Unit-III

Basic Interactivity and DHTML

3.1 Frames in HTML: FRAMESET & FRAME Tag & its Attributes

HTML Frames are used to divide the web browser window into multiple sections where each section can be loaded separately. A frameset tag is the collection of frames in the browser window. The window is divided into frames in a similar way the tables are organized: into rows and columns.

Advantages:

- It allows the user to view multiple documents within a single Web page.
- It load pages from different servers in a single frameset.
- The older browsers that do not support frames can be addressed using the tag.

Disadvantages: Due to some of its disadvantage it is rarely used in web browser.

- Frames can make the production of website complicated.
- A user is unable to bookmark any of the Web pages viewed within a frame.
- The browser’s back button might not work as the user hopes.
- The use of too many frames can put a high workload on the server.
- Many old web browser doesn’t support frames.

<Frameset> and <Frame> Tag:

<Frameset> Tag:

The frameset tag defines a row or column in a frameset. Instead of using body tag, use frameset tag in HTML to use frames in web browser. The frameset tag is used to define how to divide the browser. To define the horizontal frames use row attribute of frame tag in HTML document and to define the vertical frames use col attribute of frame tag in HTML document.

Syntax:

<frameset rows=””>   </frameset>

Or

<frameset cols=""> </frameset>

Attributes:

1. **Rows**

This attribute defines the number of rows in the frame. The **HTML <frameset> rows Attribute** is used to *specify the size and the number of rows in a frameset*. The height of each frame is separated by a comma.

Attribute Values:

- **pixels:** The height of row is set in terms of pixels. Example: **"50px"** or **"50"**.
- **%:** The height of row is set in terms of percentage. Example **"70%"**.
- ***:** The height of row is set to all available space.

Syntax

<frameset rows="*pixels|%*|*">

Example:

- in pixel
  <frameset rows = "300, 400, 300">
- In percentage value
  <frameset rows = "30%, 40%, 30%">
- In wild card values:
  <frameset rows = "30%, *, 30%">

2. **Cols**

The **HTML <frameset> cols Attribute** is used to *specify the size and the number of columns in a frameset*. The width of each frame is separated by a comma.

Syntax:

<frameset cols="pixels | % | *">

Example:

- in pixel
  <frameset cols = "300, 400, 300">
- In percentage value

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

- In wild card values:
  <frameset cols = "30%, *, 30%">

3. **border:** This attribute of frameset tag defines the width of the border of each frame in pixels. Zero value is used for no border.

   <frameset border=”10”>

4. **frameborder:** This attribute of frameset tag is used to specify whether a three-dimensional border should be displayed between the frames or not for this use two values 0 and 1, where 0 defines no border and value 1 signifies for yes there will be a border.

   <frameset frameborder=”1”>

5. **framespacing:** This attribute of frameset tag is used to specify the amount of spacing between the frames in a frameset.

   <frameset framespacing="20">

<Frame> Tag

Each frame is indicated by frame tag and it basically defines which HTML document shall open into the frame. A frame should be used within a <frameset> tag.

Attributes

1. **name:** This attribute is used to give names to the frame. It differentiate one frame from another. It is also used to indicate which frame a document should loaded into. **Example:**

   <frame name = "top" src = "frame1.html" >

   <frame name = "main" src = "frame2.html" >

   <frame name = "bottom" src = "frame3.html" >

2. **src:** This attribute in frame tag is basically used to define the source file that should be loaded into the frame.The value of src can be any url. **Example:**

   <frame name = "left" src = "img1.html" >

In the above example name of frame is left and source file will be loaded from “img1.html” in frame.

