SUBJECT: WEB TECHNOLOGY
Subject Code : 2160708
6th Semester Computer Engineering
Faculty: Chhaya Patel, Jaydeep Tadhani

Unit 3: HTML

References:

Book: Developing Web Applications, Ralph Moseley and M. T. Savaliya, Wiley-India

Web link: https://www.tutorialspoint.com/html/

Web Link: https://www.w3schools.com/html/html_intro.asp

➢ HTML

- HTML is the standard markup language for creating Web pages.
- HTML describes the structure of Web pages using markup.
- HTML elements are the building blocks of HTML pages.
- HTML elements are represented by tags.
- HTML tags label pieces of content such as "heading", "paragraph", "table", and so on.
- Browsers do not display the HTML tags, but use them to render the content of the page.

➢ HTML Elements

- Elements consists of basic three parts:
  - Opening tag
  - Element's content
  - Closing tag
- **Example**

  <p> //opening paragraph tag

This is a paragraph. //element's content

</p> //closing tag

- Every web page required four critical elements:
  - **<html> element </html>**
    1. The <html> element is the root element of an HTML page.
    2. The <html> element defines the whole document.
    3. It has a start tag <html> and an end tag </html>.
  - **<head> element</head>**
    1. The <head> element contains Meta information about the document.
    2. It is placed between the <html> tag and the <body> tag.
    3. HTML metadata is data about the HTML document. Metadata is not displayed.
    4. Metadata typically define the document title, character set, styles, links, scripts, and other Meta information.
    5. The following tags describe metadata: <title>, <style>, <meta>, <link>, <script>, and <base>.
  - **<title> element</title>**
    1. The <title> element defines the title of the document, and is required in all HTML/XHTML documents.
    2. It defines a title in the browser tab.
    3. It provides a title for the page when it is added to favorites.
    4. It displays a title for the page in search engine results
  - **<body> element</body>**
    1. The <body> element defines the document body.
    2. It has a start tag <body> and an end tag </body>.
  - **<!DOCTYPE>**
    1. The <!DOCTYPE> declaration must be the very first thing in your HTML document, before the <html> tag.
    2. 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.

- **Example**

```
<!DOCTYPE html>
```

```
<html>
<head>
<title>Page Title</title>
</head>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

➢ HTML Tags

- HTML tags are element names surrounded by angle brackets.
  - Example : <tagname>content goes here...</tagname>
- 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 end tag is written like the start tag, but with a forward slash inserted before the tag name.

➢ Web Browsers

- The purpose of a web browser (Chrome, IE, Firefox, Safari) is to read HTML documents and display them.
- The browser does not display the HTML tags, but uses them to determine how to display the document.

➢ HTML Editors

- Web pages can be created and modified by using professional HTML editors.
- However, for learning HTML we recommend a simple text editor like Notepad (PC) or TextEdit (Mac).
- Follow the four steps below to create your first web page with Notepad.
  - Open Notepad++ /sublime Text 3
  - Write Some HTML

```
<!doctype html>
<html>
```

```
<body>
This is my first HTML code
</body>
</html>
```

- Save html file

  Name the file **"first.html"** and set the encoding to **UTF-8** (which is the preferred encoding for HTML files).
- View html page in your browser
- Open the saved HTML file in your favorite browser (double clicks on the file, or right-click - and choose "Open with").

HTML Tags or HTML formatting elements

➢ HTML paragraph tag

- The <p> tag defines a paragraph.
- Browsers automatically add some space (margin) before and after each <p> element. The margins can be modified with CSS (with the margin properties).
- We can specify text alignment within a paragraph using align attribute.
- Value of alignisleft orrightorcenter.
- Example:

```
<p>This is some text in a paragraph.</p>
Or
<p>This is some text in a paragraph
```

- The closing tag is considered optional.

➢ HTML heading 1:6

- Headings are defined with the <h1> to <h6> tags.
- <h1> defines the most important heading. <h6> defines the least important heading.
- Text displays bold and size of text depends on the tag.
- Each HTML heading has a default size. However, you can specify the size for any heading with the style attribute.
- Example:

```
<h1 style="font-weight: bold;font-size: 32px;font-family: times new roman; font-style:
italic">Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
```

- Browsers automatically add some white space (a margin) before and after a heading.
- Each HTML heading has a default size. However, you can specify the size for any heading with the style attribute.

