
<input type="hidden" name="var" value="sometext" />
Remarks<textarea rows="5" cols="40"></textarea><br />
<input type="submit" value="Login data" />
<input type="reset" value="Cancel" />
</fieldset>
</form>
</body>
</html>
```

## HTML FRAMES FORMS AND WEB DEVELOPMENT

File:///C:/Users/mayank/Desktop/2023/Part 1 Q. Screen Getting Started to show image in grid. Display Images in Grid...

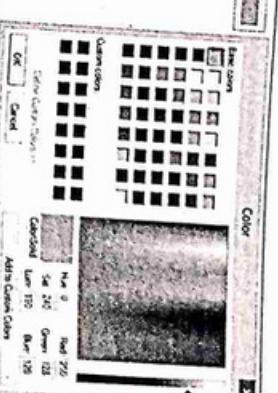
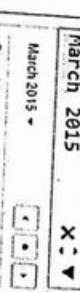
Most Visited  Getting Started  to show image in grid.  Display Images in Grid...

Enter Student details

Roll	Enter Roll
Name	Ravi
Percentage	97
Course	Java
<input checked="" type="checkbox"/> Fast Track course <input type="checkbox"/> Normal Course..	
<input type="radio"/> Cash <input checked="" type="radio"/> Cheque	
Photograph	<input type="button" value="Browse..."/>  images.jpg
Password	*****
Welcome to Form Elements. Welcome to Form Elements. Welcome to Form Elements. Welcome to Form Elements.	
Remarks	Welcome to Form Elements.
Login data	<input type="button" value="Cancel"/>

### 2.10 Different values of 'type' attribute:

There are many value can assign to 'type' attribute of <input> tag supported by new browsers. Old browsers like Internet Explorer is not supported following value of 'type' attribute.

Type="color"	<input type="color"/> 
Type="email"	<input type="text" value="savi.com"/> <small>Please include an '@' in the email address. 'savi.com' is missing an '@'.</small>
Type="month"	<input type="month" value="March 2015"/> 
Type="number"	<input type="number" value="5"/>
Type="range"	<input type="range"/>
Type="url"	<input type="url" value="imagefolder"/> <small>Please enter a URL.</small>
Type="week"	<input type="week"/> 

Try it yourself to gain more.

Value of type	Result
type="time"	12:59 AMX:
type="date"	20/03/2015 X:▼
Month 2015 *	<input type="button" value="&lt;"/> <input type="button" value="•"/> <input type="button" value="&gt;"/>
Sun Mon Tue Wed Thu Fri Sat	1 2 3 4 5 6 7
	8 9 10 11 12 13 14
	15 16 17 18 19 20 21
	22 23 24 25 26 27 28
	29 30 31

Web design itself refers to the process of creating a web page's appearance and to the choice of a right color scheme, page layout, fonts and more. Every single web page in a website has different content, but all the pages are using a similar graphic design. Often sites will use website templates which contain all the basic elements of web design - the website's CSS style, buttons, backgrounds, borders and various graphic elements like hover images, bullets and header banners. When the website template is applied to the website, all the pages assume its appearance, using the same styles, background and other graphical elements.

Before you start designing your web page, you need to know the contents of the web page, because you need to arrange them within the page's layout. There are different types of content, text, images, videos, animations and other dynamic elements. In order to create a successful web page you need to put the right content in the right place.

### Explain the project:

To finish a project in a perfect and good way you have to start it effectively and with efforts. For this well preparation is required. The right preparation will save your long effort as well as time. This results into the end of your project in proper way and will help you to end it in a way you wanted. Thus a proper planning is needed.

Some major things are given below:

- Purpose of site.
  - Type of audience or visitors will visit the website.
  - What the visitors or users wants in particular field?
  - What are the essential commodities?
  - Content of web site are matters the effectiveness of web site.
- So all above is need are requires a lot of mind work. And prepared to answer the web site designer.

### Designer searching:

We that not all designer is good in all fields. Once defining your project look for a designer who is prefect for your tastes better than any other. Well known web designers have proper portfolios on net. Search them, look for an experienced website designer who has already created a kind of website you are looking for. Also, search a designer who fits your budget. Even a simple website costs around a few hundred dollars. Many factors contribute to the cost of a web project. Following things are decides the cost of web site.

- The bulk (size) or no. of pages of website.
- Content of website.
- The level of coding required
- Price of the designer
- The flash information.