3. **marginwidth:** This attribute in frame tag is used to specify width of the spaces in pixels between the border and contents of left and right frame.
   Example:
   <frame marginwidth="20">

4. **marginheight:** This attribute in frame tag is used to specify height of the spaces in pixels between the border and contents of top and bottom frame.
   Example:
   <frame marginheight="20">

5. **scrollbar:** To control the appearance of scroll bar in frame use scrollbar attribute in frame tag. This is basically used to control the appearance of scrollbar. The value of this attribute can be yes, no, auto. Where the value no denotes there will be no appearance of scroll bar.
   Example:
   <frame scrollbar="no">

6. **Noresize**: It defines the frame element that cannot be resize by the user.
   Example:
   <frame noresize="noresize">

7. **Frameborder:** Defines if the border around the frame should be displayed or not. For this, we use two values 0 and 1, where 0 defines no border and 1 defines the border.

   Example:
   <frame frameborder="1|0">

3.2 Simple Frame Example, Forms in HTML: Introduction to Forms

Simple Frame Example

Create Horizontal frames:

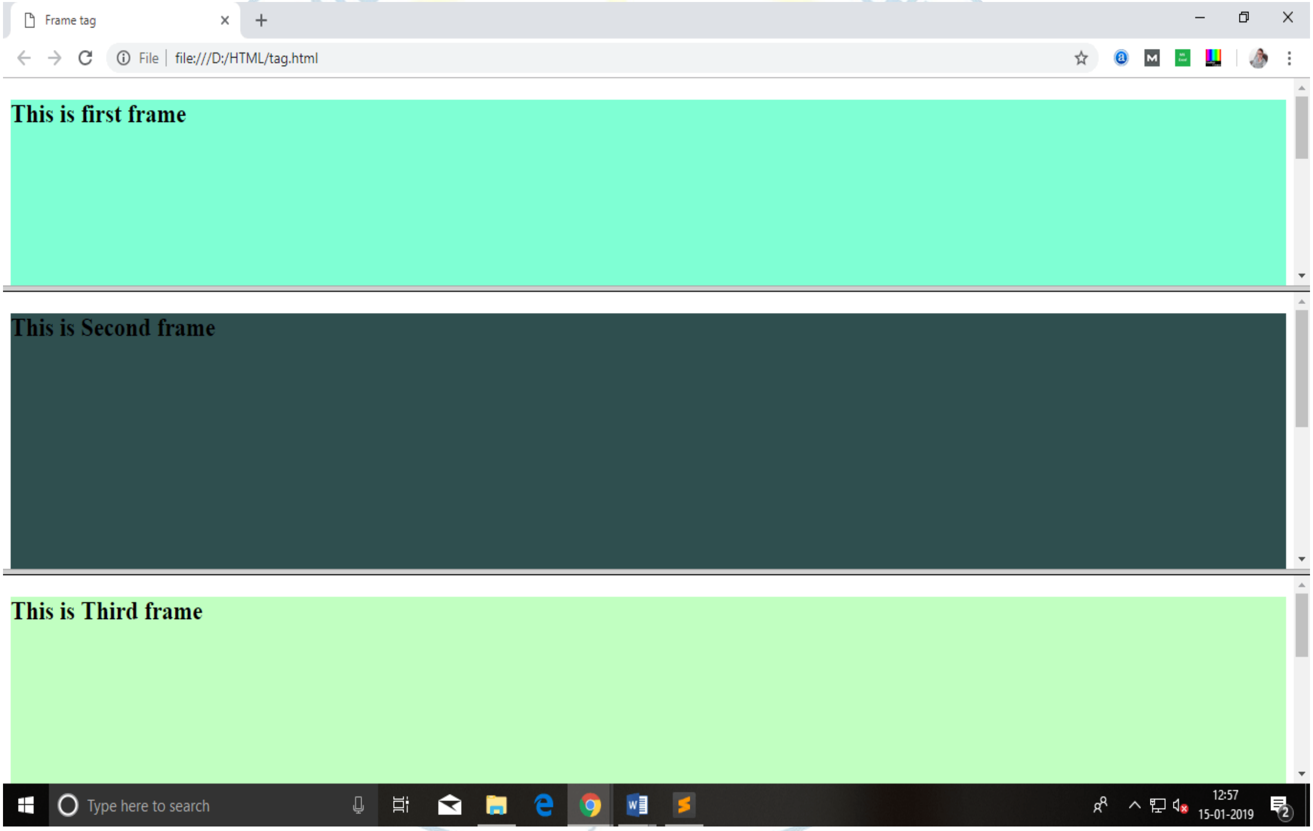
<!DOCTYPE html>

<html>

<head>

```
    <title>Frame tag</title>
</head>
  <frameset rows="30%, 40%, 30%">
    <frame name="top" src="frame1.html" >
    <frame name="main" src="frame2.html">
    <frame name="bottom" src="frame3.html">
  </frameset>
</html>
```