```
<h1 style="font-size:60px;">Heading 1</h1>
```

- The closing tag is considered optional.

➢ HTML Text formatting elements

- HTML defines special elements for defining text with a special meaning.
- HTML uses elements like <b> and <i> for formatting output, like bold or italic text.
- Formatting elements were designed to display special types of text:
    1. <b> - Bold text:The <b> element marks text to be displayed as bold without any extra importance.
    2. -Strongtext- The HTML element defines strong text, with added semantic "strong" importance.
    3. <i> - Italic text : The <i> elements marks text to be displayed as italic.
    4. - Emphasized text : The HTML element defines emphasized text, with added semantic importance.
    5. <mark> - Marked text : The HTML <mark> element defines marked or highlighted text
    6. <small> - Small text : The HTML <small> element defines smaller text
    7. <del> - Deleted text: The HTML <del> element defines ~~deleted~~ (removed) text.
    8. <ins> - Inserted text : The HTML <ins> element defines inserted (added) text.

9. <sub> - Subscript text : The HTML <sub> element defines subscripted text : The HTML <sup> element defines superscripted text.
10. <sup> - Superscript text : The HTML <sup> element defines superscripted text.

➢ HTML comments

- Comment tags are used to insert comments in the HTML source code.
- You can add comments to your HTML source by using the following syntax:

```
<! -- Write your comments here -->
```

- There is an exclamation point (!) in the opening tag, but not in the closing tag.
- Comments are not displayed by the browser, but they can help document your HTML source code.

```
<! -- This is a comment -->
<p> This is a paragraph. </p>
<! -- Remember to add more information here -->
```

➢ HTML colors

- Colors are very important to give a good look and feel to your website.
- You can specify colors on page level using <body> tag or you can set colors for individual tags using bgcolor attribute.
- The <body> tag has following attributes which can be used to set different colors −
  - bgcolor − sets a color for the background of the page.
  - text − sets a color for the body text.
  - alink − sets a color for active links or selected links.
  - link − sets a color for linked text.
  - vlink − sets a color for visited links − that is, for linked text that you have already clicked on.
- There are following three different methods to set colors in your web page −
  - Color names − you can specify color names directly like green, blue or red.
    1. You can specify direct a color name to set text or background color.

```
<body text = "blue" bgcolor = "green">
```

  - Hex codes − A six-digit code representing the amount of red, green, and blue that makes up the color.

1. A hexadecimal is a 6 digit representation of a color.
2. The first two digits(RR) represent a red value, the next two are a green value(GG), and the last are the blue value(BB).
3. Each hexadecimal code will be preceded by a pound or hash sign #.

```
<body text = "#0000FF" bgcolor = "#00FF00">
 <body text = "#AA00FF" bgcolor = "#AAFF00">
```

- RGB values − This value is specified using the rgb( ) property.
    1. This property takes three values, one each for red, green, and blue.
    2. The value can be an integer between 0 and 255 or a percentage.

```
<body text = "rgb(0,0,255)" bgcolor = "rgb(0,255,0)">
```

    3. **Note** − All the browsers does not support rgb() property of color so it is recommended not to use it.
    4. **Example**

```
<!doctype html>
<html>
<body text="blue" bgcolor="white">
<h1 style ="color:green; border: 2px solid Tomato; background-
color:black"> Color testing </h1>
<h1 style="border: solid black;background-color: gray"> Hello
World</h1>
</body>
</html>
```

➢ HTML Links - hyperlinks

- HTML links are hyperlinks.
- You can click on a link and jump to another document.
- A link is specified using HTML tag <a>. This tag is called anchor tag and anything between the opening <a> tag and the closing tag becomes part of the link and a user can click that part to reach to the linked document.
- Following is the simple syntax to use <a> tag.

