SYLLABUS

HTML5: Fundamentals of HTML, Working with Text, Organizing Text in HTML, Working with Links and URLs, Creating Tables, Working with Images, Colors, and Canvas, Working with Forms.

Fundamentals of HTML

Structuring of HTML document

- Hyper Text Mark-up Language (HTML) is the standard mark-up language for creating Web pages.
- HTML describes the structure of a Web page.
- A HTML document is created by using elements and attributes and has the .html or .htm extension.

```
<! DOCTYPE     html>  -------specifies version of HTML i.e., HTML5
<html>  ------specifies where the html document begins
        <head>
                <title>… </title>
        </head>
        <body>
                …
        </body>
</html>
```

The basic structure having the following sections

1. **Elements and Attributes**
2. **Tags**
3. **The DOCTYPE element**

1. **Elements and Attributes:**
   - **Elements** are the building blocks of HTML document.
   - An element provides instruction to browser a web browser, specifies how to display the HTML document.
   - Elements are represented by tags and tags are uses the <, /, and > symbol.
   - Most of the elements having opening tag (starting tag) (<element-name>), and closing tag (ending tag) (</element-name>).
   - Syntax of HTML element

     <element-name attribute-name =”attribute-value” > content </element-name>
   - From above structure DOCTYPE, html, head, and body are the elements.
   - The document starts with DOCTYPE. The html element having two sub elements, head and body.
   - Head element having Meta data and body element having actual content of the HTML document.
   - **Attributes** are used to provides additional information about the properties and behavior of HTML elements.
   - Attributes are **name-values pairs** separated by the equal (=) sign.
   - Attribute values are enclosed within single quotes or double quotes.
   - Example **<img src=”nature.jpg” >**
   - Elements and attributes are **not case sensitive**.
   - Elements and attributes are **predefined**.
   - Each element consists of its own set of attributes.

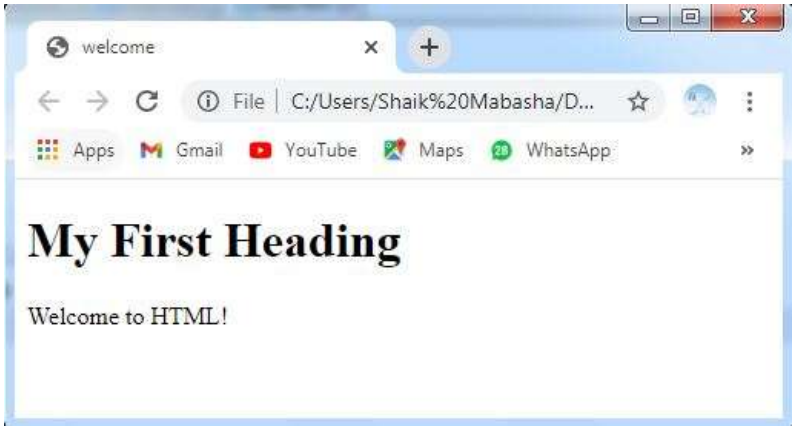
2. **Tag:**
   - As we discussed, Most of the elements having opening tag (starting tag) (<element-name>), and closing tag (ending tag) (</element-name>).
   - Opening tags are written within the < and > sign.
   - Closing tags are written within the <, >, and / sign.
   - Example of open tags <head>, <body> and closing tags </head>, </body>
   - Tags are two types
     i. **Container tag** having both open and closing tags. Eg <body> /<body>
     ii. **Empty tag** does not have closing tag. Eg <hr>,

3. **The DOCTYPE element:**
   - The DOCTYPE element provides the Document Type Definition declaration.
   - The declaration is for browsers to specify which version of HTML documenr.
   - Example <DOCTYPE html>, Here the version is HTML5

Example: Create a web page that shown a message "welcome to HTML!".

```
<!DOCTYPE      html>
<html>
        <head>
                <title>welcome</title>
        </head>
        <body>
                <h1>My First Heading</h1>
                 <p>Welcome to HTML!</p>
        </body>
</html>
```

Output:

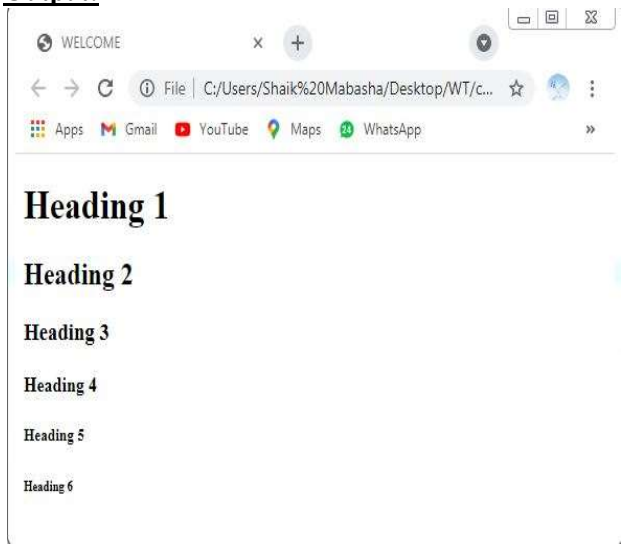


Working with **Heading Elements:**

- HTML headings are titles or subtitles on a web page.
- Heading elements is defined as <hn> tag where **n** range from 1 to 6.
- Syntax:<hn>CONTENT </hn>
- n →0 to 6
- **Example:** <h1>BAPATLA ENGINEERING COLLEGE</h1>
  <h2>list of branches</h2>
- **Example: demonstrate the heading tags**

```
<!DOCTYPE html>
<html>
    <head>
        <title>WELCOME</title>
    </head>
    <body>
        <h1>Heading 1</h1>
        <h2>Heading 2</h2>
        <h3>Heading 3</h3>
        <h4>Heading 4</h4>
        <h5>Heading 5</h5>
        <h6>Heading 6</h6>
    </body>
</html>
```

Output:



Working with **LINE BREAK:**

- It forcefully inserts the new line/single line break in a text.
- Syntax:
-
 tag is Self-closing or empty tag.
-
 tag is used to display the addresses or poems…

Working with **HORIZONTAL RULE:**

- It defines a thematic break in a html page(e.g. a shift of topic).it is used to separate the content in an html page.
- Syntax: <hr/>

Working with **COMMENTS:**

- Comment is used to add or insert the description about source code.
- Syntax: < !-- comments -->
- Comments can place anywhere in html documents and these are not displayed in the browser.

Character Entity:

It is a reference from SGML(Standard Generalized Markup Language) which is used to represent reserved characters in HTML.

Some characters have a special meaning in HTML like <, >, &, " , = , etc..

A character entity has three parts:

1. An ampersand (&)
2. An entity name or entity number.
3. Semicolon.

Syntax: , < , > , &

The most common character entities are as follows:

Character	Description	Entity Name	Entity number
<	Less than	<	<
>	Greater than	>	>
	Space		
&	Ampersand	&	&
‘	Apostrophe	&apos	'

Working with **PARAGRAPH Element:**

- Paragraph element specifies the content of an HTML as a paragraph.
- Paragraph always starts with new line and provides the description.
- Browsers are automatically adding the white spaces before and after the paragraphs.
- Paragraphs are represented by using tag.
- The browser will automatically remove any extra spaces and lines when the web page was displayed.
- SYNTAX: ** content **

Example: design a web that demonstrate the paragraph tag

```
<!DOCTYPE html>
<html>
    <head>
        <title>WELCOME</title>
    </head>
    <body>
        <h1>HTML</h1>
        <p>The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript.</p>
    </body>
</html>
```

Working with Text

- HTML provides a set of elements to change the appearance of the text by applying **formatting** features.
- The formatted text having both the starting and ending tags.
- Formatting text categorized into two ways: **Physical style** and **Logical style**.

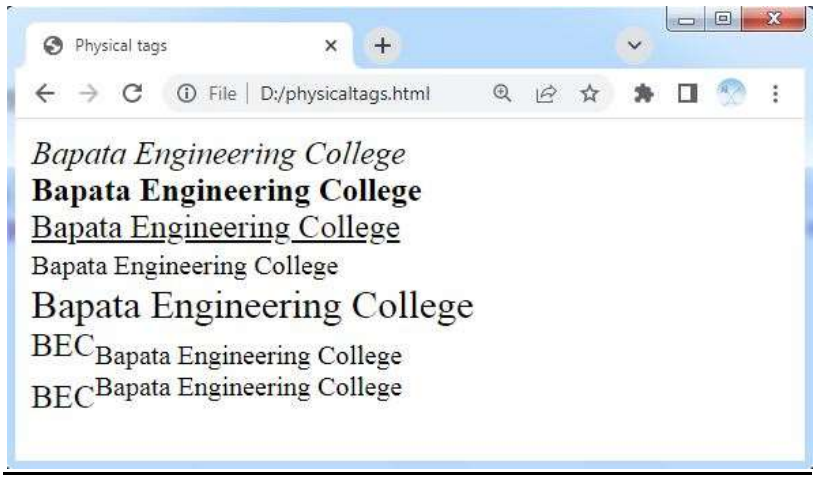
1. Physical style elements:

- Physical elements are used to change the appearance of the text specified in the tag.
- The Physical Formatted TAGS are:
- **<b>:** It is used to define BOLD text.
- **<i>** : It is used to display the text in ITALIC format.
- **<u>:** It is used to define UNDERLINED TEXT.
- **<small>:** It is used to display the text in small size.
- **<big>:** It is used to display the text in big size.
- **<sub>:** It is used to define text in subscript size .
- **<sup>:**It is used to define text 6in superscript size .it is display the text in powers.

Example: **design web page to demonstrate the physical formatted tags.**

```
<!DOCTYPE html>
<html>
    <head>
        <title>Physical tags</title>
    </head>
    <body >
        <i>Bapata Engineering College</i><br>
        <b>Bapata Engineering College</b><br>
        <u>Bapata Engineering College</u><br>
        <small>Bapata Engineering College</small><br>
        <big>Bapata Engineering College</big><br>
        BEC<sub>Bapata Engineering College</sub><br>
        BEC<sup>Bapata Engineering College</sup><br>
    </body>
</html>
```

Output:



2. Logical style elements:

- Logical style can specify meaning or uses of the text specifies within the logical formatted tags.
- The logical formatted tags are:
- **<abbr>** :It defines an abbreviation or an acronym of a text, like "HTML", "CSS", "BEC", etc.
- **<code>** :It is used to define a piece of html code in the browser.
- **<samp>:** It is used to define sample output from a computer program. The content inside is displayed in the browser's default mono space font.
- **<kdb>:** The text specified within the content inside a tag is displayed in the browser's default keyboard font.
- **:**Itis used to define emphasized text to make text as a important. The content inside is typically displayed in italic.
- **** :it is used to define text with strong importance. The content inside is typically displayed in bold.
- **<dfn>:**It stands for the "definition element", and it specifies a term that is going to be defined within the content.
- **<q>:**It specified text will be display in double codes.
- **<blockquote>:**Its specified text will be display strong double codes. It means browser can add spaces between before and ending of the text and also it display as a new paragraph.
- **<ins>:**It display the text as specified text as recently added to the existing line.
- **<del>:**It display the text as specified text as recently deleted to the strikeoff.
- **<mark>** :It is defines text that should be marked or highlighted with some colour. that colour can be change as per the requirement.
- **<bdo>**:It is By directional override. It is used to override the current text direction.