Output:



This is first frame

This is Second frame

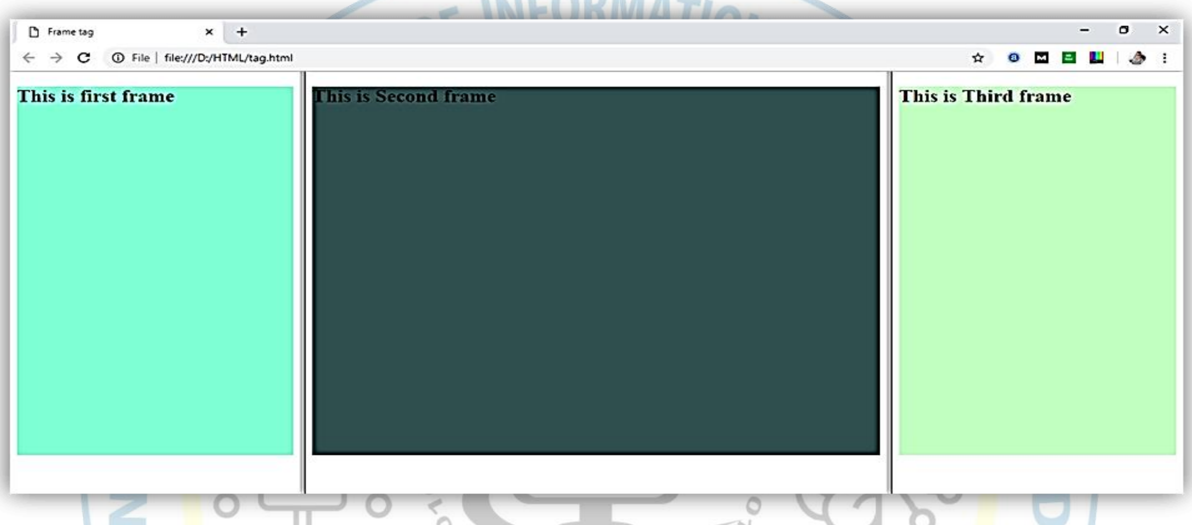
This is Third frame

Create Vertical frames:

```
<!DOCTYPE html>
<html>
<head>
    <title>Frame tag</title>
</head>
```

```
  <frameset cols="25%,50%,25%">
    <frame src="frame1.html" >
    <frame src="frame2.html">
    <frame src="frame3.html">
  </frameset>
</html>
```

Output:



This is first frame

This is Second frame

This is Third frame

Forms in HTML: Introduction to Forms

An **HTML form** is *a section of a document* which contains controls such as text fields, password fields, checkboxes, radio buttons, submit button, menus etc.

An HTML form facilitates the user to enter data that is to be sent to the server for processing such as name, email address, password, phone number, etc.

<form> is an HTML element to collect input data containing interactive controls. It provides facilities to input text, number, values, email, password, and control fields such as checkboxes, radio buttons, submit buttons, etc., or in other words, form is a container that contains input elements like text, email, number, radio buttons, checkboxes, submit buttons, etc. Forms are generally used when you want to collect data from the user. For example, a user wants to buy a bag online, so he/she has to first enter their shipping address in the address form and then add their payment details in the payment form to place an order.

Common Uses of Forms include

- Comment Response
- Online Order Forms
- Subscription Forms
- Registration Forms

Forms are created by placing input fields within paragraphs, preformatted text, lists and tables. This gives considerable flexibility in designing the layout of forms.