```
<a href = "Document URL" ... attributes-list>Link Text</a>
```

- **The target Attribute**
- Target attribute is used to specify the location **where linked document is opened**. Following are the possible options –
    - **_blank**
        1. Opens the linked document in a new window or tab.
    - **_self**
        1. Opens the linked document in the same frame (default).
    - **_parent**
        1. Opens the linked document in the parent frame.
    - **_top**
        1. Opens the linked document in the full body of the window.

    Example

```
<!doctype html>
<html>
<head>
       <title>Hyperlink Example</title>
       <base href="https://www.tutorialspoint.com/">
</head>
<body>
<p>Click any of following links</p>
<a href="/html/index.htm"   target="_blank">Open in new
window</a><br/>
<a href="/html/index.htm"   target="_self">Open in self</a><br/>
<a href="/html/index.htm"   target="_parent">Open in
parent</a><br/>
<a href="/html/index.htm"   target="_top">Open in top</a><br/>
</body>
</html>
```

- Hypertext references can be internal, local or global
    - **Internal:** links to anchors on the current page.

```
<h2 id="book"></h2>
```

```
<a href = "#book">Click here to get books</a>
```

- **Local:** Links to the other pages within your domain.

```
<a href = "../pics/image.jpg">Click here to get books</a>
```

- **Global:** Links to other domains outside of your website.

```
<a href = "https://google.com">Google.com</a>
```

➢ HTML Lists

- List is nothing but the collections of items or elements.
- ** element** is used to represent an item in a list.
- There are three types of lists.
  - **Unordered lists**
    1. An unordered list is a collection of related items that have no special order or sequence.
    2. This list is created by using HTML **<ul>** tag.
    3. **type** attribute for <ul> tag is used to specify the type of bullet you like. By default, it is a disc(bullet).
    4. Following are the possible options −
       - <ul type = "square">
       - <ul type = "disc">
       - <ul type = "circle">
       - <ul type = "none">
    5. **Example**

```
<html>
<head><title>Use of Unorder list</title></head>
<body>
<h1>All about Computer Software</h1>
Following are some popular operationg systems
      <ul type="square">
            <li>DOS</li>
            <li>Window 98</li>
            <li>Window XP</li>
```

```
            <li>Window Vista</li>
            <li>Ubuntu</li>
            <li>MAC OS</li>
        </ul>
</body>
</html>
```

- **Ordered list**
    1. An ordered list starts with the **<ol>** tag.
    2. Each list item starts with the **<li>** tag.
    3. The numbering starts at one and is incremented by one for each successive ordered list element tagged with <li>.
    4. **type** attribute for <ol> tag to specify the type of numbering you like. By default, it is a number.
    5. Following are the possible options –
        - <ol type = "1"> - Default-Case Numerals.
        - <ol type = "I"> - Upper-Case Numerals.
        - <ol type = "i"> - Lower-Case Numerals.
        - <ol type = "A"> - Upper-Case Letters.
        - <ol type = "a"> - Lower-Case Letters.
    6. **Example:**

```
<!DOCTYPE html>
<html>
<head>
        <title>Ordered list</title>
</head>
<body>
        <h2>Following are some core subjects of computer
science</h2>
        <ol type="a">
            <li>Computer Nerwork</li>
            <li>Operating System</li>
            <li>Database management System</li>
            <li>Web programming</li>
            <li>Software Engineering</li>
        </ol>
```

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

7. **start Attribute**
8. You can use start attribute for <ol> tag to specify the starting point of numbering you need.
9. Following are the possible options −
10. <ol type = "1" start = "4"> - Numerals starts with 4.
11. <ol type = "I" start = "4"> - Numerals starts with IV.
12. <ol type = "i" start = "4"> - Numerals starts with iv.
13. <ol type = "a" start = "4"> - Letters starts with d.
14. <ol type = "A" start = "4"> - Letters starts with D.

- **Definition list / Description List**
  1. HTML and XHTML supports a list style which is called definition lists where entries are listed like in a dictionary or encyclopedia.
  2. The definition list is the ideal way to present a glossary, list of terms, or other name/value list.
  3. Definition List makes use of following three tags.
     - <dl> − Defines the start of the list
     - <dt> − A term
     - <dd> − Term definition
     - </dl> − Defines the end of the list
  4. **Example**

```
<!DOCTYPE html>
<html>
<head>
	<title>HTML Defination List</title>
</head>
<body>
	<dl>
		<dt>HTML</dt>
		<dd>Hyper Text Markup language</dd>
		<dt>HTTP</dt>
```