Example: **design web page to demonstrate the logical formatted tags.**

```
<!DOCTYPE html>
<html>
<head>
        <title> HTML Text Formatting </title>
</head>
<body>
      <abbr title="Hyper Text Markup Language"> HTML</abbr> is used to develop web pages
      <h4>Sample HTML code</h4>
      <code>
            &lt;!DOCTYPE html&gt;<br>
            &lt;html&gt;<br>
            &lt;head&gt;<br>
            &lt;title&gt;HTML Text Formatting&lt;/title&gt;<br>
            &lt;/head&gt;<br>
            &lt;body&gt;<br>
            &lt;h2&gt;Sample code&lt;/h2&gt;<br>
            &lt;/body&gt;<br>
            &lt;/html&gt;<br>
      </code><br><br>
      press <kbd>ENTER</kbd> to run the program<br>
      This is <em>emphasizing text</em><br>
      This is <strong>strong text</strong><br><br>
      <dfn title="Hyper Text Markup Language"> HTML</dfn> is used to develop web pages<br><br>
      There are two types of formatting tags <q> physical and logical</q><br><br>
```

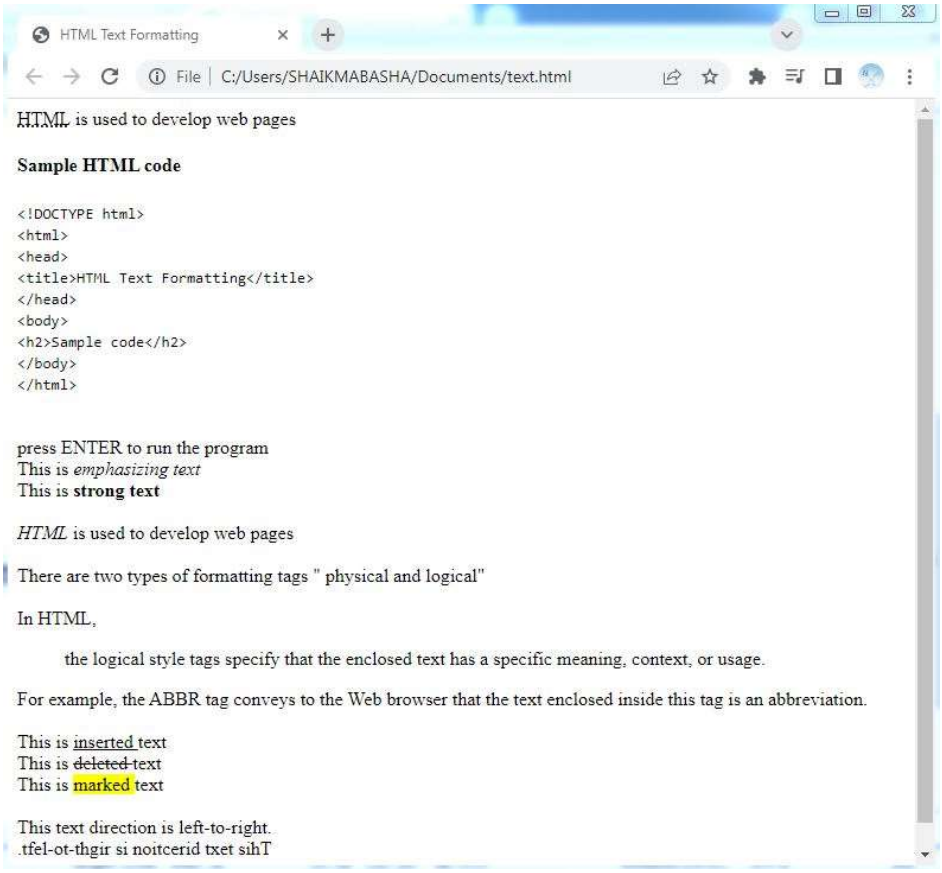
In HTML, <blockquote>the logical style tags specify that the enclosed text has a specific meaning, context, or usage.</blockquote> For example, the ABBR tag conveys to the Web browser that the text enclosed inside this tag is an abbreviation.

This is <ins> inserted </ins>text
This is <del>deleted </del> text
This is <mark>marked </mark> text

<bdo dir="ltr"> This text direction is left-to-right.</bdo>
<bdo dir="rtl"> This text direction is right-to-left.</bdo>

</body>
</html>

Output:



Attributes used with the Physical and Logical Style Elements.

Attributes	Description
Class	Indicate a class name for an element
Dir	Indicates the direction text, such as left-to-right or right-to-left
Id	Indicates a unique id for an element
Lang	Indicates language code for the content in an element
Style	Indicates an inline style for an element
Title	Specifies a title of an element

Organizing Text in HTML

- Organizing text means the proper placement of all the HTML elements and their content in a web page.
- By default, the web browser displays the all text of html page as single paragraph.
- HTML allows arranging the text into different formats and style, such as paragraphs, layers, and columns.
- The following tasks are involved in arranging the text of the web page.
  - Allowing **Word Break---- <wbr>ddhd</wbr> ---- HTML5**
  - Defining the **Preformatted Text--- <pre> ---HTML5**
  - Defining the **DIV Element**
  - Defining the **SPAN Element ------HTML5**
  - Formatting text in **table--- <table> <tr> <td> <th>**
  - Defining the **Ruby (caption) Text---- <ruby> <rp> <rt> ----HTML5**
  - **Lists**

Word Break:

- **WBR Element** to insert word break between words or paragraph in an HTML document.
- It is mostly used when the used word is too long and there are chances that the browser may break lines at the wrong place for fitting the text.
- **WBR Element** does not force the web browser to break the line.
- Where **
** element breaks the line forcefully.

Preformatted Text:

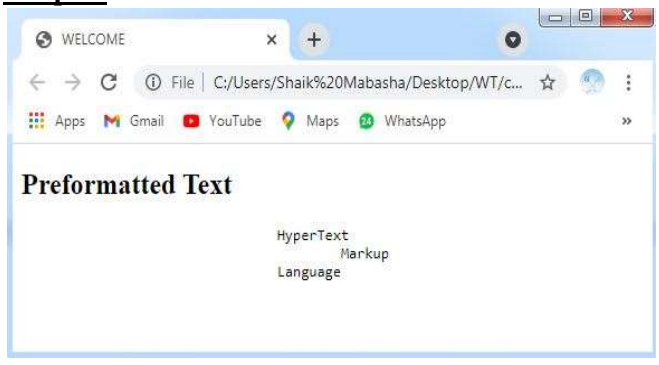
- PRE Element instructs the browser that text is a formatted text and should not be formatted again.
- Text in a <pre> element is displayed in a fixed-width font, and the text preserves both spaces and line breaks.
- The text will be displayed exactly as written in the HTML source code.
- Attributes of the PRE Element

Attribute	Description
Class	Indicates a class name for PRE Element
Dir	Indicates the direction of the enclosed text.
Id	Indicates a unique id for the element.
Lang	Indicates a base language used for the element.
Style	Indicates an inline style for element.
Title	Indicates extra information about element.

Example: design a web that demonstrate the PRE element

```
<!DOCTYPE html>
<html>
    <head>
        <title>WELCOME</title>
    </head>
    <body>
        <h2>Preformatted Text</h2>
        <PRE>
            HyperText
                Markup
            Language
```

Output:



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

DIV Element:

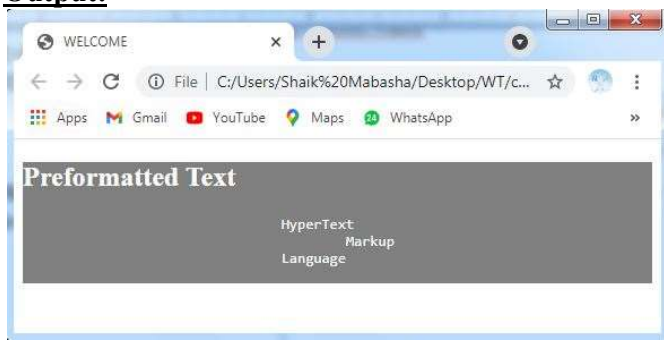
- DIV Element is used to divide the webpage into different division or section.
- DIV Element basically works as a container for other HTML Elements.
- DIV Element used to group the HTML element and apply the CSS on them.
- By default, browsers always place a line break before and after the element.
- Attributes of the Element

Attribute	Description
Class	Define a class name for an Element
Dir	Define the direction of the content in the element.
Id	Define a unique id for an element.
Lang	Define a base language code for the element.
Style	Define an inline style for element.
Title	Define extra information about element.

- Example:

```
<!DOCTYPE html>
<html>
    <head>
        <title>WELCOME</title>
    </head>
    <body>
        <div style=" background-color:gray;   color:white ">
        <h2>Preformatted Text</h2>
        <PRE>
        HyperText
                    Markup
            Language
        </PRE>
        </div>
    </body>
</html>
```

Output:



SPAN Element:

- It is used to change the style of text enclosed within the element using the style attribute.
- Span element provides additional formatting capabilities to HTML elements using **style attribute**.
- Attributes of the Element

Attribute	Description
Class	Define a class name for an Element
Dir	Define the direction of the content in the element.
Id	Define a unique id for an element.
Lang	Define a base language code for the element.
Style	Define an inline style for element.
Title	Define extra information about element.

RUBY (caption) element:

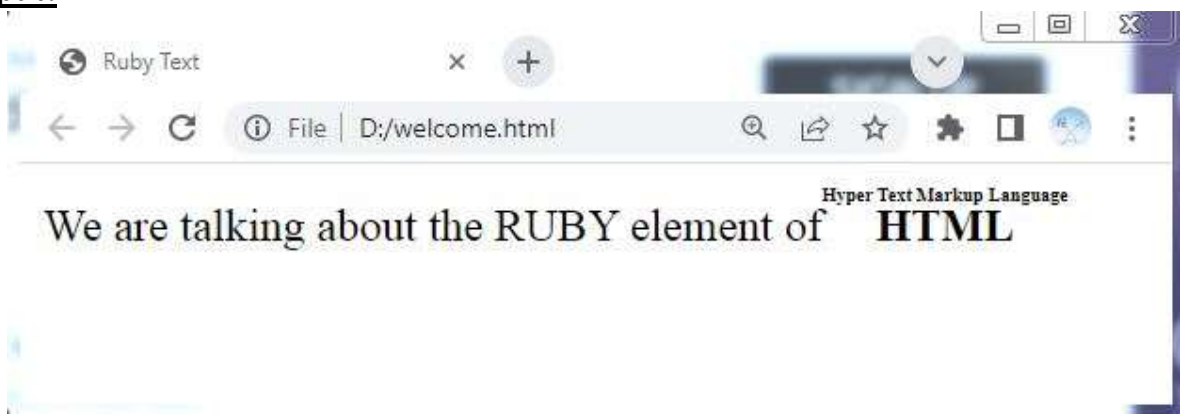
- RUBY Element is used to provide caption for your text in your webpage.
- A ruby annotation is a small extra text, attached to the main text to indicate the pronunciation or meaning of the corresponding characters.
- This kind of annotation is often used in Japanese publications.
- RUBY Element contains the following elements:
  RP – contains parenthesis.
  RT – Acts as a container for the ruby text.
- Attributes of the Element

Attribute	Description
Class	Indicates a class name for PRE Element
Dir	Indicates the direction of the enclosed text.
Id	Indicates a unique id for the element.
Lang	Indicates a base language used for the element.
Style	Indicates an inline style for element.
Title	Indicates extra information about element.
Cols	Indicates the number of columns used in the table.
Language	Indicates scripting language used for an element

