

UNIT - 2

❖ HTML

HTML is the building block for web pages. HTML is a format that tells a computer how to display a web page. 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 stands for Hyper Text MarkupLanguage
- An HTML file is a text file containing small markup tags
- The markup tags tell the Web browser how to display the page
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- An HTML file must have an htm or html file extension

❖ HTML Browsers

The purpose of a web browser (Chrome, Edge, Firefox, Safari) is to read HTML documents and display them correctly.

A browser does not display the HTML tags, but uses them to determine how to display the document:

1. Internet Explorer



Internet Explorer is a product of microsoft. This is the most commonly used browser in the universe. This was introduced in 1995 along with windows 95 launch and it has passed Netscape popularity in 1998.

2. Netscape



Netscape is one of the original web browser. Microsoft designed internet explorer to complete against Netscape. Netscape was introduced in 1994.

3. Mozilla



Mozilla is an open-source web browsers, designed for standard performance and portability. Browsers based on mozilla code is the second largest browser family on the internet today.

4. Firefox



Firefox is a new browser derived from mozilla. It was released in 2004 and has grown to be the second most popular browser on the internet.

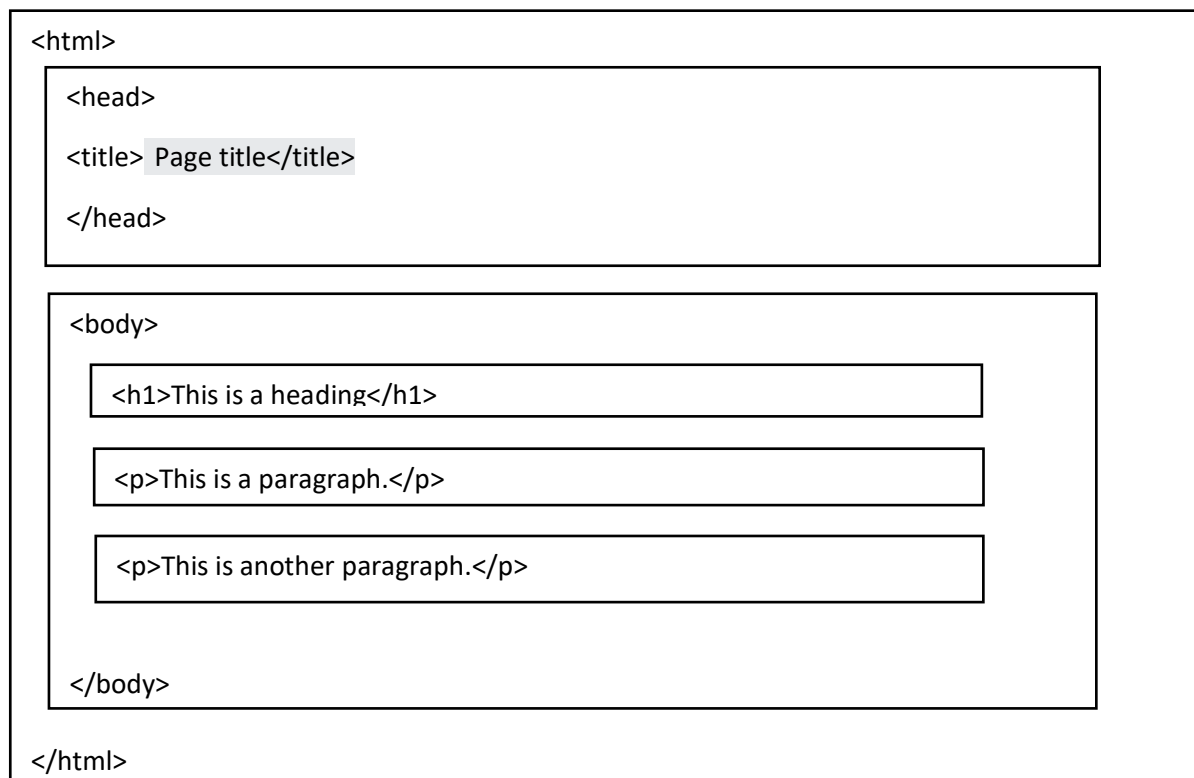
5. Opera



Opera is smaller and faster than most other browsers. Fast, user-friendly with keyboard interface, multiple windows, zoom functions and more.