```
            <dd>Hyper Text Trasfer Protocol</dd>
        </dl>

</body>
</html>
```

➢ HTML Tables

- The HTML tables allow web authors to arrange data like text, images, links, other tables, etc. into rows and columns of cells.
- The HTML tables are created using the **<table>** tag in which the **<tr>** tag is used to create table rows and **<td>** tag is used to create data cells. The elements under **<td>** are regular and left aligned by default.
- **Example**

```
<!DOCTYPE html>
<html>
<head>
      <title>HTML Table</title>
</head>
<body>
<table border="1" cellpadding = "5" cellspacing = "5"
bordercolor="green"    bgcolor="yellow" width="400" height="150">
      <caption>This is HTML Table</caption>
            <tr>
                  <th>Name</th>
                  <th>Roll Number</th>
                  <th>Semenster</th>
            </tr>
            <tr>
                  <td>Riya</td>
                  <td>19 Years</td>
                  <td>6</td>
            </tr>
            <tr>
                  <td>Diya</td>
                  <td>20 Years</td>
                  <td>8</td>
            </tr>
```

```
                <tr>
                        <td>Priya</td>
                        <td>20 Years</td>
                        <td>8</td>
                </tr>
        </table>
</body>
</html>
```

- **Table Border**
  - The **border** is an attribute of <table> tag and it is used to put a border across all the cells. If you do not need a border, then you can use border = "0".

- **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.
  - Headings, which are defined in <th> tag are centered and bold by default.

- **Cellpadding and Cellspacing Attributes**
  - There are two attributes called cellpadding and cellspacing which you will use to adjust the white space in your table cells.
  - The cellspacing attribute defines space between table cells, while cellpadding represents the distance between cell borders and the content within a cell.

- **Colspan and Rowspan Attributes**
  - You will use **colspan** attribute if you want to merge two or more columns into a single column. Similar way you will use **rowspan** if you want to merge two or more rows.

```
<td rowspan = "2">Row 1 Cell 1</td>
<td colspan = "3">Row 3 Cell 1</td>
```

- **Tables Backgrounds**
  - You can 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.

- **Table Height and Width**
  - You can specify table width or height in terms of pixels or in terms of percentage of available screen area.

- **Table Caption**
  - The **caption** tag will serve as a title or explanation for the table and it shows up at the top of the table.

➢ HTML Images

- Images can improve the design and the appearance of a web page.
- **Insert Image**
  - You can insert any image in your web page by using **<img>** tag.
  - The <img> tag is an empty tag, which means that, it can contain only list of attributes and it has no closing tag.
  - Following is the simple syntax to use this tag.

```
<img src = "Image URL" ... attributes-list/>
```

- **Example**

```
<!DOCTYPE html>
<html>
<head>
      <title>Image in Web page</title>
</head>
<body>
<p>Simple Image insert</p>
<a href="https://www.w3schools.com/html/html_images.asp">
<img src="image.jpg" alt="Image not found" width="500"
height="500" border="3">
</a>
</body>
```

```
</html>
```

- **alt attribute**
  - If a browser cannot find an image, it will display the value of the alt attribute.

- **Image Size - Width and Height**
  - You can specify width and height of the image in terms of either pixels or percentage of its actual size.

- **Image Border**
  - By default, image will have a border around it, you can specify border thickness in terms of pixels using border attribute.
  - A thickness of 0 means, no border around the picture.
- **Image Alignment**
  - By default, image will align at the left side of the page, but you can use **align**attribute to set it in the center or right.

- **Image as a Link**
  - To use an image as a link, put the <img> tag inside the <a> tag.

➢ HTML Forms

- HTML Forms are required, when you want to collect some data from the site visitor.
- For example, during user registration you would like to collect information such as name, email address, phone number, 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, textarea fields, drop-down menus, radio buttons, checkboxes, etc.
- **Example:**

```
<form>
……
```

```
form elements
…….
</form>
```

- Following is a list of the most frequently used form attributes –
    - **action**
        1. Backend script ready to process your passed data.
    - **method**
        1. Method to be used to upload data. The most frequently used are GET and POST methods.
    - **Target**
        1. Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.

- **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

- **Text Input Controls**
    - There are three types of text input used on forms –
        1. **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**.
            - **Example**