- **Example**:

```
<!DOCTYPE HTML>
<HTML>
    <HEAD>
        <TITLE>Ruby Text</TITLE>
    </HEAD>
    <BODY>
        We are talking about the RUBY element of
        <RUBY><B>HTML</B><RP>(</RP><RT><B><SMALL>Hyper Text Markup
        Language</SMALL></B></RT><RP>)</RP></RUBY>
    </BODY>
</HTML>
```

Output:



Lists:

- List is used to display the information in the form list on web page.
- **LI Element** is used to specify the item.
- Information in the list is displayed in an **order or sequential** way or **un-order or random** ways.
- **There are three types of list.**

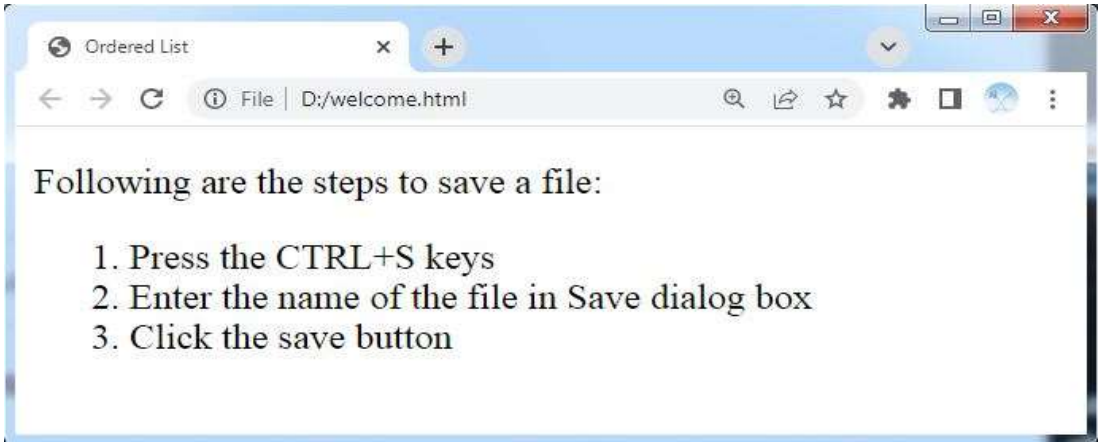
1.Ordered list or Numbered list -→ <OL> ---<LI>
2.Unordered list or **Bulleted List → <UL> ---<LI>**
3.Description or Definition List -→ <DL> --- <DT> <DD>

1. **Order List: OL Element** is used to display the list information in **order and sequential** manner.

Example:

```
<!DOCTYPE HTML>
<HTML>
<HEAD>
    <TITLE>
        Ordered List
    </TITLE>
    <BODY>
        <P>Following are the steps to save a file:</P>
        <OL>
            <LI>Press the CTRL+S keys
            <LI>Enter the name of the file in Save dialog box
            <LI>Click the save button
        </OL>
    </BODY>
</HTML>
```

Output:

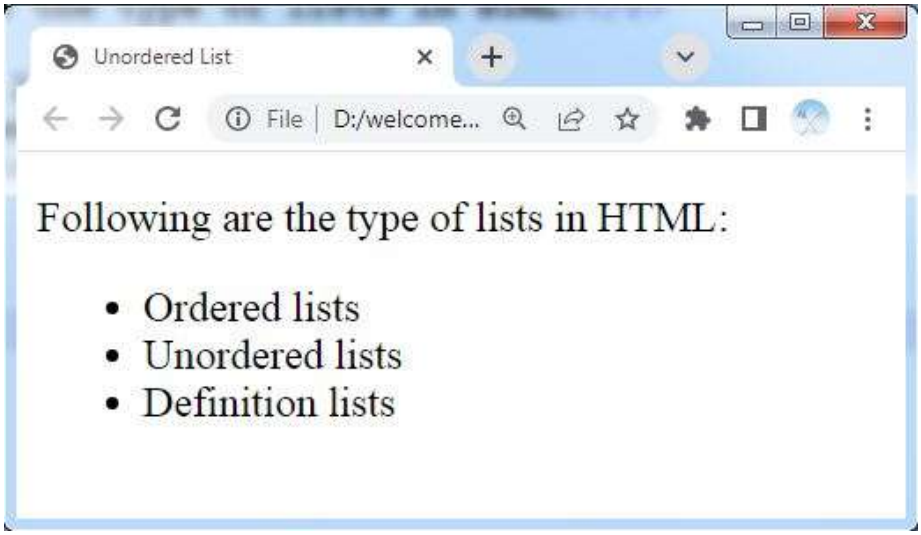


2. **Un-order List: UL Element** is used to display the list information in **un-order and random** manner.

Example:

```
<!DOCTYPE HTML>
<HTML>
    <HEAD>
        <TITLE>
            Unordered List
        </TITLE>
    </HEAD>
    <BODY>
        <P>Following are the type of lists in HTML:</P>
        <UL>
            <LI>Ordered lists
            <LI>Unordered lists
            <LI>Definition lists
        </UL>
    </BODY>
</HTML>
```

Output:



- **Nested List:** Insert one list in another list is called nested list.

```
<!DOCTYPE HTML>
<HTML>
    <HEAD>
        <TITLE>
            Nested List
        </TITLE>
    </HEAD>
    <BODY>
        <UL>
            <LI>How to make tea:
            <OL>
                <LI>Boil water. Take 3/4th cup for every cup you want to make.
                <LI>Now add tea powder according to number of cups.
                <LI>Now add sugar as per your taste buds.
                <LI>Bring the mixture to a boil.
                <LI>Sieve and pour in cups.
                <LI>Add warm milk for milk tea.
            </OL>
                <LI>How to make coffee:
            <OL>
                <LI>Heat up the water and get a mug and put in two tablespoons of sugar and a teaspoon of instant coffee.
                <LI>When the water is ready, pour in as much to fill the cup, allowing for any liquid you plan to add.
                <LI>Add creamer to the coffee and stir.
            </OL>
        </UL>
    </BODY>
</HTML>
```

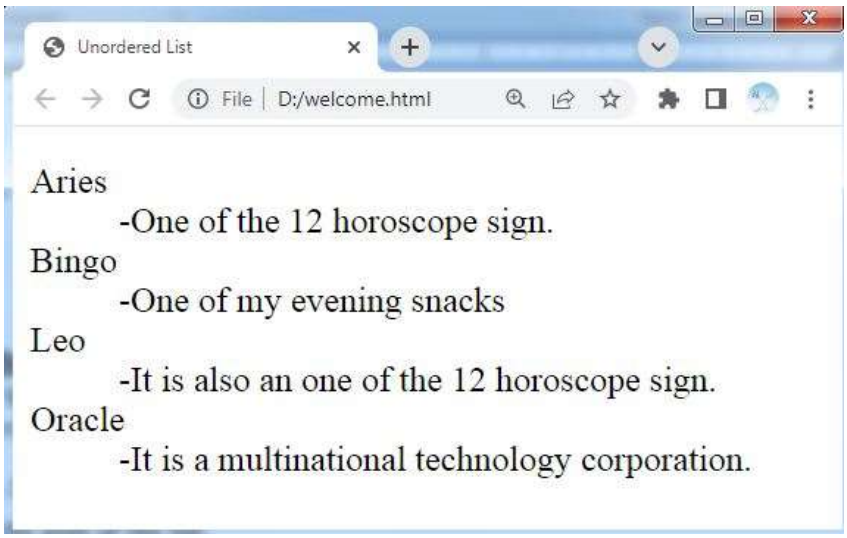
3. Description List or Definition List:

- **DL Element** is used to give the description of items.
- The definition list is very appropriate when you want to present glossary, list of terms or other name-value list.
- The HTML definition list contains following three tags:
  1. **<dl> tag** defines the start of the list.
  2. **<dt> tag** defines a term.
  3. **<dd> tag** defines the term definition (description).

Example:

```
<!DOCTYPE HTML>
<HTML>
        <HEAD>
                <TITLE>
                        Definition List
                </TITLE>
        </HEAD>
        <BODY>
                <dl>
                        <dt>Aries</dt>
                                <dd>-One of the 12 horoscope sign.</dd>
                        <dt>Bingo</dt>
                                <dd>-One of my evening snacks</dd>
                        <dt>Leo</dt>
                                <dd>-It is also an one of the 12 horoscope sign.</dd>
                        <dt>Oracle</dt>
                                <dd>-It is a multinational technology corporation.</dd>
                </dl>
        </BODY>
</HTML>
```

Output:



Working with Links and URLs

Exploring Hyperlinks:

- A hyperlink is an underlined word or image or icon that contains specific address of a web page.
- The address in hyperlink in the form of **Uniform Resource Locator** (URL).
- A hyperlink interconnects the current web page with the other web pages available on the internet.
- The hyperlink redirects the user to another web page, image, or file.
- The hyperlink is created by using element A [<a> tag].
- Attributes of the <a> tag

Attribute	Description
href	URL of Target recourse.
target	Specify the window where you open.
id	Specify the fragment identifier.

The *target* Attribute:

- The A element uses the target attribute to specify the window where you open a document when hyperlink is clicked.
- You can open a document in the same window or another window by sing target attribute.
- HTML defines five target values: **_blank**, **_self**, **_parent**, **_top**, and **framename**.
- Name of the frame must be preceded by underscore (_) character.

Value	Description
_blank	Opens the linked document in a new unnamed window
_self	Opens the linked document in the current window
_parent	Opens the linked document in the parent window
_top	Opens the linked document in the top most window
framename	Opens the linked document in a named window

The *id* Attribute:

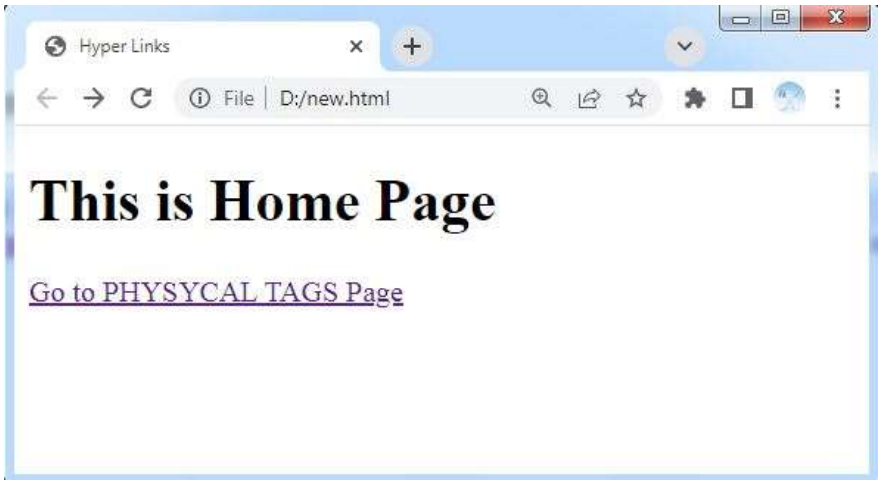
- The A element uses the id attribute to create a fragment identifier within a document.
- A fragment identifier specifies a particular location within a document.
- You can navigate different locations within a document by using the id attribute.
- The id attribute takes a character string as a value.
- This value must be unique in the same document; however, it can be used in different documents.
- For example, **<A id="top"></A>** first defines a location within the document and then **<A href="#top"> Go to top</A>** provides reference to that location with a hyperlink containing fragment identifier.

Example 1: Design a webpage that demonstrate the hyperlink.