❖ HTML Page Structure

Below is a visualization of an HTML page structure:



❖ HTML Tags

HTML tags are like keywords which defines that how web browser will format and display the content. With the help of tags, a web browser can distinguish between an HTML content and a simple content. HTML tags contain three main parts: opening tag, content and closing tag. But some HTML tags are unclosed tags

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 and The first tag in a pair is the start tag, the second tag is the end tag . The text between the start and end tags is the element content . HTML tags are not case sensitive, means the same as

The most important tags in HTML are tags that define headings, paragraphs and line breaks.

	Defines an unordered list
	Defines an ordered list
	Defines a list item
<dl>	Defines a description list
<dt>	Defines a term/name in a description list
<dd>	Defines a description/value of a term in a description list
<form>	Defines an HTML form for user input

<input>	Defines an input control
<textarea>	Defines a multiline input control (text area)
<label>	Defines a label for an <input> element
<fieldset>	Groups related elements in a form
<legend>	Defines a caption for a <fieldset> element
<select>	Defines a drop-down list
<optgroup>	Defines a group of related options in a drop-down list
<option>	Defines an option in a drop-down list
<button>	Defines a clickable button
<p>	Defines a paragraph
 	Defines a single line break
<hr>	Defines horizontal row
<hrsize>	Defines horizontal size
<!DOCTYPE>	Defines the document type
<!--...-->	Defines a comment
<a>	Defines a hyperlink
<title>	Defines a title for the document
<imgsrc>	Adding image links
	Defines bold text
<i>	Defines italic text
<u>	underline

<table>	Defines a table
<tt>	Defines teletype text
<td>	Defines a cell in a table
<tr>	Defines a row in a table
<th>	Defines a header cell in a table
<thead>	Groups the header content in a table
<body>	Defines the document's body
<head>	Contains metadata/information for the document
<h1> to <h6>	Defines HTML headings
<header>	Defines a header for a document or section
<footer>	Defines a footer for a document or section
<sub>	Defines subscripted text
<sup>	Defines superscripted text
<big>	Defines big text
<small>	Defines smaller text
<caption>	Defines a table caption
<colgroup>	Specifies a group of one or more columns in a table for formatting
<col>	Specifies column properties for each column within a <colgroup> element
	Defines a section in a document
<tbody>	Groups the body content in a table
<tfoot>	Groups the footer content in a table

	Defines text that has been deleted from a document
<abbr>	Defines an abbreviation or an acronym
<bdo>	Overrides the current text direction
<pre>	Defines preformatted text
<address>	Defines contact information for the author/owner of a document
<blockquote>	Defines a section that is quoted from another source
<marquee>	Defines slide animation for text

Simple example of HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Title of the document</title>
</head>
<body>
  <h1>This is a heading</h1>
  <p>This is a paragraph.</p>
</body>
</html>
```

Output :

This is a heading

This is a paragraph

❖ HTML URL

URL stands for Uniform Resource Locator. It is actually a web address. A URL can contain words i.e. (javatpoint.com) or an Internet Protocol (IP) address i.e.195.201.68.81. But most of the user use URL in the form of words because it is easy to remember than numbers.

Syntax :

scheme://**prefix.domain:port/path/filename**

- **scheme** is used to define the type of Internet service (most common is http or https).
- **prefix** is used to define a domain prefix (default for http is www).
- **domain** is used to define the Internet domain name (like javaTpoint.com).
- **port** is used to define the port number at the host (default for http is 80).
- **path** is used to define a path at the server (If omitted: the root directory of the site).
- **filename** is used to define the name of a document or resource.

Following is a list of some common types of schemes used in URL:

- http(HyperText Transfer Protocol):Common web pages. Not encrypted.
- https (Secure HyperText Transfer Protocol):Secure web pages. Encrypted.

- ftp(File Transfer Protocol): Downloading or uploading files.
- file: A file on your computer.
- **HTML tag**

The <html> tag represents the root of an HTML document.

The <html> tag is the container for all other HTML elements (except for the <!DOCTYPE> tag).

- **Head Tag**

The <head> element is a container for metadata (data about data) and is placed between the <html> tag and the <body> tag.

Metadata is data about the HTML document. Metadata is not displayed.

Metadata typically define the document title, character set, styles, scripts, and other meta information.

The following elements can go inside the <head> element:

- <title> (required in every HTML document)
- <style>
- <base>
- <link>
- <meta>
- <script>
- <noscript>

- **Title Tag**

The <title> tag defines the title of the document. The title must be text-only, and it is shown in the browser's title bar or in the page's tab.

The <title> tag is required in HTML documents!

The contents of a page title is very important for search engine optimization (SEO)! The page title is used by search engine algorithms to decide the order when listing pages in search results.

The <title> element:

- defines a title in the browser toolbar
- provides a title for the page when it is added to favorites
- displays a title for the page in search-engine results

- **Body Tag**

The <body> tag defines the document's body.

The <body> element contains all the contents of an HTML document, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.

- **Header Tag**

The <header> element represents a container for introductory content or a set of navigational links.

A <header> element typically contains:

- one or more heading elements (<h1> - <h6>)

- logo or icon
- authorship information

- **Footer Tag**

The <footer> tag defines a footer for a document or section.

A <footer> element typically contains:

- authorship information
- copyright information
- contact information
- sitemap
- back to top links
- related documents

Different ways to format text using tags :

**1. **

It specifies size, color and font face of the text.

- **Attributes :**

Color : specifies the color for text.

Size : specifies the font size.

Face : specifies font names separated by commas.

Example :

 Font tag
example

2. Heading tags <h1> to <h6>

The <h1> to <h6> tags define HTML headings.

<h1> Defines the most important heading .

<h6> Defines the least important heading

Browser displays headings in larger and bolder font than normal text

Example:

```
<h1> Hello </h1>
```

```
<h3> How r u </h3>
```

Output :

Hello

How r u

3. Physical character formatting

- : Displays bold text
- <i> : displays italic text
- <u> : Displays underline text

4. Superscript and subscript formatting

- <sub> : Displays a subscript. Subscript makes text smaller and lowers it below the baseline.

Example : H₂O

H₂O

Output : H₂O

- `<sup>` : Displays a superscript. Superscript formatting makes text smaller and raises it off the baseline

Example : H⁸⁹⁰T

H`⁸⁹⁰`T

Output: H⁸⁹⁰T

5. Other physical formatting tags

- `` : Displays Strike through.

Example : ` Strike `

Output : ~~Strike~~

- `<small>` : Displays text small than usual.

Example : This is `<small> small </small>`

Output : This is small

- `<big>` : Displays text larger than usual.

Example : This is `<big> Big </big>`

Output : This is Big

Background and foreground color

Many web designers avoid dark background on web pages because they make text reading difficult. For example : in www.google.com body text is almost black on a white background.

Specifying colors

There are various ways to specify colors. Some of which are :

1. Named color : the 16 basic colors can be used to apply foreground or background color. For example : red, green, black, pink, gray, white, blue.
2. rgb (red, green, blue) : using this method we can specify RGB value from 0 to 255. For eg. For red=>rgb(255, 0, 0), for yellow=>(255, 255, 0)
3. Hexadecimal RGB value : we can also use “#rrggbb” format to specify color. Here RGB value can range from 00 (0) to ff (255). For eg. For red “#ff0000”, for yellow “#ffff00”.

```
<html>
  <style type="text/css">
    Body
    {
      Color : rgb (255, 255, 0);
      Background : rgb (200, 200, 200);
    }
  </style>
  <body>
    <h1> specifying colors </h1>
  </body>
</html>
```

Applying a background color

- To specify a background color for an entire page we use background-color or background property.
- For example : <body style="background-color:yellow"> make the background of an entire page yellow.

- We can also use rgb() method or hexadecimal RGB value to apply background color

`<body style="background-color:#FFFF00">`

`<body style="background-color:rgb(255, 255, 0)">`

Applying a foreground color

- The foreground color specifies text color. To set foreground color we can use color property.
- For example : `<body style="color:#FFFF00">`

`<body style="color:rgb(255, 255, 0)">`

❖ Lists

HTML offers web authors three ways for specifying lists of information.

All lists must contain one or more list elements. Lists are of three types

- 1) Un ordered list
- 2) Ordered List
- 3) Definition list

HTML Unordered Lists:

- Unordered list is used to display list of items where ordering is not important.
- It is enclosed between `` and `` tags and each item in a list is enclosed within `` and `` tags.

Example:

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
ul
{
List-style-type: square;
}
</style>
</head>
<body>
<ul> <li> Ginger </li>
<li>Radish </li>
<li> Beetroot </li>
<li> Potato </li>
</ul>
</body>
</html>
```

Output :

- Ginger
- Radish
- Beetroot
- Potato

HTML Ordered Lists:-

- Ordered lists are used to display a list of items which are numbered.
- It is enclosed between and .
- Each item in the list is enclosed between and .