```
<!DOCTYPE html>
<html>
<head>
	<title>Single line 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>
```

- **Attributes**
- **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 specifying the maximum number of characters a user can enter into the text box.

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**

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

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

- **Attributes**
- **type:** Indicates the type of input control and for text 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 specifying the maximum number of characters a user can enter into the text box.

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**

```
<!DOCTYPE html>
<html>
<head>
        <title>Multiple Line Input</title>
</head>
```

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

- **Attributes**
- **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.

- **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**

```
<!DOCTYPE html>
<html>
<head>
   <title>Check Box control</title>
</head>
<body>
<form>
   <input type="checkbox" name="maths" >Maths
   <input type="checkbox" name="physics" checked="on">Physics
</form>
</body>
</html>
```

  - **Attributes**

- **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.

- **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.**
  - **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>
```

  - **Attributes**
  - **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.

- **Select Box Control / Menus**

- 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.
- **Example**

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

- **Attributes**
- **name:** Used to give a name to the control which is sent to the server to be recognized and get the value.
- **size:**This can be used to present a scrolling list box.
- **multiple:**If set to "multiple" then allows a user to select multiple items from the menu.

- **File Upload box**
  - 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**

```
<!DOCTYPE html>
<html>
<head>
   <title>File Upload box</title>
```

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

- **Attributes**
- **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.

- **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 –
    1. **submit**
       - This creates a button that automatically submits a form.
    2. **reset**
       - This creates a button that automatically resets form controls to their initial values.
    3. **button**
       - This creates a button that is used to trigger a client-side script when the user clicks that button.
    4. **image**
       - This creates a clickable button but we can use an image as background of the button.
  - **Example**

```
<!DOCTYPE html>
```

```
<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="button" value="OK">
   <input type="image" name="imagebutton" src="image.jpg"/>
</form>
</body>
</html>
```

- **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 will be displayed next based on the passed current page.
  - **Example**

```
<html>
<head>
<title>File Upload Box</title>
</head>

<body>
<form>
<p>This is page 10</p>
<input type = "hidden" name = "pagename" value = "10" />
<input type = "submit" name = "submit" value = "Submit" />
<input type = "reset" name = "reset"  value = "Reset" />
</form>
</body>
```

```
</html>
```

➢ XHTML

- XHTML stands for EXtensible HyperText Markup Language
- XHTML is almost identical to HTML.
- XHTML is stricter than HTML.
- XHTML is supported by all major browsers.
- XHTML is a family of XML languages which extend or mirror versions of HTML.
- It does not allow omission of any tags or use of attribute minimization.
- XHTML requires that there be an end tag to every start tag and all nested tags must be closed in the right order.
- For example, while <br> is valid in HTML, it would be required to write
in XHTML.

➢ Difference between HTML and XHTML

No	HTML	XHTML
1	HTML is Hypertext Markup Language.	XHTML is Extensible Hypertext Markup Language.
2	An application of SGML (Standard Generalized Markup Language).	An application of XML(Extensible Markup Language)
3	Can have empty/open tags e.g. , .	All the unclosed tags must be closed e.g. , …… .
4	No hard rule on structures of the elementse.g. The difference .	Structure of the elements should be followed e.g. The difference .
5	Attributes have quotes as optional e.g. <font color=#ff0000>.	Attributes have quotes mandatory e.g. <font color=”#ff0000?”>.
6	Attributes values not significant e.g. <input type=”checkbox” checked>.	Attributes values are important e.g. <input type=”checkbox” checked = “checked”>.
7	Case insensitive:The tags and attributes	Case sensitive:The tags and attributes

	can be of uppercase or lowercase as per the preference.	must be of lowercase.
8	All the content can be put under body element.	All the content has to be put in blocks (p, under body element).
9	HTML is not mandatory for one root element.	XHTML documents must have one root element.