Page1:

```
<!DOCTYPE html>
<html>
        <head>
                <title>Hyper Links</title>
        </head>
        <body>
                <h1> This is Home Page</h1>
                <a href="physicaltags.html">Go to PHYSYCAL TAGS Page</a>
        </body>
</html>
```

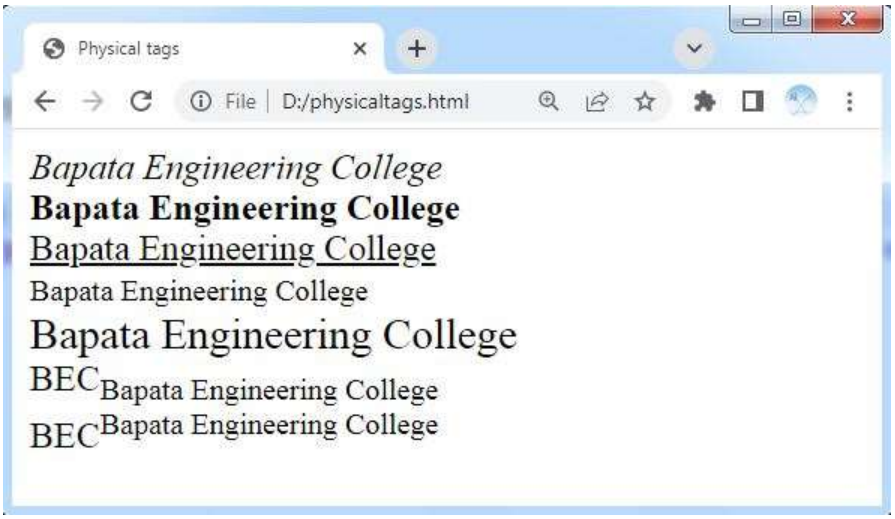
Output:



Page 2:

```
<!DOCTYPE html>
<html>
	<head>
		<title>Physical tags</title>
	</head>
	<body >
		<i>Bapata Engineering College</i><br>
		<b>Bapata Engineering College</b><br>
		<u>Bapata Engineering College</u><br>
		<small>Bapata Engineering College</small><br>
		<big>Bapata Engineering College</big><br>
		BEC<sub>Bapata Engineering College</sub><br>
		BEC<sup>Bapata Engineering College</sup><br>
	</body>
</html>
```

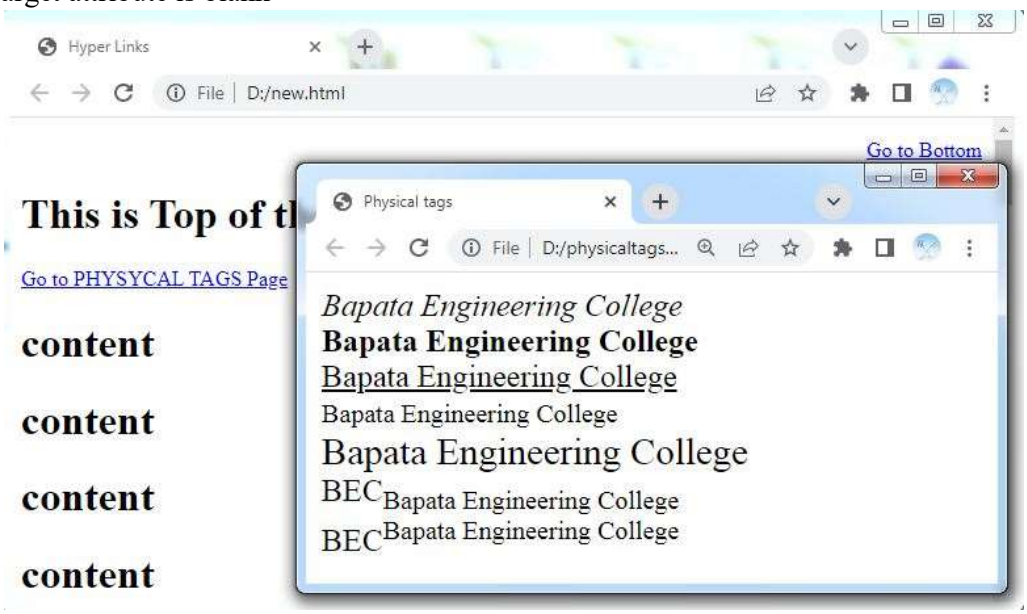
Output:



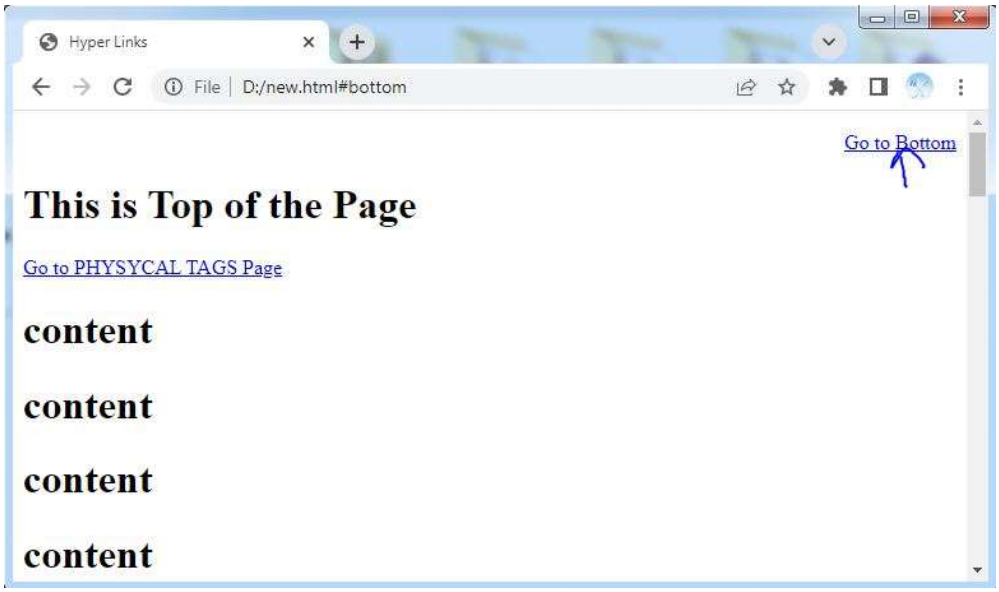
Example 1: Design a webpage that demonstrate the target and id attributes in hyperlink.

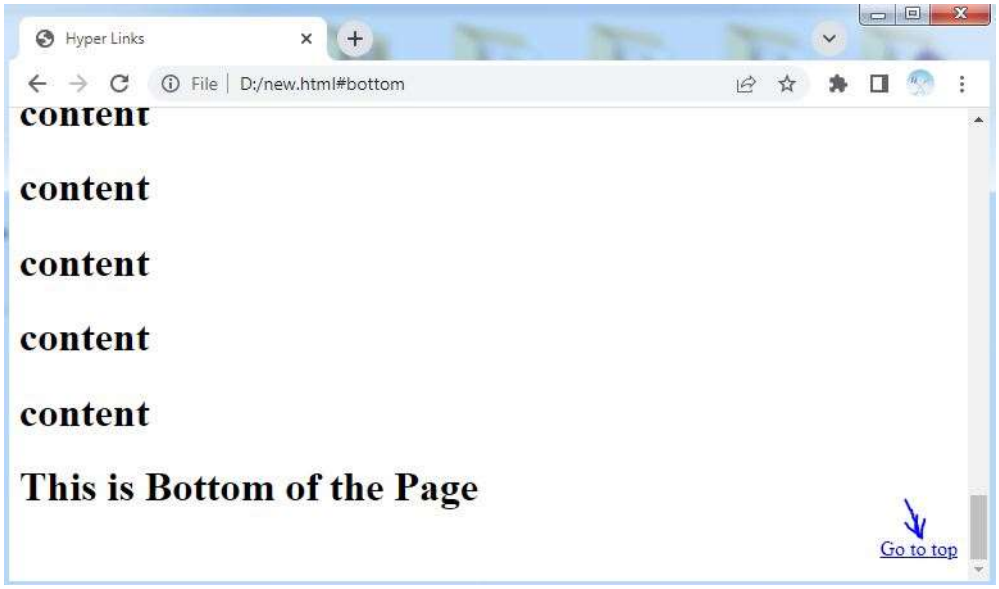
```
<!DOCTYPE html>
<html>
	<head>
		<title>Hyper Links</title>
	</head>
	<body><a id="top"></a>
		<p style="text-align:right"><a href="#bottom">Go to Bottom</a></p>
		<h1> This is Top of the Page</h1>
		<a href="physicaltags.html" target="_blank">Go to PHYSYCAL TAGS Page</a>

		<h1>content</h1>
		<h1>content</h1>
		<h1>content</h1>
		<h1>content</h1>
		<h1>content</h1>
		<h1>content</h1>
		<h1>content</h1>
		<h1>content</h1>
		<h1>content</h1>
		<h1>content</h1>
		<h1>content</h1>
		<h1>content</h1>
		<h1>content</h1>
		<h1>content</h1>
		<h1> This is Bottom of the Page</h1>

		<a id="bottom"></a>
		<p style="text-align:right"><a href="#top">Go to top</a></p>
	</body>
</html>
```

Output1: Target attribute is blank



Output 2: id attribute





LINK Element:

- The LINK element to link a web page with an external resource or document, such as CSS file.
- The LINK element is used to import the content of a script or resource, such as a stylesheet into your HTML document.
- Example:

```
<link  rel="stylesheet"  type="text/css"  href="styles.css">
```

- **Attributes are**
  - **href**: It s used to specify the URL of the linked document.
  - **rel**: It is used to specify the relationship between the current and the linked document.
  - **type**: It is used to set/return the content type of the linked document.

Creating Tables

- Tables allow you to organize data, such as text, images, and links, in the form of rows and columns.
- Tables are used on the web site for two purposes
  - ✓ Arrange the information in tabular format
  - ✓ Create a page layout
- You can create multiple tables inside a table, which are known as "Nested tables".
- You can combine rows and columns of the table.

Describing TABLE Element:

- Table can create by using **TABLE** element.
- Table can consist of rows and columns. Where each row is divided into several **data cell**.
- A **cell** can contain text, list, image etc.
- Following are the elements used in <TABLE> element
  - ✓ **TH**
  - ✓ **TD**
  - ✓ **TR**
  - ✓ **CAPTION**
  - ✓ **COL**
  - ✓ **COLGROUP**
  - ✓ **TBODY**
  - ✓ **THEAD**
  - ✓ **TFOOT**

TD and TH:

- In HTML, the table cells are divided into two categories:
  - ✓ **header cells**
  - ✓ **standard cells**
- The header cell of a table contains the header information, which can be the heading of a column and other content; these are created by using **TH element**.
- Syntax for TH: **<th>heading</th>**
- The standard cells of a table contain text, images, links, and other tables. These cells are created by using **TD element**.
- Syntax for TH: **<td>heading</td>**

TR:

- The TR element is used to define the rows of a table.
- A row contains one or more table cells and table data.
- You can use the TH element in following context
  - ➢ Child of THEAD element
  - ➢ Child of TBODY element
  - ➢ Child of TFOOT element
  - ➢ Child of TABLE element