Example :

```
<!DOCTYPE html>

<html>
<head>
<style type="text/css">
ol
{
List-style-type: lower-alpha;
}
</style>
</head>
<body>
<ol>
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ol>
</body>
```

</html>

Output :

- a. Beetroot
- b. Ginger
- c. Potato
- d. Radish

HTML Definition Lists:-

- Definition list is not list of items. It is list of terms and definition of terms.
- Definition list is enclosed between <dl> and </dl>, term is enclosed between <dt> and </dt> and their definition is enclosed between <dd> and </dd>.

Example :

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<dl>
```

```
<dt>coffee</dt>
```

```
<dd>Black Hot Water </dd>
```

```
<dt>Milk</dt>
```

```
<dd>White Cold Water </dd>
```

```
</dl>
```

```
</body>
```

</html>

Output :

coffee
Black Hot Water
Milk
White Cold Water

❖ HTML tables:

- The <table> tag defines an HTML table.
- An HTML table consists of one <table> element and one or more <tr>, <th>, and <td> elements.
- The <tr> element defines a table row, the <th> element defines a table header, and the <td> element defines a table cell.
- An HTML table may also include <caption>, <colgroup>, <thead>, <tfoot> and <tbody> elements.

Example:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>HTML Tables</title>
```

```
</head>
<body>
<table border="1">
<tr>
<td>Row 1, Column 1</td>
<td>Row 1, Column 2</td>
</tr>
<tr>
<td>Row 2, Column 1</td>
<td>Row 2, Column 2</td>
</tr>
</table>
</body>
</html>
```

Output :

Row 1, Column 1	Row 1, Column 2
Row 2, Column 1	Row 2, Column 2

Table Heading: Table heading can be defined using <th>tag. This tag will be put to replace <td> tag, which is used to represent actual data cell. Normally you will put your top row as table heading as shown below, otherwise you can use <th> element in any row.

Tables Backgrounds: set table background using one of the following two ways:

1)bgcolor attribute - You can set background color for whole table or just for one cell.

2)background attribute - You can set background image for whole table or just for one cell. You can also set border color also using bordercolor attribute.

Example:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>HTML Tables</title></head>
```

```
<body>
```

```
<table border="1"bordercolor="red" bgcolor="yellow">
```

```
<tr>
```

```
<th>Name</th>
```

```
<th>Salary</th>
```

```
</tr>
```

```
<td>Jayapal </td>
<td>50,000.00</td>
</tr>
<tr>
<td>Ravi</td>
<td>45,000.00</td>
</tr>
</table>
</body>
</html>
```

Output :

Name	Salary
Jaypal	50,000.00
Ravi	45,000.00

❖ Insert Image:

insert any image in the web page by using tag.

Attribute values :

Left - Align the image to the left

Right - Align the image to the right

middle - Align the image in the middle

top - Align the image at the top

bottom - Align the image at the bottom

Example :

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Using Image in Webpage</title>
```

```
</head>
```

```
<body><p>Simple Image Insert</p>
```

```

```

```
</body>
```

```
</html>
```

Output :

Sample Image Insert



❖ Creation Of Link

HTML links are hyperlinks.

The HTML <a> tag defines a hyperlink.

It has the following syntax:

```
<a href="url">Link text</a>
```

This example shows how to create a link to schoolbus.com:

```
<a href="https://www.schoolbus.com/">Visit  
schoolbus.com!</a>
```


❖ HTML FORMS:

HTML Forms are required to collect some data from the site visitor. For example, during user registration you would like to collect information such as name, email address, credit card, etc. A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application 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, text area fields, drop-down menus, radio buttons, checkboxes, etc.

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

The <form> Element

The HTML <form> element is used to create an HTML form for user input:

```
<form>  
.  
form elements  
.  
</form>
```

The `<form>` element is a container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc.

The `<input>` Element

The HTML `<input>` element is the most used form element.

An `<input>` element can be displayed in many ways, depending on the type attribute.

Here are some examples:

Type	Description
<code><input type="text"></code>	Displays a single-line text input field
<code><input type="radio"></code>	Displays a radio button (for selecting one of many choices)
<code><input type="checkbox"></code>	Displays a checkbox (for selecting zero or more of many choices)
<code><input type="submit"></code>	Displays a submit button (for submitting the form)
<code><input type="button"></code>	Displays a clickable button

HTML Form Attributes

- The Action Attribute

The action attribute defines the action to be performed when the form is submitted.

Usually, the form data is sent to a file on the server when the user clicks on the submit button.

- The Target Attribute

The target attribute specifies where to display the response that is received after submitting the form.