Syntax:

```
<form>

    <!--form elements-->

</form>
```

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

```
<form action = "Script URL" method = "GET|POST">
  form elements like input, textarea etc.
</form>
```

3.3 Form elements & it's attributes

<u>Form elements</u>

These are the following HTML <form> elements:

1. **input:** This is one of the most used form elements. It is used in several ways depending on what type of field we want to define.

    ```
    <input type="text" id="fname" name="fname">
    ```

2. **label:** It is used to define the label for the field that we have defined. It is useful when the user tries to focus on the **input** element. So we use the **label** element with the **input** element, as the 'for' attribute of the **label** element is equal to the 'id' attribute of the **input** element.

    ```
    <label for="fname" >FirstName </label>
    ```

3. **select:** It is used to create a drop-down list. This allows users to select as many options as the user wants. The behavior of the attribute can be changed according to the situations while we are using this element to create a drop-down list.

```
<select>
<option>cars</option>
<option>Malls</option>
</select>
```

4. **textarea:** This is used to create a multi-line text field. This is used mainly in a form, to collect reviews and comments. The size of the text area can be specified by the attributes like cols and rows. So the cols attribute is used to specify the width and the row attribute is used to specify the number of lines in the text area.

```
<textarea row="4" cols="50"></textarea>
```

5. **button:** This is used to create a clickable button. It has one attribute that is the 'type' attribute. This attribute is used to specify the button that will be used on the page depending upon the situation.

```
<button type="submit">submit</button>
```

Form Attributes

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

1. **action**

The action attribute of <form> element defines the process to be performed on form when form is submitted, or it is a URI to process the form information. The action attribute value defines the web page where information proceed. It can be .php, .jsp, .asp, etc. or any URL where you want to process your form.

<form action="action.html">

2. **method**

The method attribute defines the HTTP method which browser used to submit the form. The possible values of method attribute can be:

- **get:** The get value of method attribute is default value while submitting the form. But this is not secure as it displays data in URL after submitting the form.

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

- **post:** We can use the post value of method attribute when we want to process the sensitive data as it does not display the submitted data in URL.

  Example:

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

3. value

The HTML value attribute defines the initial value or default value of an input field.

Example:

```
<input type="text" name="uname" value="Enter Name">
```

4. name

The HTML name attribute defines the name of an input element. The name and value attribute are included in HTTP request when we submit the form.

```
<input type="name" name="uname">
```

5. size

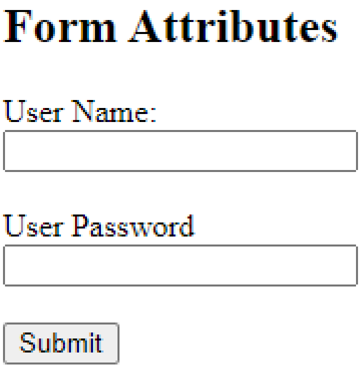
The size attribute controls the size of the input field in typed characters.

```
<input type="text" name="uname" size="40">
```

Example:

```
<!DOCTYPE html>
<html>
 <body>
        <h2>Form Attributes</h2>
          <form action="action.html" method="post">
                    <label>User Name:</label><br>
                    <input type="text" name="name"><br><br>
                    <label>User Password</label><br>
                    <input type="password" name="pass"><br><br>
                     <input type="submit">
             </form>
  </body>
</html>
```

Output:



3.4 Form Controls: Text Controls, Password Field, Multiline Text Input

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.

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.

Attributes

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

1. **type**

The HTML <input> type Attribute is used to specify the type of <input> element to display. The default type of <input> type attribute is text.

Syntax:

<input type="value">

Example:

<input type=”text”>

2. **name**

The **HTML <input> name Attribute** is used to specify a name for an <input> element and get the value.

Syntax:

<input name="name">

Example:

<input name="fname">

3. **value**

The **value** attribute for <input> element in HTML is used to specify the initial value of the input element.

Syntax:

<input value = "text">

Example:

<input value="John">

4. **size**

The **HTML <input> size Attribute** is used to specify the visible width in terms of characters of an <input> element. It works with many input fields such as text, search, tel, URL, email, and password.