➢ Meta Tags

- Metadata is data (information) about data or in this context we can say meta data is machine understandable information about web resources.
- The <meta> tag provides metadata about the HTML document. Metadata will not be displayed on the page, but will be machine parsable.
- Meta elements are typically used to specify page description, keywords, author of the document, last modified and other metadata.
- The metadata can be used by browsers (how to display content or reload page), search engines (keywords), or other web services.
- This tag is an empty element and so does not have a closing tag but it carries information within its attributes.
- Meta tag (Meta data) is included in head section of your web page.
- **Example**

```
<!DOCTYPE html>
<html>
<head>
   <title>Meta Tag Example</title>
   <meta charset="utf-8" name="keywords" content="HTML,Meta Tags"
   Metadeta" />
   <meta charset="utf-8" name="description" content="Learning about
   Meta Tags./">
   <meta  name="revised" content="Web Technology, 10/11/2017">
   <meta charset="utf-8" http-equiv="refresh"
   content="5;https://www.tutorialspoint.com/html/html_meta_tags.htm"
   >
   <meta  name="author" content="Chhaya Patel"/>
</head>
```

```
<body>
<p>Hello HTML 5</p>
</body>
</html>
```

- **Attributes**
  - **name:** Name for the property. Can be anything. Examples include, keywords, description, author, revised, generator etc.
  - **content:** Specifies the property's value.
  - **scheme:** Specifies a scheme to interpret the property's value (as declared in the content attribute).
  - **http-equiv:** Used for http response message headers. For example, http-equiv can be used to refresh the page or to set a cookie. Values include **content-type, expires, refresh and set-cookie.**

- **Use of Meta Tag**
  - **Specifying Keywords :** You can use <meta> tag to specify important keywords related to the document and later these keywords are used by the search engines while indexing your webpage for searching purpose.

```
<head>
<title>Meta Tags Example</title>
<meta name = "keywords" content = "HTML, Meta Tags, Metadata" />
</head>
```

  - **Document Description:** You can use <meta> tag to give a short description about the document. This again can be used by various search engines while indexing your webpage for searching purpose.

```
<head>
<title>Meta Tags Example</title>
<meta name = "keywords" content = "HTML, Meta Tags, Metadata" />
<meta name = "description" content = "Learning about Meta Tags." />
</head>
```

  - **Document Revision Date:**You can use <meta> tag to give information about when last time the document was updated. This information can be used by various web browsers while refreshing your webpage.

```
<head>
```

```
<title>Meta Tags Example</title>
<meta name = "keywords" content = "HTML, Meta Tags, Metadata" />
<meta name = "description" content = "Learning about Meta Tags." />
<meta name = "revised" content = "Tutorialspoint, 3/7/2014" />
</head>
```

- **Document Refreshing:** A <meta> tag can be used to specify a duration after which your web page will keep refreshing automatically**.**

```
<head>
<title>Meta Tags Example</title>
<meta name = "keywords" content = "HTML, Meta Tags, Metadata" />
<meta name = "description" content = "Learning about Meta Tags." />
<meta name = "revised" content = "Tutorialspoint, 3/7/2014" />
<meta http-equiv = "refresh" content = "5" />
</head>
```

- **Page Redirection:** You can use <meta> tag to redirect your page to any other webpage. You can also specify a duration if you want to redirect the page after a certain number of seconds.

```
<head>
<title>Meta Tags Example</title>
<meta name = "keywords" content = "HTML, Meta Tags, Metadata" />
<meta name = "description" content = "Learning about Meta Tags." />
<meta name = "revised" content = "Tutorialspoint, 3/7/2014" />
<meta http-equiv = "refresh" content = "5; url = http://www.tutorialspoint.com" />
</head>
```

- **Setting Cookies:** Cookies are data, stored in small text files on your computer and it is exchanged between web browser and web server to keep track of various information based on your web application need.You can use <meta>tag to store cookies on client side and later this information can be used by the Web Server to track a site visitor.

```
‹head>
<title>Meta Tags Example</title>
<meta name = "keywords" content = "HTML, Meta Tags, Metadata" />
<meta name = "description" content = "Learning about Meta Tags." />
<meta name = "revised" content = "Tutorialspoint, 3/7/2014" />
<meta http-equiv = "cookie" content = "userid = xyz;
```

```
    expires = Wednesday, 08-Aug-15 23:59:59 GMT;" />
</head>
```

- **Setting Author Name:** You can set an author name in a web page using meta tag.

```
<meta name = "author" content = "Chhaya Patel" />
```

- **Specify Character Set:** You can use <meta> tag to specify character set used within the webpage.