### Building website according to the user's need:

Make our website attractive is first demand of website owner as attractiveness of website directly on business. Every website owner wants his website to be attractive to the visitor. So to do this the website owner should make his website according to what the users want to find. But a major myth by all owners are we have made our website to be attractive to users as what we want is what the user's want. This is not easy as all kinds of people who will visit your website, and how we gets minds of all. So how will we make to look for the business name or products we offer to the customer? Probably not. There are many way to promote you site affects. The navigation structure of your site should be well enough so that they can find. A good site will continue to change and update contents regularly. Some ways to make website more power full are given below:

- People or the visitors are always attracted to the sites which offer something new every time they open it. This means that you have to update it regularly.
- The search engines are most prevalent today. They include those sites which give relevant content according to the current needs.

### 4. Website Planning

Website planning is an essential step when you decide to create website even if it's your personal webpage. Website planning probably is the most important part of website development process. It is proved that most often problems and error occur because of lack of website planning.

#### Steps of planning a website are:

##### 4.1 Setting up website goals:

It is very essential to define the purpose of the website as one of the first steps in the planning architecture. Goals will find what the websites will accomplish and what users will get from it. A clearly defined purpose will help the rest of the planning purpose as the audience is identified and the content of the site is developed. Setting short and long term goals for the website will help make the purpose clear and plan for the future when expansion, modification and improvement will take place. This helps us tracking the progress of the site and determine its success level.

##### 4.2 Defining website audience:

Defining the audience is a key step in the planning process. A group of people who are expected to visit the website are called audience. These people may be viewing the website for a particular reason. It is important for you to judge that what they are looking for in your website.

- A list of characteristics common to users are:
  - Audience characteristics.
  - Audience information preference.
  - Audience computer specification.
  - Audience web experience. An effective website will be created taking into account the audience.

#### **4.3 Website content planning:**

Content evaluation and organization requires that the purpose of the website be clearly defined and content to design the site structure, content and interactions that are most suitable for a website. Collecting a list of necessary content then organizing it accordingly to the audiences needs is a key step in website planning. It is a good idea to test the content and purpose on a focus group and compare the offerings to the audience needs. The basic information are organized by categorization.

#### **4.4 Document planning:**

Documentation is used to visually plan the site while taking into account the purpose, audience and content, to design the site structure, content and interactions that are most suitable for a website. Documentation is a prototype of a website. This review process increases the likelihood of success of the website. Documentation methods may include using pieces of paper and drawing lines to connect them. Some or all of the individual pages may be designed in greater detail as website wireframe make up model or comprehensive layout. If what the page will actually look like.

### **5. Navigation and Theme**

#### **5.1 Navigation**

If your site has large number of pages, and users will expect web search options to find content in the site. In a larger site, with maybe hundreds or thousands of pages of content, web search is the only efficient means to locate particular content pages or to find all pages that mention a keyword or search phrase. Browse interfaces composed of major site and content landmarks are essential in the initial phases of a user's visit to your site.

Search is the most efficient means to reach specific content, particularly if that content is not heavily visited by other users and is therefore unlikely to appear as a link in a major navigation page. As the user's needs get more specific than a browser interface can handle, search engines are the means to find content out there in the long tail where it might otherwise remain undiscovered as shown in figure below.

Figure 5.1 Search is the best practical navigation in the 'long tail'



#### **5.2 Themes**

Themes can be applied on the site, category, course and activity levels by users with permissions to do so. Themes can be designed for specific devices such as mobile phones or tablets.

You can use contributed themes or create your entire own to share with the community. Themes can also be based on parent themes with only few customizations. Themes accomplish this using CSS, changing the underlying markup structure and also adding JavaScript to add more advanced behaviors.

Most theme developers simply add a few changes to their new theme by basing it on an existing one. For example Moodle, a theme architecture is designed in such a way whereby the base theme will act as a fallback that is used when nothing has been defined in the theme based on it. This makes it easy to create new themes that simply seek out to make minor changes.

When you want to change the appearance of your website, you can simply install a new template. Templates are also sometimes called "themes." You can change templates whenever you like without altering your website's content or taking it offline.

### **6. Web Page Elements**

A *web page*, as an information set, can contain numerous types of information, which is able to be seen, heard or interact by the end user.

Navigation of links on your site plays a role in determining the stickiness of your. The visitors have to see from their point of view as what would they do as and when they open your site. They would probably read the content of your page and would look around for more.