Syntax:

<input size="number">

Example:

<input size=”50”>

5. **Maxlength**

The **<input> maxlength attribute** is used to specify the maximum number of characters enters into the <input> element.

Syntax:

```
<input maxlength="number">
```

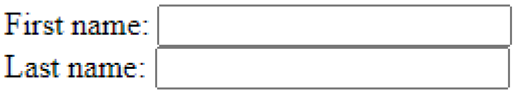
Example:

```
<input maxlength="10">
```

<u>Example</u>

```
<! DOCTYPE html>
<html>
  <head>
    <title>Text Input Control</title>
  </head>
  <body>
    <form >
      First name: <input type = "text" name = "first_name" size=”25” maxlength=”30” >
      <br>
      Last name: <input type = "text" name = "last_name" size=”25” maxlength=”30” >
    </form>
  </body>
</html>
```

Output:



First name:

Last name:

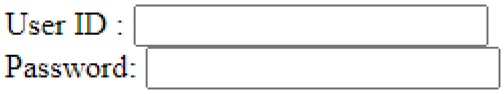
<u>Password input controls</u>

This is also a single-line text input but it masks the character as soon as a user enters it. It displays *****(asterisks) instead of actual input. They are also created using HTML <input>tag but type attribute is set to **password**.

Example

```
<! DOCTYPE html>
<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"  size=”25” maxlength=”30” >
    </form>
  </body>
</html>
```

Output:



User ID :

Password:

Multiple-Line Text Input Controls

This is used when the user is required to give details that may be longer than a single sentence. The **HTML <textarea> tag** is used to define a multi-line text input control. It can hold unlimited number of characters and the texts are displayed in a fixed-width font. The size of the HTML textarea is defined by <cols> and <rows> attributes.

Syntax

<textarea rows="" cols=""> Contents... </textarea>

Attributes

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

1. **name**

   It holds the name to input control.

2. **rows**

   It specifies height of textarea.

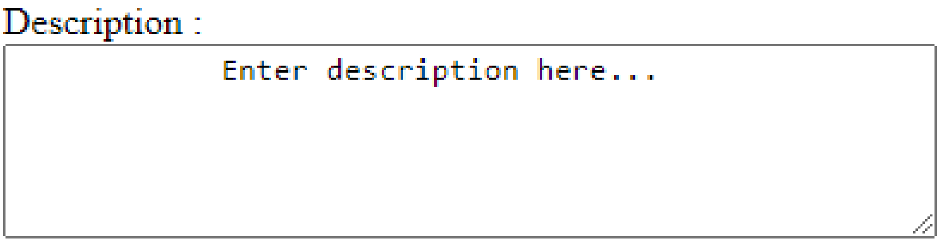
3. **cols**

   It specifies width of textarea.

Example

```
<! DOCTYPE html>
<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>
```

Output:



Description :

```
Enter description here...
```

1. Pull Down Menus, Check Box, Radio Buttons, Scrolled Lists

With HTML, you can create a simple drop-down list of items to get user input in HTML forms. A select box also called drop-down box provides an option to list down various options in the form of drop-down list, from where a user can select one or more options.

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.**

Attributes

Following is the list of attributes for <checkbox> tag.

1. **type**

   Indicates the type of input control and for checkbox input control it will be set to checkbox.

2. **name**

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

3. **value**

   The value that will be used if the checkbox is selected.

4. **checked**

   Set to checked if you want to select it by default.

Example

```
<!DOCTYPE html>
<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>
```

Output:

```
□ Maths □ Physics
```

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 <input> tag but type attribute is set to **radio**.

Attributes

Following is the list of attributes for radio button.

1. **type**

   Indicates the type of input control and for checkbox input control it will be set to radio.

2. **Name**

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

3. **value**

   The value that will be used if the radio box is selected.

4. **checked**

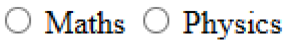
   Set to *checked* if you want to select it by default.