  By default, Web servers and Web browsers use ISO-8859-1 (Latin1) encoding to process Web pages. Given example set UTF-8 encoding as character set.

```
<meta http-equiv = "Content-Type" content = "text/html; charset = UTF-8" />
```

➢ HTML -Character Entities

- Some characters are reserved in HTML and they have special meaning when used in HTML document. For example, you cannot use the greater than and less than signs or angle brackets within your HTML text because the browser will treat them differently and will try to draw a meaning related to HTML tag.
- HTML processors must support following five special characters listed in the table that follows.

Symbol	Description	Entity Name	Number Code
"	quotation mark	&quot;	&#34;
'	apostrophe	&apos;	&#39;
&	ampersand	&amp;	&#38;
<	less-than	&lt;	&#60;
>	greater-than	&gt;	&#62;

- Example : If you want to write as a code then you will have to write as follows −

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

```
<head>
<title>HTML Entities</title>
</head>
<body>
&lt;div id = &quot;character&quot;&gt;
</body>
</html>
```

➢ HTML - Frames and Frame Sets

- HTML frames 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.
- **Disadvantages of Frames**
    - There are few drawbacks with using frames, so it's never recommended to use frames in your WebPages.
    - Some smaller devices cannot cope with frames often because their screen is not big enough to be divided up.
    - Sometimes your page will be displayed differently on different computers due to different screen resolution.
    - The browser's back button might not work as the user hopes.
    - There are still few browsers that do not support frame technology.
- **Creating Frames**
- 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.
- **The <frame> tag deprecated in HTML5. Do not use this element.**
- **Example**

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

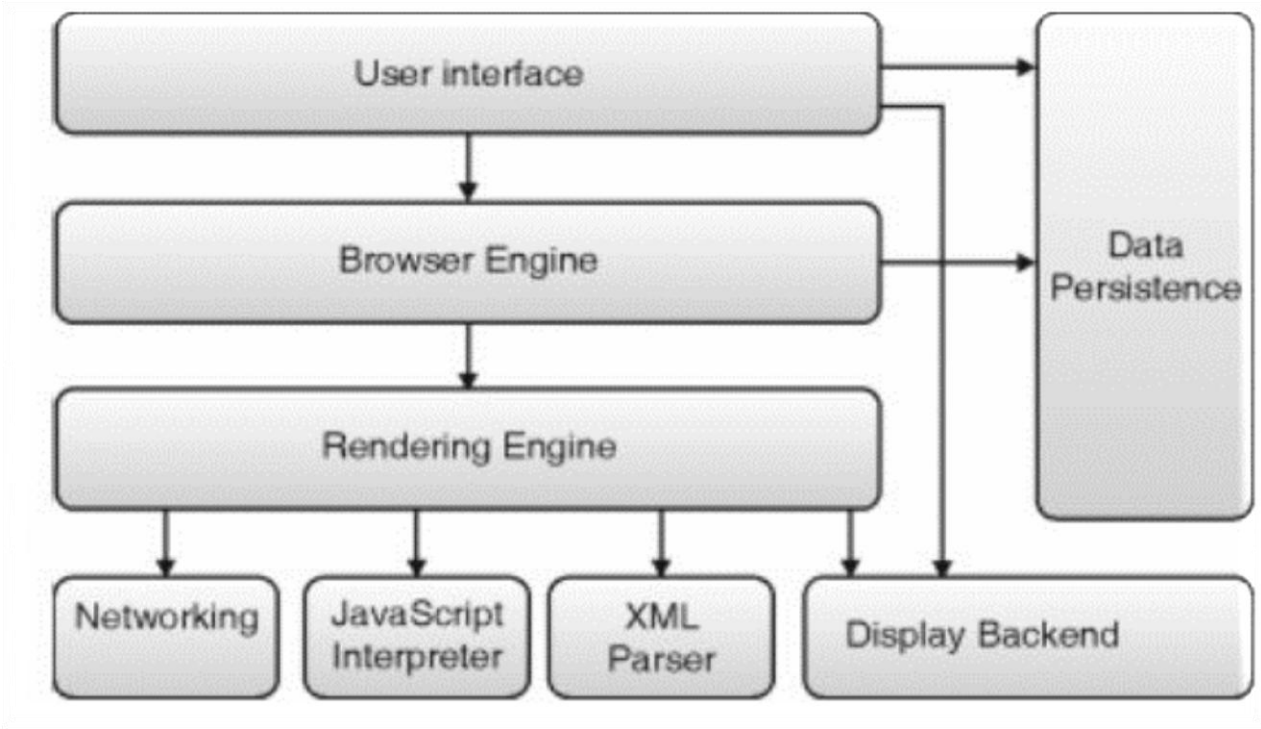
```
    <title>HTML Frames</title>
</head>
<frameset>
    <frame name="top"
    src="https://www.w3schools.com/html/html_intro.asp"/></frame>
    <frame name="bottom"
    src="https://www.tutorialspoint.com/html/html_frames.htm"></frame>
    <noframes>
          <body>
                Your browser does not support frames.
          </body>
    </noframes>
</frameset>
</html>
```