**Importance of Layout and Navigation:** A clean layout and neat navigation and using a lot of white space for your site enhances its looks. Try to keep the focus on your content. All pages or a group of pages should have one basic design and only the content varies. This will make your website more attractive and easy such that a person will turn that this page seems to be like that page so it must be the part of the previous page.

#### **Important Points:**

- Prepare navigation before designing to prevent clustering the website's forgotten links.
- If website having large number of links then it is better to use dropdown menus to save the window area and for better look.
- If you have fireworks or any other software then macromedia drop-down menus are one of their most popular in-built behaviours.
- Maintain an index or a site map to help people what they are looking for.
- The navigation should be flexible enough to accommodate any amount of additional links in case you probably will be adding pages time to time.
- Keep the main links together so that the audience or the visitors could know what your site is conveying.

A basic web page has four main elements

- Hypertext Markup Language
- Text
- Graphics
- Links

#### Hypertext Markup Language

Hypertext Markup Language is the code that browsers interprets to GUI form to "read" a web page. A web page is a Hypertext Markup Language document. When you save your web page in Composer documents are saved with the file extensions .html or .htm. When you save your web page in Composer, the .htm extension will be automatically added. For details see previous chapters.

#### Text

Text consists of headings, descriptions, directions, information that you wish to include in your web page. Text can be added to a web page by typing directly onto the open Composer page.

#### Graphics

Web page graphics are image files (photographs, icons, pictures, cartoons, animations) that you can insert into your web page. Graphic files are used as backgrounds, bullets, page dividers, signposts, illustrations and sometimes, just for fun. Web graphic files typically carry the file extension .gif or .jpg. Graphics can take a long time to load and should be used sparingly.

## 7. WEBSITE DEVELOPMENT AND STEPS OF CREATING WEBSITE

### 7.1 GET YOUR DOMAIN NAME

The first thing you need is to get a domain name. This is the name you want to give to your website. To get a domain name, you have to pay an annual fee to a registrar for the right to use that name. Getting a name does not give you a website or anything like that. It's just a name. It is like registering a business name.

### 7.2 CHOOSE A WEB HOST

A web host is basically a company that has many computers connected to the Internet. When you place your web pages on their computers, everyone in the world will be able to connect to it and view them. You will need to sign up for an account with a web host so that your website has a home. If getting a domain name is analogous to getting a business name, getting a web hosting account is just getting offices or shop for your business.

After you sign up for a web hosting account, you will need to point your domain to that account on your web host.

### 7.3 DESIGN YOUR WEB PAGES

Once you have settled your domain name and web host, next step is design the web site itself. For design a website you must have good knowledge of HTML, DHTML, CSS, JavaScript and any server side language i.e. PHP, ASP.NET or JEE. And creates a fully-functional website with multiple pages and a feedback form, and maintain your site.

## HTML FRAMES FORMS AND WEB DEVELOPMENT

### 7.4 TESTING YOUR WEBSITE

Now you will need to test your web pages as you design them in the major browsers: the latest browser. Directly testing your site in all browsers is the only way you can really be sure that it works the way you want it to on your visitors' machines. Check all validation required.

### 7.5 GETTING YOUR SITE NOTICED

When your site is ready, you can submit it to search engines like Google and Bing. If your site is already linked to by other websites, you may not even need to submit it to these search engines. They will probably find it themselves the links on those websites. Apart from submitting your site to the search engine, you may also promote it in other ways.. advertisements in the newspapers etc. There are some companies which promote your website.

### 8. WEBSITE PUBLISHING:

Web publishing, or "online publishing," is the process of publishing content on the Internet. It includes creating and uploading websites, updating webpages, and posting blogs online. The published content may include text, images, videos, and other types of media.

- To publish a website, you must know what a website must have to be published. A website has a unique address called as domain name. There is a domain name server (DNS) with which every domain name is registered. It identifies a domain name with unique id or password. Also, a web Site website must have an allocated webspace. This is provided by webserver maintained by many companies.
- Look on the web if the name which you selected already is a registered name for a website. After this Upload the content of the web pages as it is to the web server with FTP address, and password provided by service provider.
- Web development software.
- An Internet connection.
- And a web server.

The software may be a professional web design program like Dreamweaver or a simple web-based interface like WordPress. The Internet connection serves as the medium for uploading the content to the web server. Large sites may use a dedicated web host, but many smaller sites often reside on shared servers, which host multiple websites. Most blogs are published on public web servers through a free service like Blogger.