Example

```
<!DOCTYPE html>
<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>
```

Output


○ Maths ○ Physics

Scrolled List / Select Box 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.

Syntax

```
<select>
    <option></option>
 </select>
```

Attributes

Following is the list of important attributes of <select> tag –

1. **Name**

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

2. **Size**

   This can be used to present a scrolling list box.

3. **Multiple**

   If set to "multiple" then allows a user to select multiple items from the menu.

HTML <option> tag

HTML <option> tag is used to define options in a dropdown list within <select> or <datalist> element. A dropdown list must have at least one <option> element.

Syntax

<option value="">.....**</option>**

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

1. **Value**

   The value that will be used if an option in the select box box is selected.

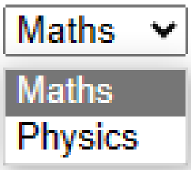
2. **Selected**

   Specifies that this option should be the initially selected value when the page loads.

Example

```
<!DOCTYPE html>
<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>
```

Output:



2. Reset Button and Submit Button

Buttons are defined by changing the input element’s type attribute. There is two different kinds of buttons – the submit button and the reset button.

Reset Button

The reset button is used to reset all the input fields in HTML forms. Basically, the reset button is used to reset all the form data values and set them to their initial default value. In case of user entered the wrong data then the user can easily correct it by clicking on the “Reset Button”. The reset button is used to clear all inputs by the user.

Syntax

```
<input type="reset" value="Reset" >
```

Submit Button

The submit button is used whenever you want to submit. The **HTML <input type= ”submit”>** is used to define a submit button. It is used to submit all the user value to the form handler.

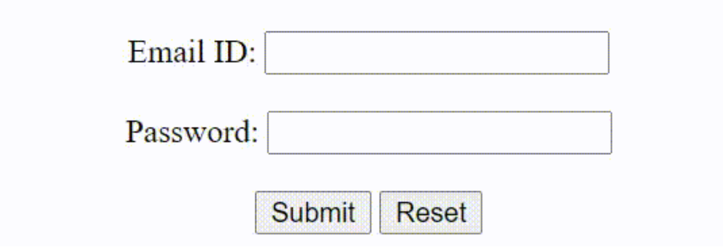
Syntax:

```
<input type="submit" value=”Submit”>
```

Example

```
 <form>
        Email ID:
        <input type="email">
        <br>
        <br> Password:
        <input type="password">
        <br><br>
        <input type="submit">
        <input type="reset">
    </form>
```

Output:



The submit-button must always be the last value selected, as it initiates the form submission whereas the reset-button can be pressed at all times during the form fill out.

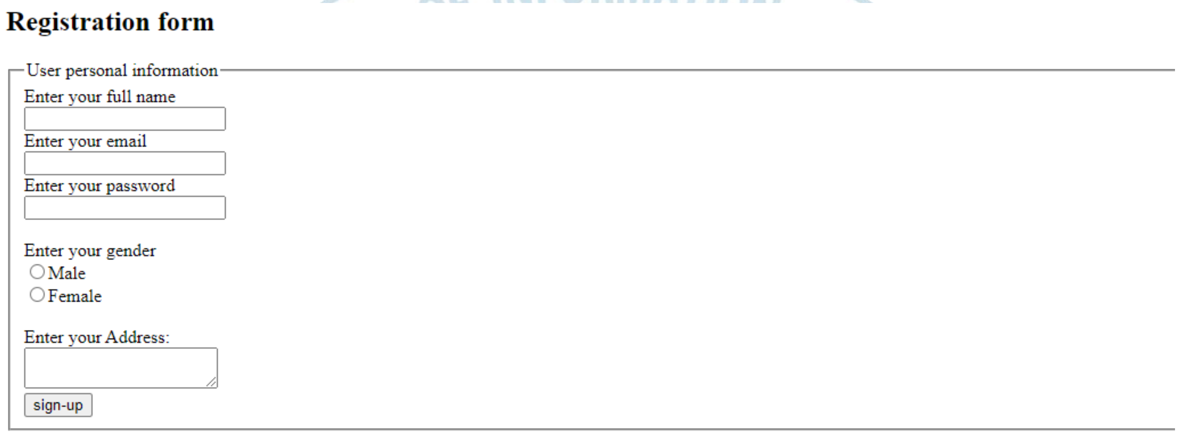
How to create Registration page using form attributes

```
<!DOCTYPE html>
 <html>
 <head>
  <title>Form in HTML</title>
</head>
 <body>
     <h2>Registration form</h2>
    <form>
     <fieldset>
        <legend>User personal information</legend>
        <label>Enter your full name</label><br>
        <input type="text" name="name"><br>
         <label>Enter your email</label><br>
         <input type="email" name="email"><br>
         <label>Enter your password</label><br>
         <input type="password" name="pass"><br>

         <br><label>Enter your gender</label><br>
         <input type="radio" id="gender" name="gender" value="male"/>Male  <br>
         <input type="radio" id="gender" name="gender" value="female"/>Female <br/>
```

```
          <br>Enter your Address:<br>
         <textarea></textarea><br>
         <input type="submit" value="sign-up">
      </fieldset>
   </form>
  </body>
</html>
```