The target attribute can have one of the following values:

Value	Description
<code>_blank</code>	The response is displayed in a new window or tab
<code>_self</code>	The response is displayed in the current window
<code>_parent</code>	The response is displayed in the parent frame
<code>_top</code>	The response is displayed in the full body of the window
<code>framename</code>	The response is displayed in a named iframe

The default value is `_self` which means that the response will open in the current window.

- The Method Attribute

The method attribute specifies the HTTP method to be used when submitting the form data.

The form-data can be sent as URL variables (with method="get") or as HTTP post transaction (with method="post").

The default HTTP method when submitting form data is GET.

- The Autocomplete Attribute

The autocomplete attribute specifies whether a form should have autocomplete on or off.

When autocomplete is on, the browser automatically complete values based on values that the user has entered before.

- The Novalidate Attribute

The novalidate attribute is a boolean attribute.

When present, it specifies that the form-data (input) should not be validated when submitted.

HTML Form Elements

- The <input> Element

One of the most used form element is the <input> element.

The <input> element can be displayed in several ways, depending on the type attribute.

- The <label> Element

The <label> element defines a label for several form elements.

The <label> element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focus on the input element.

The <label> element also help users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the <label> element, it toggles the radio button/checkbox.

- The <select> Element

The <select> element defines a drop-down list

- The <textarea> Element

The <textarea> element defines a multi-line input field (a text area)

- The <button> Element

The <button> element defines a clickable button

- The <fieldset> and <legend> Elements

The <fieldset> element is used to group related data in a form.

The <legend> element defines a caption for the <fieldset> element.

- The <datalist> Element

The <datalist> element specifies a list of pre-defined options for an <input> element.

- The <output> Element

The <output> element represents the result of a calculation (like one performed by a script)

HTML Input Types

- Input Type Text

<input type="text"> defines a single-line text input field

- Input Type Password

<input type="password"> defines a password field

- Input Type Submit

<input type="submit"> defines a button for submitting form data to a form-handler.

- Input Type Reset

<input type="reset"> defines a reset button that will reset all form values to their default values

- Input Type Radio

<input type="radio"> defines a radio button.

Radio buttons let a user select ONLY ONE of a limited number of choices

- Input Type Checkbox

`<input type="checkbox">` defines a checkbox.

Checkboxes let a user select ZERO or MORE options of a limited number of choices.

- Input Type Button

`<input type="button">` defines a button

- Input Type Color

The `<input type="color">` is used for input fields that should contain a color.

- Input Type Date

The `<input type="date">` is used for input fields that should contain a date

- Input Type Datetime-local

The `<input type="datetime-local">` specifies a date and time input field, with no time zone.

- Input Type Email

The `<input type="email">` is used for input fields that should contain an e-mail address.

- Input Type Image

The `<input type="image">` defines an image as a submit button.

- Input Type File

The `<input type="file">` defines a file-select field and a "Browse" button for file uploads.

- Input Type Hidden

The `<input type="hidden">` defines a hidden input field (not visible to a user).

A hidden field lets web developers include data that cannot be seen or modified by users when a form is submitted.

A hidden field often stores what database record that needs to be updated when the form is submitted.

- Input Type Month

The `<input type="month">` allows the user to select a month and year.

- Input Type Number

The `<input type="number">` defines a **numeric** input field.

- Input Type Range

The `<input type="range">` defines a control for entering a number whose exact value is not important (like a slider control). Default range is 0 to 100.

- Input Type Search

The `<input type="search">` is used for search fields (a search field behaves like a regular text field)

- Input Type Tel

The `<input type="tel">` is used for input fields that should contain a telephone number.

- Input Type Time

The `<input type="time">` allows the user to select a time (no time zone).

- Input Type Url

The `<input type="url">` is used for input fields that should contain a URL address.

- Input Type Week

The `<input type="week">` allows the user to select a week and year.

HTML Input Attributes

- The value Attribute

The input value attribute specifies an initial value for an input field

- The readonly Attribute

The input readonly attribute specifies that an input field is read-only

- The disabled Attribute

The input disabled attribute specifies that an input field should be disabled.

A disabled input field is unusable and un-clickable

- The size Attribute

The input size attribute specifies the visible width, in characters, of an input field.

The default value for size is 20.

- The maxlength Attribute

The input maxlength attribute specifies the maximum number of characters allowed in an input field.

- The min and max Attributes

The input min and max attributes specify the minimum and maximum values for an input field.

The min and max attributes work with the following input types: number, range, date, datetime-local, month, time and week.