CAPTION

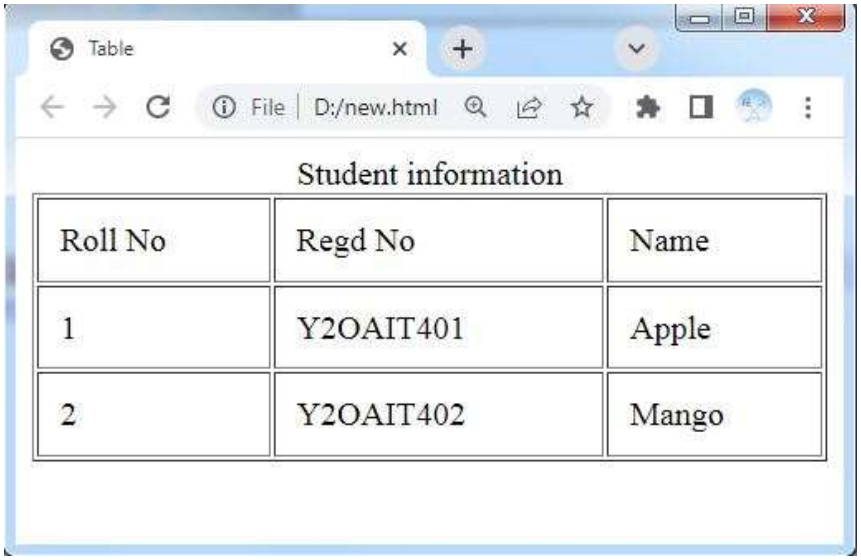
- The CAPTION element is used to create the caption of the table and is used in conjunction with the TABLE element.
- A table should have only one CAPTION element and must be placed after starting tag of the table.
- Syntax:

```
<table>
    <caption> Caption content here … </caption>
</table>
```

Example: Design the web page that demonstrates the table with **TD, TH, TR,** and **CAPTION** elements.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Table</title>
    </head>
    <body>
            <table width="100%" cellpadding="10" border="1">
            <caption>Student information</caption>
            <tr>
                <td>Roll No</td>
                <td>Regd No</td>
                <td>Name</td>
            </tr>
            <tr>
                <td>1</td>
                <td>Y2OAIT401</td>
                <td>Apple</td>
            </tr>
            <tr>
                <td>2</td>
                <td>Y2OAIT402</td>
                <td>Mango</td>
            </tr>
        </table>
    </body>
</html>
```

Output:



COL:

- COL element is used to define the properties, such as font color, and background color, border of each column of a table separately.
- It has an attribute span which specifies the number of columns a column group should span

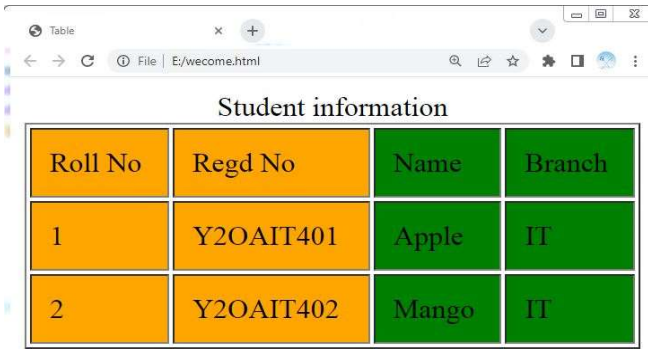
COLGROUP:

- The COLGROUP element is used to specify properties, such as font color, and background color, border, for a group of columns in a table.
- You can use the span attribute of the COLGROUP element to specify the number of columns on which you want to apply properties.

Example: Design the web page that demonstrates the table with **COL,** and **COLGROUP** elements.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Table</title>
    </head>
    <body>
        <table width="100%" cellpadding="10" border="1">
        <caption>Student information</caption>
        <COL style="background-color:orange" />
        <COL style="background-color:orange" />
        <COLGROUP span="2" style="background-color:green" />
        <tr>
            <td>Roll No</td>
            <td>Regd No</td>
            <td>Name</td>
            <td>Branch</td>
        </tr>
        <tr>
            <td>1</td>
            <td>Y2OAIT401</td>
            <td>Apple</td>
            <td>IT</td>
        </tr>
        <tr>
            <td>2</td>
            <td>Y2OAIT402</td>
            <td>Mango</td>
            <td>IT</td>
        </tr>
        </table>
    </body>
</html>
```

OUTPUT:



THEAD:

- The THEAD element is used to define the header for the table and is used in conjunction with the TBODY and TFOOT elements.
- You must define the THEAD element as a child of the TABLE element.

TBODY:

- The TBODY element is used to group the rows of a table and is used in conjunction with the THEAD and TFOOT elements.

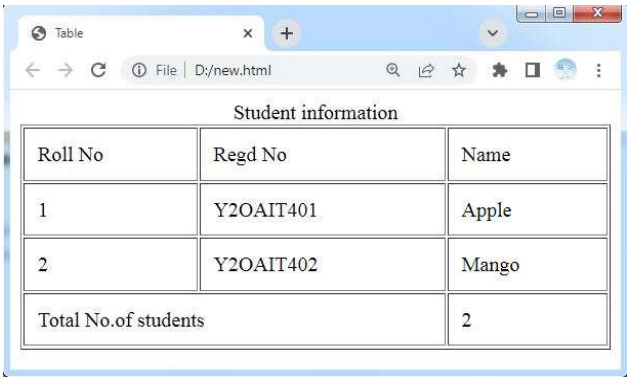
TFOOT:

- The TFOOT element is used to define the footer for the table and is used in conjunction with the TBODY and THEAD elements.

Example: Design the web page that demonstrates the table with **THEAD, TBODY,** and **TFOOT** elements.

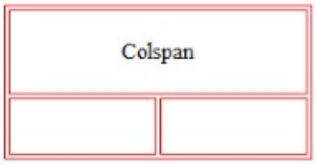
```
<!DOCTYPE html>
<html>
    <head>
        <title>Table</title>
    </head>
    <body>
        <table width="100%" cellpadding="10" border="1">
        <caption>Student information</caption>
        <thead>
            <tr>
                <td>Roll No</td>
                <td>Regd No</td>
                <td>Name</td>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>1</td>
                <td>Y2OAIT401</td>
                <td>Apple</td>
            </tr>
            <tr>
                <td>2</td>
                <td>Y2OAIT402</td>
                <td>Mango</td>
            </tr>
        </tbody>
        <tfoot>
            <tr>
                <td colspan="2">Total N
                <td>2</td>
            </tr>
        </tfoot>
        </table>
    </body>
</html>
```

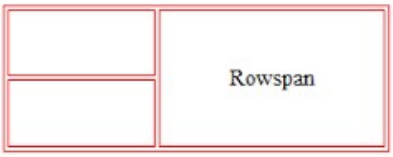
Output:



Spanning Rows and Columns:

- The process of combining one or more adjacent cells is known as spanning.
- You can span cells in two ways
  1. **Spanning rows**
     - The vertical spanning of cells is known as row spanning.
     - Row spanning represented by using ***rowspan*** attribute with the TD and TH elements.
     - ***rowspan*** attributes require a number value that indicates the number of cell.
  2. **Spanning columns**
     - The horizontal spanning of cells is known as column spanning.
     - Column spanning represented by using **colspan** attribute the TD and TH elements.
     - ***colspan*** attributes require a number value that indicates the number of cell.

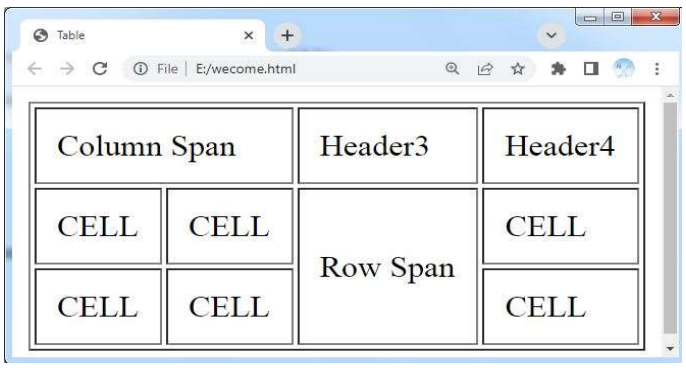




Example: Design the web page that demonstrates the table with **rowspan** and **colspan** attributes.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Table</title>
    </head>
    <body>
        <table width="100%" cellpadding="10" border="1">
        <tr>
            <td colspan="2">Column Span</td>
            <td>Header3</td>
            <td>Header4</td>
        </tr>
        <tr>
            <td>CELL</td>
            <td>CELL</td>
            <td rowspan="2">Row Span</td>
            <td>CELL</td>
        </tr>
        <tr>
            <td>CELL</td>
            <td>CELL</td>
            <td>CELL</td>
        </tr>
        </table>
    </body>
</html>
```

Output:



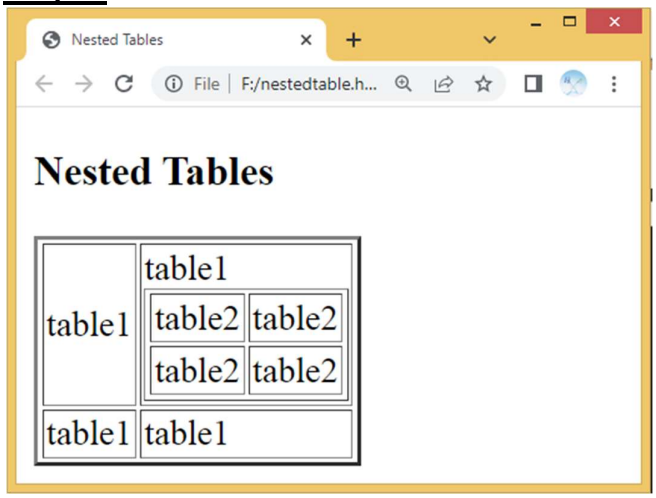
Nested tables: A table placed in another table is called nested tables

Example: Design a web page that demonstrate the nested tables

Source code:

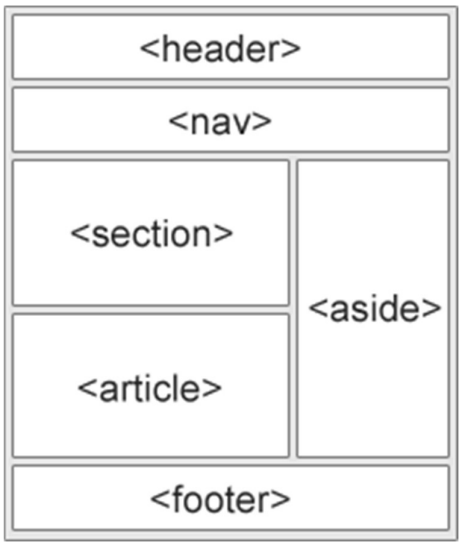
```
<!DOCTYPE html>
<html>
  <head>
    <title>Nested Tables</title>
  </head>
  <body>
        <h3>Nested Tables</h3>
        <table border="2">
                <tr>
                        <td>table1</td>
                        <td>table1
                                <table border="1">
                                        <tr>
                                                <td>table2</td>
                                                <td>table2</td>
                                        </tr>
                                        <tr>
                                                <td>table2</td>
                                                <td>table2</td>
                                        </tr>
                                </table>
                        </td>
                </tr>
                <tr>
                        <td>table1</td>
                        <td>table1</td>
                </tr>
        </table>
  </body>
</html>
```

Output:



Create page layout:

- Websites often display content in multiple columns
- HTML has several semantic elements that define the different parts of a web page:



- <header> - Defines a header for a document or a section
- <nav> - Defines a set of navigation links
- <section> - Defines a section in a document
- <article> - Defines an independent, self-contained content
- <aside> - Defines content aside from the content (like a sidebar)
- <footer> - Defines a footer for a document or a section
- <details> - Defines additional details that the user can open and close on demand
- <summary> - Defines a heading for the <details> element

Working with FRAMES

Note − The <frame> tag deprecated in HTML5. Do not use this element.

- 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.

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.
- Following are important attributes of the <frameset> tag −

S.N	Attribute	Description
1	**Cols**	Specifies how many columns are contained in the frameset.
2	**Rows**	Specifies how many rows are contained in the frameset.
3	**Border**	Specifies the width of the border of each frame in pixels.
4	**Frameborder**	Specifies whether a three-dimensional border should be displayed between frames. This attribute takes value either 1 (yes) or 0 (no).
5	**framespacing**	specifies the amount of space between frames in a frameset.

- Following are the important attributes of <frame> tag −

S.N	Attribute	Description
1	**Src**	This attribute is used to give the file name that should be loaded in the frame.
2	**Name**	This attribute allows you to give a name to a frame.
3	**Frameborder**	This attribute specifies whether or not the borders of that frame are shown
4	**Marginwidth**	This attribute allows you to specify the width of the space between the left and right of the frame's borders and the frame's content.
5	**marginheight**	This attribute allows you to specify the height of the space between the top and bottom of the frame's borders and its contents.
6	**Noresize**	By default, you can resize any frame by clicking and dragging on the borders of a frame.
7	**Scrolling**	This attribute controls the appearance of the scrollbars that appear on the frame. This takes values either "yes", "no" or "auto".
8	**Longdesc**	This attribute allows you to provide a link to another page containing a long description of the contents of the frame

Disadvantages of Frames

- 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.

Example: Design a web page that demonstrate the FRAMES

Frame.html:

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Frames</title>
  </head>

  <frameset rows = "10%,80%,10%" >
          <frame name = "head" src = "head.html" />
          <frameset cols = "10%,80%">
                  <frame name = "side" src = "links.html" />
                  <frame name = "section" />
          </frameset>
          <frame name = "foot" src = "foot.html" />

    <noframes>
      <body>Your browser does not support frames.</body>
    </noframes>
  </frameset>
</html>
```