Output:



Registration form

User personal information
Enter your full name
Enter your email
Enter your password
Enter your gender
○Male
○Female
Enter your Address:

3.5 Introduction of DHTML, Ramifications of DHTML

DHTML stands for Dynamic HTML, it is totally different from HTML. Dynamic HTML is not a markup or programming language but it is a term that combines the features of various web development technologies for creating the web pages dynamic and interactive.

The browsers which support the dynamic HTML are some of the versions of Netscape Navigator and Internet Explorer of version higher than 4.0. The DHTML is based on the properties of the HTML, javascript, CSS, and DOM (Document Object Model which is used to access individual elements of a document) which helps in making dynamic content. It is the combination of HTML, CSS, JS, and DOM. The DHTML make use of Dynamic object model to make changes in settings and also in properties and methods.

DHTML is used to create interactive and animated web pages that are generated in real-time, also known as dynamic web pages so that when such a page is accessed, the code within the page is analyzed on the web server and the resulting HTML is sent to the client's web browser.

Ramification of DHTML

DHTML consists of the following four components or languages:

- HTML 4.0
- CSS
- JavaScript
- DOM.

HTML 4.0

HTML is a client-side markup language, which is a core component of the DHTML. It defines the structure of a web page with various defined basic elements or tags.

CSS

CSS stands for Cascading Style Sheet, which allows the web users or developers for controlling the style and layout of the HTML elements on the web pages.

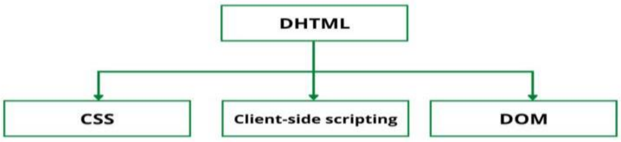
JavaScript

JavaScript is a scripting language which is done on a client-side. The various browser supports JavaScript technology. DHTML uses the JavaScript technology for accessing, controlling, and manipulating the HTML elements. The statements in JavaScript are the commands which tell the browser for performing an action.

DOM

DOM is the document object model. It is a w3c standard, which is a standard interface of programming for HTML. It is mainly used for defining the objects and properties of all elements in HTML.

DHTML is not a technology; rather, it is the combination of three different technologies, client-side scripting (JavaScript or VBScript), cascading style sheets and document object model.