- The multiple Attribute

The input multiple attribute specifies that the user is allowed to enter more than one value in an input field.

The multiple attribute works with the following input types: email, and file.

- The pattern Attribute

The input pattern attribute specifies a regular expression that the input field's value is checked against, when the form is submitted.

The pattern attribute works with the following input types: text, date, search, url, tel, email, and password.

- The placeholder Attribute

The input placeholder attribute specifies a short hint that describes the expected value of an input field (a sample value or a short description of the expected format).

The placeholder attribute works with the following input types: text, search, url, tel, email, and password.

- The required Attribute

The input required attribute specifies that an input field must be filled out before submitting the form.

The required attribute works with the following input types: text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file.

- The step Attribute

The input step attribute specifies the legal number intervals for an input field.

The step attribute works with the following input types: number, range, date, datetime-local, month, time and week.

- The autofocus Attribute

The input autofocus attribute specifies that an input field should automatically get focus when the page loads.

- The height and width Attributes

The input height and width attributes specify the height and width of an `<input type="image">` element.

- The list Attribute

The input list attribute refers to a `<datalist>` element that contains pre-defined options for an `<input>` element.

- The autocomplete Attribute

The input autocomplete attribute specifies whether a form or an input field should have autocomplete on or off.

the autocomplete attribute works with `<form>` and the following `<input>` types: text, search, url, tel, email, password, datepickers, range, and color.

How to Create a Form in Html

We have to create the login and registration form in the Html document many times to show forms on a web page. So, this page will describe how to create both the forms in the Html documents using various tags.

Login Form

If we want to create a login form in the Html document to show the form on the web page then we have to follow the steps or use the following tags. Using these tags or steps, we can easily create a form.

Step 1: Firstly, we have to type the Html code in any text editor or open the existing Html file in the text editor in which we want to create the login form

```
<!Doctype Html>  
<Html>  
<Head>  
<Title>  
Create a Login form  
</Title>  
</Head>  
<Body>  
  
</Body>  
</Html>
```

Step 2: Now, we have to place the cursor at that point where we want to create a form between the starting and closing of <body> tag in the Html document. And, then we have to

type the tag whose name is `<form>`. It is the first tag which is used for creating an Html form.

`<Body>`

The following tags are used in this Html code for creating the Login form:

`<form>`

`</form>`

`</Body>`

Step 3: Now, we have to use the `<label>` tag, which defines the name for elements. So, we have to type the second tag for displaying the **User Id**. After typing the name, we have to close the `</label>` tag.

`<label>`User Id: **`</label>`**

Step 4: Now, we have to use the **`<input>`** tag, which allows users to insert the characters into the box. So, we have to type this tag with its attribute for inserting the User-id. There is no need to close the input tag because it is unpair tag.

`<input type="text">`

Step 5: Again, we have to type the **`<label>`** tag for displaying the label as password. And, then we have to type the password by using the `<input>` tag which is shown in the following block:

```
<label>Password :</label>  
<input type="password">
```

Step 6: And after then, we have to give the submit value in type attribute for submitting the form.

➤

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

Step 6: And, at last, we have to save the Html file and then run the file and then we will see the login form on the web page in the browser.

```
<html>  
<Body>  
<form>  
    <label>User Id: </label> <br>  
    <input type="text"> <br> <br>  
    <label>Password:</label> <br>  
    <input type="password"> <br> <br>  
    <input type="submit" value="Submit">  
</form>  
</Body>  
</Html>
```

Output :

The following tags are used in this Html code for creating the Login form:

User Id:

Password :

Submit

❖ HTML frames:

These are used to divide your browser window into multiple sections where each section can load a separate HTML document. A collection of frames in the browser window is known as a frameset. The window is divided into frames in a similar way the tables Are organized: into rows and columns.

To use frames on a page we use <frameset> tag instead of <body> tag. The <frameset> tag defines, how to divide the window into frames. The rows attribute of <frameset> tag defines horizontal frames and cols attribute defines vertical frames. Each frame is indicated by <frame> tag and it defines which HTML document shall open into the frame.

Example :

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Page Title</title>
```

```
</head>
```

```
<body>
```

```
<iframe src="sample1.html" height="400" width="400" frameb  
order="1">
```

```
<h1>This is a Heading</h1>
```

```
<p>This is a paragraph.</p>
```

```
</iframe>
```

```
</body>
```

```
</html>
```

Output :

Sample 1

This is a Heading

This is a paragraph