➢ Browser Architecture and web site structure

- Web browser is application software that allows us to view and explore information on the web. User can request for any web page by just entering a URL into address bar.
- Web browser can show text, audio, video, animation and more. It is the responsibility of a web browser to interpret text and commands contained in the web page.
- Following are the most common web browser available today:

Browser	Vendor
Internet Explorer	Microsoft
Google Chrome	Google
Mozilla Firefox	Mozilla
Netscape Navigator	Netscape Communications Corp.
Opera	Opera Software
Safari	Apple
Sea Monkey	Mozilla Foundation
K-meleon	K-meleon

- **Architecture**



The User Interface: This includes the address bar, back/forward button, bookmarking menu etc. Every part of the browser display except the main window where you see the requested page. It provides features such as toolbars, visual page-load progress, smart download handling, preferences, and printing.

The Browser Engine: Browser Engine is embeddable component that provides a high-level interface for querying and manipulating the rendering engine.

The Rendering Engine: It is responsible for displaying requested content. For example if the requested content is HTML, the rendering engine parses HTML and CSS, and displays the parsed content on the screen. By default the rendering engine can display HTML and XML documents and images. It can display other types of data via plug-ins or extension; for example, displaying PDF documents using a PDF viewer plug-in. Different browsers use different rendering engines: Internet Explorer uses Trident, Firefox uses Gecko, Safari uses WebKit. Chrome and Opera use Blink, a fork of WebKit.

Networking: It is used for network calls, like HTTP requests. It has platform independent interface and underneath implementations for each platform. It translates between different

character sets, and resolves MIME media types for files. It may implement a cache of recently retrieved resources.

The JavaScript Interpreter: It evaluates JavaScript code, which may be embedded in web pages. Certain Java-Script functionality, such as the opening of pop-up windows, may be disabled by the Browser Engine or Rendering Engine for security purposes.

The XML Parser: It parses XML documents into a Document Object Model (DOM) tree. almost all browser implementations leverage an existing XML Parser rather than creating their own from scratch.

The Display Backend: It provides drawing and windowing primitives, a set of user interface widgets, and a set of fonts. It exposes a generic interface that is not platform specific. Underneath it uses the operating system user interface methods.

The Data Storage: This is a persistence layer. The browser may need to save all sorts of data locally, such as cookies. It also stores high-level data such as bookmarks or toolbar settings. Browsers also support storage mechanisms such as localStorage, IndexedDB, WebSQL and FileSystem.

➢ Introduction of HTML 5

- HTML5 is the latest and most enhanced version of HTML supporting desktop like behavior for web applications.
- HTML 5 added many new elements and attributes to encourage the development of rich internet application.

- **New Semantic Elements:** These are like <header>, <footer>, and <section>.

- **Forms 2.0:** Improvements to HTML web forms where new attributes have been introduced for <input> tag. (like color, date, number, placeholder etc.)

- **Persistent Local Storage:** To achieve without resorting to third-party plug-in.

- **Web Socket:** A next-generation bidirectional communication technology for web applications.

- **Server-Sent Events:** HTML5 introduces events which flow from web server to the web browsers and they are called Server-Sent Events (SSE).

- **Canvas:** This supports a two-dimensional drawing surface that you can program with JavaScript.

- **Audio and Video:** You can embed audio or video on your web pages without resorting to third-party plug-ins.

- **Geolocation :** Now visitors can choose to share their physical location with your web application.

- **Micro data:** This lets you create your own vocabularies beyond HTML5 and extend your web pages with custom semantics.

- **Drag and drop:** Drag and drop the items from one location to another location on the same webpage.

****