Difference between HTML and DHTML

Following table describes the differences between HTML and DHTML:

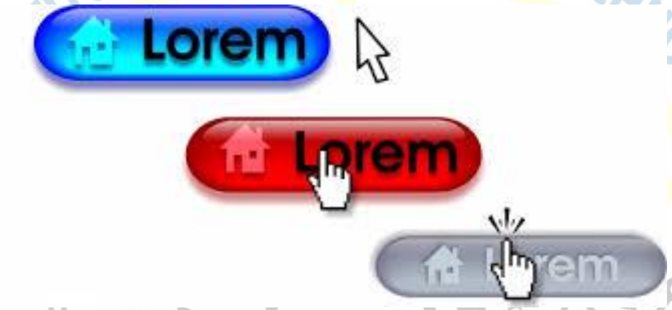
HTML (Hypertext Markup language)	DHTML (Dynamic Hypertext Markup language)
1. HTML is simply a markup language.	1. DHTML is not a language, but it is a set of technologies of web development.
2. It is used for developing and creating web pages.	2. It is used for creating and designing the animated and interactive web sites or pages.
3. This markup language creates static web pages.	3. This concept creates dynamic web pages.
4. It does not contain any server-side scripting code.	4. It may contain the code of server-side scripting.
5. The files of HTML are stored with the .html or .htm extension in a system.	5. The files of DHTML are stored with the .dhtm extension in a system.
6. A simple page which is created by a user without using the scripts or styles called as an HTML page.	6. A page which is created by a user using the HTML, CSS, DOM, and JavaScript technologies called a DHTML page.
7. This markup language does not need database connectivity.	7. This concept needs database connectivity because it interacts with users.

3.6 Rollover Buttons

A rollover button is a dynamic Web button that changes appearance depending on the location of the user's mouse pointer. It contains three states: normal, over and down. The normal state appears when your mouse is off the button, the over state applies when your mouse rolls over the button, and the down state applies to when you click on the button.

To create a rollover button, you will first need atleast two to represent each of the button's states inactive, active and unavailable.

A sample set of rollover images.



Rollover buttons come in different shapes, sizes, colors and styles. Since images can be converted to rollover buttons, a rollover button can also contain images.

Rollover buttons are used primarily as navigational buttons on a Web page to direct people to other locations. They are also used in drop-down and pop-up menus. Other rollover buttons are used for animated effects and sounds, so an image, color, shape, text or sound can change as the user rolls over the button on the Web.

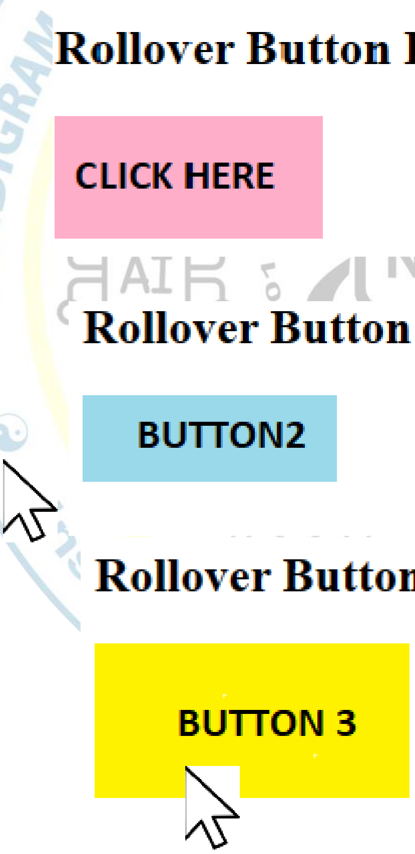
Rollover buttons extend the normal functionality of a button. Not only are they visually attractive, but rollover buttons are also dynamic, altering the user to the fact that they're active elements on the page. A rollover button can be created with image-editing software, such as Adobe Photoshop, Image Ready, Flash and Fireworks. Web-editing software programs like Dreamweaver and Microsoft Visual Studio Developer can also be used to create rollover button effects.

Example: Rollover Button on Web page

```
<!DOCTYPE html>
```

```
<html>
  <head>
      <title>  Rollover button  in Html</title>
  </head>
   <body>
      <h2>Rollover t in Html</h2>
<img      src="C:\Users\Admin\Desktop\B1.PNG"      onmouseover="this.src='B2.PNG'"
onmouseout="this.src='B3.PNG'" />
   </body>
</html>
```

Output:



Rollover Button Effect in Html

Rollover Button Effect in Html

Rollover Button Effect in Html