Head.html:

```
<!DOCTYPE html>
<html>
  <body>
                  <h1 align="center">Welcome to Frames</h1>
  </body>
</html>
```

Links.html:

```
<!DOCTYPE html>
<html>
  <body>
                  <nav>
                          <a href="welcome.html" target="section">Welcome</a> <br><br>
                          <a href="Table.html" target="section">Tables</a> <br><br>
                          <a href="nestedtable.html" target="section">Nested Tables</a> <br><br>
                  </nav>
  </body>
</html>
```

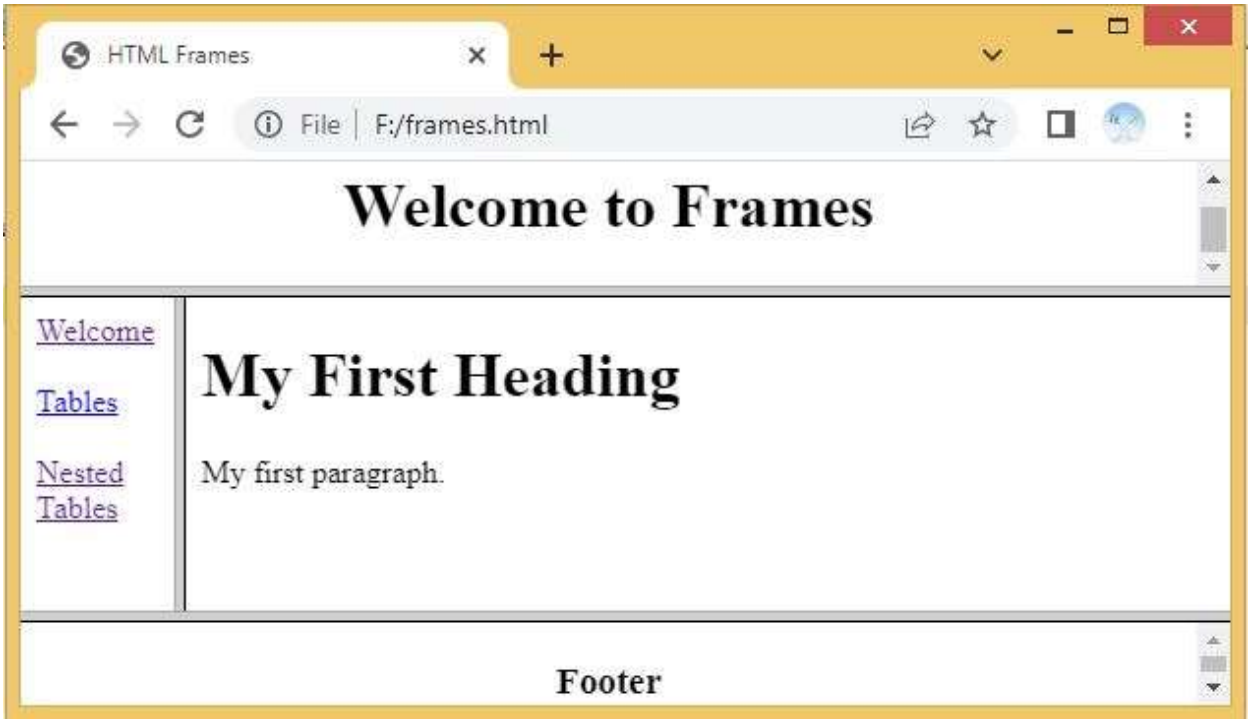
Welcome.html

```
<!DOCTYPE html>
<html>
        <body>
                <h1>My First Heading</h1>
                <p>My first paragraph.</p>
        </body>
</html>
```

Foot.html:

```
<!DOCTYPE html>
<html>
  <body>
                <h3 align="center">Footer</h3>
  </body>
</html>
```

Output:



Working with Images, Colors and Canvas

Inserting image in Web page:

- HTML allows you to insert an image in web page with the help of **IMG element**.
- IMG element can represent by using **<img> tag**.
- Images are not technically inserted into a web page; images are linked to web pages.
- The <img> tag creates a holding space for the referenced image.
- The <img> tag is empty tag; it contains attributes only, and does not have a closing tag.
- The <img> uses several attributes, such as src, id, alt, dir, lang, etc.
- The <img> tag has one required attributes:
  - ✓ **src** - Specifies the path to the image
- If image and HTML files are in **same folder**, then there is **no need** to specify **full path** of the image file in src attribute.
- If image and HTML files are in **different folder**, then there is **you need** to specify **full path** of the image file in src attribute.
- Syntax of img tag

<img src="file name"> //when both files are in same folder

(Or)

<img src="full path of the image with file name"> //when both files are in different folder

- Attributes are

Attribute	Description	required
src	Specify the location of image	**YES**
alt	Specifies an alternate text for an image	NO
width	Specifies the width of an image	NO
height	Specifies the height of an image	NO
ismap	Specifies an image as a server-side image map	NO
usemap	Specifies an image as a client-side image map	NO
class	Define a class name for an Element	NO
dir	Define the direction of the content in the element.	NO
id	Define a unique id for an element.	NO
lang	Define a base language code for the element.	NO
style	Define an inline style for element.	NO
title	Define extra information about element.	NO

Common Image Formats

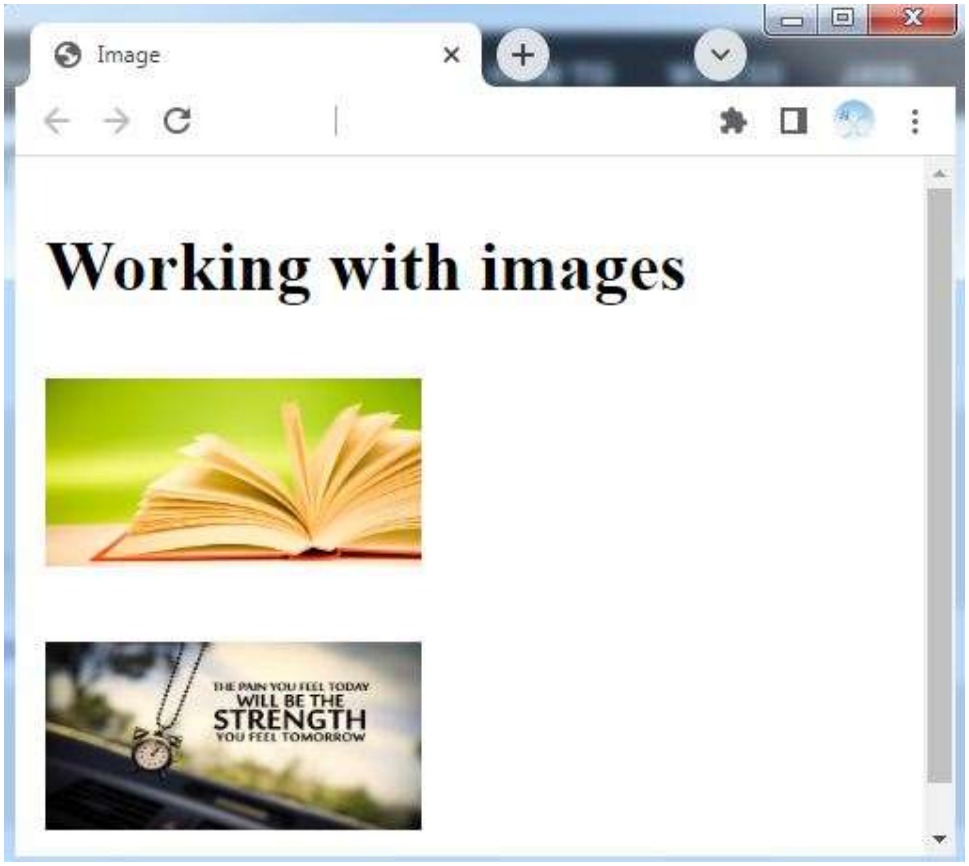
- Here are the most common image file types, which are supported in all browsers (Chrome, Edge, Firefox, Safari, Opera):

Abbreviation	File Format	File Extension
APNG	Animated Portable Network Graphics	.apng
GIF	Graphics Interchange Format	.gif
ICO	Microsoft Icon	.ico, .cur
JPEG	Joint Photographic Expert Group image	.jpg, .jpeg, .jfif, .pjpeg, .pjp
PNG	Portable Network Graphics	.png
SVG	Scalable Vector Graphics	.svg

Example: Design the web page that demonstrates the <img> tag

```
<!DOCTYPE html>
<html>
        <head>
                <title>Image</title>
        </head>
        <body>
                <h3>Working with images</h3>
                <p><img src="D://image1.jpg"   alt="image not found"   width="100"   height="50"></p>
                <p><img src="image2.jpg"   alt="image not found"   width="100"   height="50"></p>
        </body>
</html>
```

Output:



Describing image map:

- A technique in which an image is divided into multiple sections and each section is linked to different web pages is known as image map.
- Image map represented by **<map>** tag
- The linked region of image is called hot region.
- **Syntax:**

```
<img   src="path of the image"    usemap="#name of the map">
<map  name=" name of the map ">
    <area  shape="name of the shape" coords="coordinate position" href="target page">
</map>
```

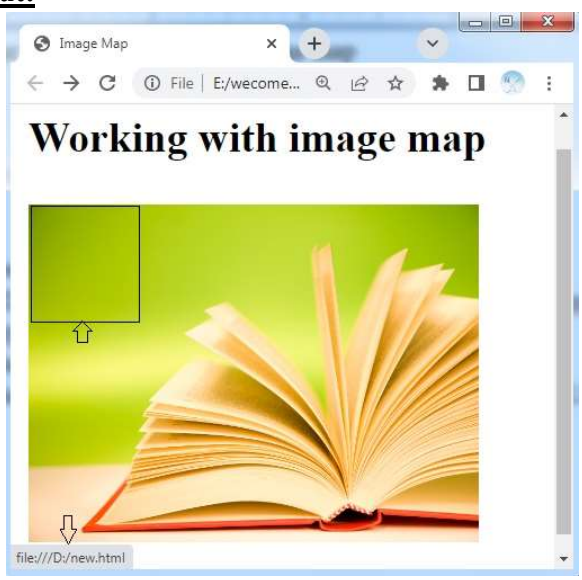
- Example:

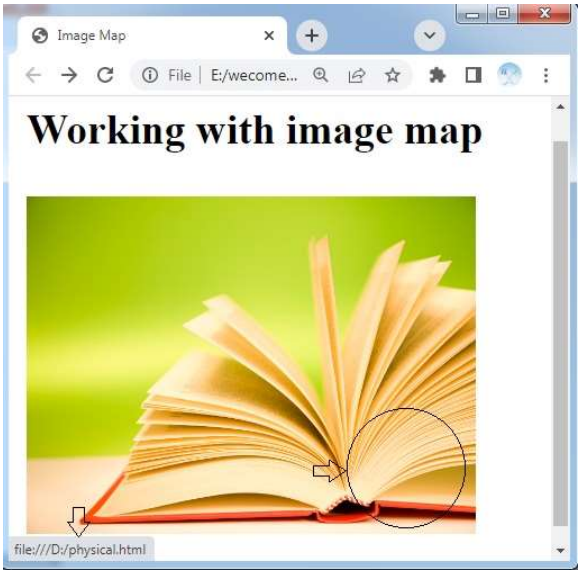
```
<img    src="image1.jpg"  usemap="#mymap">
<map   name="mymap">
        <area shape="rect"   coords="0,0,50,50" href="physical.html" />
        <area shape="circle"   coords="150,150,50"  href="logical.html" />
</map>
```

Example: Design the web page that demonstrates the image map

```
<!DOCTYPE html>
<html>
    <head>
        <title>Image</title>
    </head>
    <body>
        <h3>Working with images</h3>
        <p><img src="D://image1.jpg" width="200" height="200" usemap="#mymap"></p>
        <map name="mymap">
            <area shape="rect" coords="0,0,50,50" href="D:/new.html" />
            <area shape="circle" coords="150,150,50" href="D:/physical.html" />
        </map>
    </body>
</html>
```

Output:





Exploring Colors:

- HTML can specify colors to the text and background of a web page to make it attractive.
- You can apply colors in a web page with the help of
  1. Color name
  2. RGB configuration
  3. Hex Value
  4. Web-safe colors

1. Color name:

- You can use the color in the web page by simply specifying its name.
- HTML defines 16 widely known colors that you can apply to a web page.
- Example:

<body bgcolor="black">

2. RGB configuration:

- The process of displaying the colors by using different combination of **red, green** and **blue** is known as RGB configuration.
- It is set of **three dials**, where the first dials represent red, the second dials represent green, and the third dials represent blue colors.
- The value for each color range from 0 to 255 in decimal.
- Example:

<body bgcolor="rgb(0,0,0)">

3. Hex value:

- Hex value (Hexadecimal number) is **6 digit** or **3 byte** numbers that starts with # sign.
  - ✓ Byte 1 represent red color
  - ✓ Byte 2 represent green color
  - ✓ Byte 3 represent blue color
- Each byte contains some numbers from 00 to 99 and alphabets from AA to FF in hexadecimal notation.
- For example, the hexadecimal number for black is **#000000**.
- Here, The **first two digits(00)** represents the amount of **red color**
  The **second two digits(00)** represents the amount of **green color**
  The **last two digits(00)** represents the amount of **blue color**
- Example:

<body bgcolor="#000000">

4. Web-safe colors:

- A computer system is used to support only 256 colors, out of which 216 colors are standard colors displayed correctly by all the computer system. These colors are known as web-safe colors.
- The web-safe colors are defined by using only values **00, 33, 66, 99, AA, CC,** or **FF**.
- For example, #99FF00 is a web safe color but #221144 is not.

Example: Design the web page that demonstrates the colors

```
<!DOCTYPE HTML>
<HTML>
<HEAD>
        <TITLE>using color names</TITLE>
</HEAD>
<BODY>
        <h1>Displaying the colors</h1>
        <p style="color:Fuchsia">Displaying the colors using name</p>
        <p style="color:rgb(255, 0, 255)">Displaying the colors using GRB configuration</p>
        <p style="color:#FF00FF">Displaying the colors using Hex value</p>
        <p style="color:#FF00FF">Displaying the colors using web safe</p>
</BODY>
</HTML>
```

Output:

Introducing canvas:

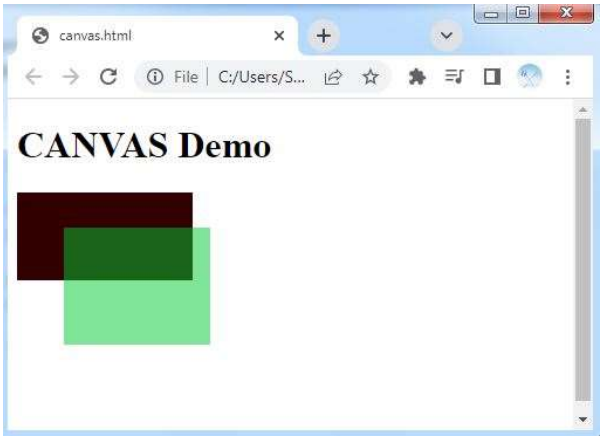
- Canvas element is introduced in HTML5 to display 2D shapes and graphics on a web page.
- It is also used to apply various transformations, such as rotate and blur on image.
- The content in the canvas element is only displayed when the web browser doesn't support the element.
- Attributes are height and width.
- Syntax:

<canvas id="canvas" width="100" height="100"> content</canvas>

Example: Design the web page that demonstrates the canvas

```
<!DOCTYPE HTML>
<HTML>
	<HEAD>
		<SCRIPT type="application/x-javascript">
			function displayCanvas()
			{
				var mycanvas = document.getElementById("myCanvas");
				if (mycanvas.getContext) {
				var contex = mycanvas.getContext('2d');

				contex.fillStyle = "rgb(50,0,0)";
				contex.fillRect (0, 0, 150, 75);

				contex.fillStyle = "rgba(0, 200, 50, 0.5)";
				contex.fillRect (40, 30, 125,100 );

				}
			}
		</SCRIPT>
	</HEAD>
	<BODY onload="displayCanvas();">
		<h1>CANVAS Demo</h1>
		<CANVAS id="myCanvas" width="300" height="200">
			Your browser does not support the CANVAS element.
		</CANVAS>
	</BODY>
</HTML>
```

Output:



Working with FORMS

- A Form is an area on web page that consists of **plain text, HTML elements,** and **controls**.
- Plain text and HTML elements are used to structure the form.
- Controls (**Forms Fields**) used to make the form interactive by allowing a user to enter information.
- A form is created by using FORM element **(<form>** tag) and it having starting and ending tags.
- Forms contain many types of form elements, such as text boxes, radio buttons, check boxes, buttons and drop-down lists.
- Syntax: **<form> form elements </form>**
- **Attributes of the FORM element**

Attribute	Description
name	Refers to the name of the form
method	Specifies how information is sent from Browser to server
action	Refer to the URL of program in server that processes form
autocomplete	Enables autocomplete feature in form
target	Opens the action URL in specified target

Elements: **FORM** element contains following elements

S.No	Element	S.No	Element
1	LABEL	7	TEXTAREA
2	INPUT	8	FIELDSET
3	BUTTON	9	LEGEND
4	SELECT	10	DATALIST
5	OPTION	11	KEYGEN
6	OPTGROUP	12	OUTPUT

1. LABEL Elements

- **LABEL Element (<label> tag)** provides description for the controls.
- Syntax:

<label> form_content **</label>**

Attribute:

Attribute	Value	Description
for	element_id	Specifies the id of the form element the label should be bound to
form	form_id	Specifies which form the label belongs to

2. INPUT element

- **INPUT** element creates interactive controls for web based form that enables the user to enter the data.
- The **INPUT** element is the most important form element.
- The **INPUT** element can be displayed in several ways, depending on the **type** attribute.
  Syntax: **<input type = "control name" >**
  Example **<input type = "text" >**
- Various types of input controls are

S.N	Control Name	Description	Syntax	Image
1	**text**	• A single-line text field. • Line-breaks are automatically removed from the input value. • This is default type value. • Attributes are: **name, size, maxlength**, and **value.**	<input type="text">	
2	**search**	• A single-line text field for entering search strings. • Line-breaks are automatically removed from the input value. • Attributes are: **name, size, maxlength**, and **value.**	<input type="search">	
3	**password**	• Used to protect the information that a user does not want to share with others. • Similar to text box but the entered text was not readable format. • Attributes are: **name, size, maxlength**, and **value.**	<input type="password">	
4	**number range**	• number type is used to enter only number in input controls • range type of the input controls allows you to enter a value within a specific range. • Attributes for number and range are min and max.	<input type="number"> <input type="range" min="0" max="10">	
5	**url**	• Used to enter valid path of webpage.	<input type="url">	
6	**email**	• Creates an input filed which allow a user to enter the e-mail address with pattern validation.	<input type="email">	
7	**tel**	• Creates an input filed to enter the telephone number. • The "tel" type does not have default validation such as email • Because telephone number pattern can vary worldwide.	<input type="tel">	
8	**checkbox**	• Defines checkboxes which allow select or deselect one or more items from a given set of items. • Attributes are: **name, value** and **checked**	<input type="checkbox">	
9	**Radio**	• Defines a radio button which allows select one item from a given set of items. • Attributes are: **name, value** and **checked**	<input type="radio">	

10	**date and time**	• Defines an input field for selection of date and time. • Depending on browser support, a date picker can show up in the input field.	<input type="date"> <input type="time"> <input type="datetime-local">	
11	**file**	• File is used to select one or more files from user device storage. • Once you select the file, and after submission, this file can be uploaded to the server with the help of JS code and file API. • Attributes are **size, maxlength** and **accept**	<input type="file">	
12	**submit**	• Submit defines a **submit button** to submit the form to the server when the "click" event occurs. • Attributes are: **name** and **value**.	<input type="submit">	
13	**reset**	• defines a **reset button** that will erase all form values and set the default values	<input type="reset">	
14	**hidden**	• Used to pass variables and values from one form to another form, without forcing the user to re-enter the information. • Include data that cannot be seen or modified by users when a form is submitted. • Attributes are: **name,** and **value**	<input type="hidden">	It not visible

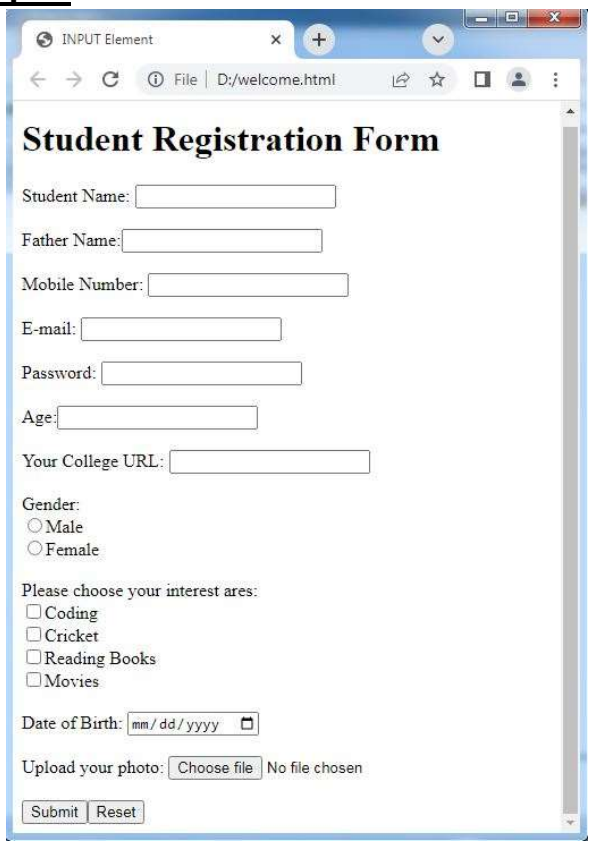
Example: Design a web page to demonstrate the INPUT elements in forms