Since web publishing doesn't require physical materials such as paper and ink, it costs almost nothing to publish content on the web. Therefore, anyone with the three requirements above can be a web publisher. Additionally, the audience is limitless since content posted on the web can be viewed by anyone in the world with an Internet connection. These advantages of web publishing have led to a new era of personal publishing that was not possible before.

Web publishing generally refers to uploading content to unique websites.

Once you have built your website the next task is to make sure you've publicized your site as widely as possible. There are two key audience to which you can publicize your website to people who are already communicating with Methods to reach audience.

### 9. WEBSITE PUBLICIZING:

**Exercise****102**

For publicizing yours URL, add your web address to every piece of communication that comes from your organization. Your web address should be located in:

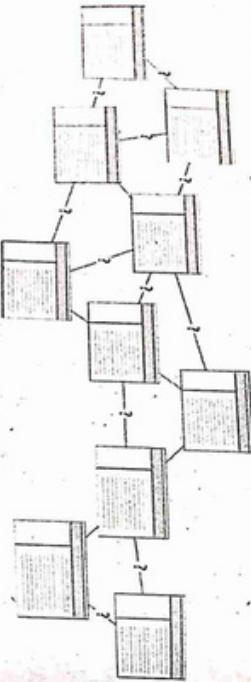
- Business cards
- Newsletters
- Fax cover sheets
- Email Messages: List your web address at the bottom of every email message sent by people in your organization
- Tell your membership If you have a database of members or other who take an active interest in your work, be sure to tell your member about your website.

**10. Structuring the Web Site**

No web site design can be completed until the page structure of the site is established and agreed between the client and the web designer. The structure will dictate the names of the navigation buttons and the linking between the various pages on the web site. Getting the structure right will go a long way to ensuring the visitor enjoys a good experience on the site.

The structure of the site should make it possible for the designer to create a logical and easy follow navigation system to permit the visitor to reach the content they want to see quickly and reliably. It should also make it easy to add fresh content, particularly news items or new products. without major change to the graphic design of the site. A logical, consistently named site organization allows users to make successful predictions about services, without major change to the graphic design of the site.

A logical, consistently named site organization allows users to make successful predictions about services, without major change to the graphic design of the site. Consistent methods of organizing and displaying information permit users where to find things. Consistent methods of organizing and displaying information permit users to extend their knowledge from familiar pages to unfamiliar ones. If you mislead users with a structure that is neither logical nor predictable, or constantly uses different or ambiguous terms to describe features, users will be frustrated by the difficulties of getting around and understanding what you have to offer.



Don't make a confusing web of links. Designers aren't the only ones who make models of sites. Users try to imagine the site structure as well and a successful information architecture will help user build a firm and predictable mental model of your site.

**Very Short Questions**

- Q1. Write down the names of the tags which are used to create frames in HTML.
- Q2. What is roll of <frameset> tag?
- Q3. What is roll of <frame> tag?
- Q4. Write down the values of NORESIZE and SCROLLING attribute and their effects.
- Q5. Write down roll of action attribute of <form> tag.
- Q6. How many submit button can apply on a form?
- Q7. Why <SELECT> tag <OPTION> tag is used in form?
- Q8. Explain difference between working of submit and reset button?
- Q9. What is use of hidden fields in form?
- Q10. What is web publicizing?

**Short Questions**

- Q1. Write a note on <frameset> tag.
- Q2. What are disadvantage of frame?
- Q3. What is roll of method attribute in form? Explain difference between GET and POST methods.  
Which is better?
- Q4. Write down different types of controls of form.
- Q5. Explain different text input controls of form.
- Q6. Why name of different radio button in group is same? Explain with example.
- Q7. How dropdown list and list box are created in HTML, explain with code.
- Q8. How to create web page?
- Q9. Write short note on following:
  - a. Navigation in web site.
  - b. Theme in web site.

- Q10. Write a note on web page elements.
- Q11. Write down steps of web publishing and publicizing.
- Q12. What do you mean by web structuring in web site development?

**Long Questions**

- Q1. What is used of frames in HTML? Explain all tags which are used to create frame in HTML.
- Q2. Write down proper code to create frame in HTML.
- Q3. Why forms are useful? Explain different types of button used in form. Write note on GET and POST methods.
- Q4. Write down steps for planning of web site.
- Q5. Explain all steps of creating and developing of website.