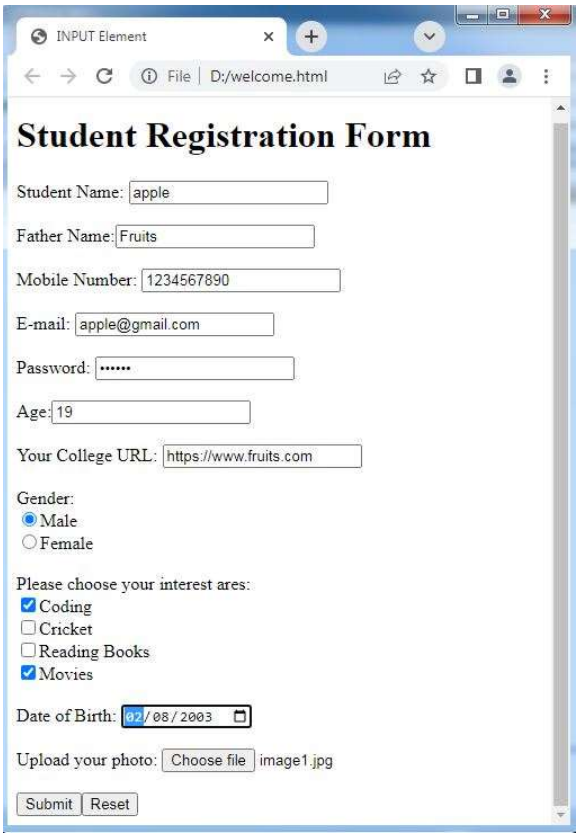
Source code:

```
<!DOCTYPE html>
<html lang="en-US">
    <head>
        <title>INPUT Element</title>
    </head>
    <body>
        <h1>Student Registration Form</h1>
        <form>
        Student Name: <input type="text"><br><br>
        Father Name:<input type="text"><br><br>
        Mobile Number: <input type="tel"><br><br>
        E-mail: <input type="email"><br><br>
        Password: <input type="password"><br><br>
        Age:<input type="number"><br><br>
```

```
Your College URL: <input type="url"><br><br>
Gender:<br><input type="radio" name="gender" value="Male">Male<br>
<input type="radio" name="gender" value="Female">Female<br><br>
Please choose your interest ares:<br>
<input type="checkbox" name="interest" value="Coding">Coding<br>
<input type="checkbox" name="interest" value="Cricket">Cricket<br>
<input type="checkbox" name="interest" value="Reading">Reading
Books<br>
<input type="checkbox" name="interest" value="Movies">Movies<br><br>
Date of Birth: <input type="date" ><br><br>
Upload your photo: <input type="file" ><br><br>
<input type="hidden" name="regdno" value="401" >
<input type="submit" value="Submit" ><input type="reset" value="Reset" >
</form>
</body>
</html>
```

Output:





3. BUTTON Element:

- BUTTON element is used to add button control on the form.
- Button control can perform submitting, or resetting of the details of the form.
- The difference between input element and button element is only in change the appearance i.e., in input type element you can take text only where as in button you can take text, image, or any other multimedia as BUTTON element.
- Button element can have both starting and ending tag of BUTTON element and place the control in between the tag to make a BUTTON.
- The <button> tag defines a clickable button.
- BUTTON element has **type** attribute and it can create three kinds of buttons: submit, reset, and normal.
- BUTTON element attributes are as follows

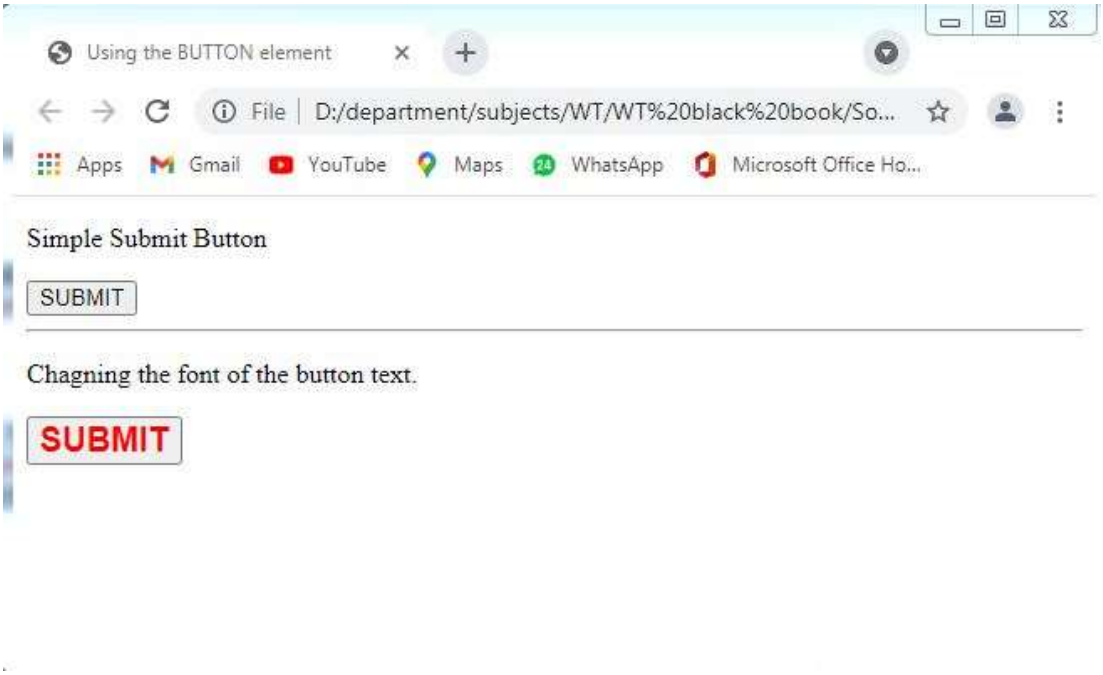
Attribute	Value	Description	New to HTML5
Autofocus	autofocus	Specifies that a button should automatically get focus when the page loads	YES
Disabled	disabled	Specifies that a button should be disabled	NO
Form	form_id	Specifies which form the button belongs to	YES
formaction	URL	Specifies where to send the form-data when a form is submitted. Only for type="submit"	YES
formenctype	application/x-www-form-urlencoded multipart/form-data text/plain	Specifies how form-data should be encoded before sending it to a server. Only for type="submit"	YES
formmethod	get post	Specifies how to send the form-data (which HTTP method to use). Only for type="submit"	YES
formnovalidate	formnovalidate	Specifies that the form-data should not be validated on submission. Only for type="submit"	YES
formtarget	_blank _self _parent _top framename	Specifies where to display the response after submitting the form. Only for type="submit"	YES
name	Name	Specifies a name for the button	NO
type	button reset submit	Specifies the type of button	NO
value	Text	Specifies an initial value for the button	NO

Example: Design a web page to demonstrate the BUTTON element in FORM

Source code:

```
<!DOCTYPE HTML>
<HTML>
    <HEAD>
        <TITLE>
            Using the BUTTON element
        </TITLE>
    </HEAD>
    <BODY>
        <P>Simple Submit Button</P>
        <BUTTON type="submit">SUBMIT</BUTTON>
        <HR>
        <P>Chagning the font of the button text.</P>
        <BUTTON type="submit"   style="color:red; font-size:20px;">
            <B>SUBMIT</B>
        </BUTTON>
    </BODY>
</HTML>
```

OUTPUT:



Multiple-Choice Elements

- It offers multiple choices to the user in web page.
- Multiple choice elements are **SELECT**, **OPTION**, **OPTGROUP**.

4 SELECT Element:

- SELECT element is used to select single option from multiple options.
- SELECT element does provide default options like radio button.
- SELECT element has both station and ending tag and all options are written in between the tags.
- SELECT element having following attributes

Attribute	Value	Description	New in HTML
autofocus	autofocus	Specifies that the drop-down list should automatically get focus when the page loads	YES
disabled	disabled	Specifies that a drop-down list should be disabled	NO
form	*form_id*	Defines which form the drop-down list belongs to	YES
multiple	multiple	Specifies that multiple options can be selected at once	NO
name	*name*	Defines a name for the drop-down list	NO
size	*number*	Defines the number of visible options in a drop-down list	NO

5. OPTION Element:

- OPTION Element is used to define option written within SELECT element.
- The options are created by embedding the OPTION element with in opening and ending tags.
- The content within the option element is plain text only.
- OPTION Element attributes as follows

Attribute	Value	Description
disabled	disabled	Specifies that an option should be disabled
label	*Text*	Specifies a shorter label for an option
selected	selected	Specifies that an option should be pre-selected when the page loads
value	*Text*	Specifies the value to be sent to a server

6. OPTGROUP Element:

- It is used to create nested and cascading drop-down list.
- In both types of lists, the related items are grouped under specific headings.
- OPTGROUP element having the following attributes.

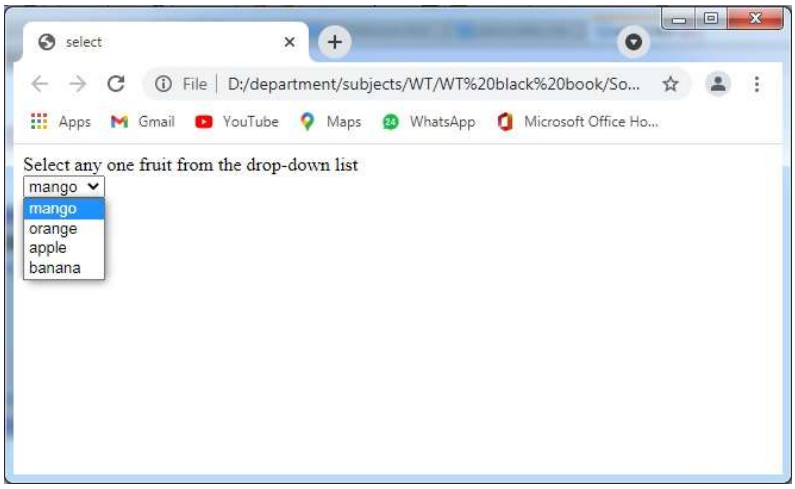
Attribute	Value	Description
disabled	Disabled	Specifies that an option-group should be disabled
label	*Text*	Specifies a label for an option-group

Example: Design a web page to demonstrate the **SELECT** and **OPTION** elements

Source code:

```
<!DOCTYPE HTML >
<HTML>
    <HEAD>
        <TITLE>select</TITLE>
    </HEAD>
    <BODY>
        <FORM>
            Select any one fruit from the drop-down list<BR/>
            <SELECT>
                <OPTION value="mango">mango</OPTION>
                <OPTION value="orange">orange</OPTION>
                <OPTION value="apple">apple</OPTION>
                <OPTION value="banana">banana</OPTION>
            </SELECT>
        </FORM>
    </BODY>
</HTML>
```

Output:

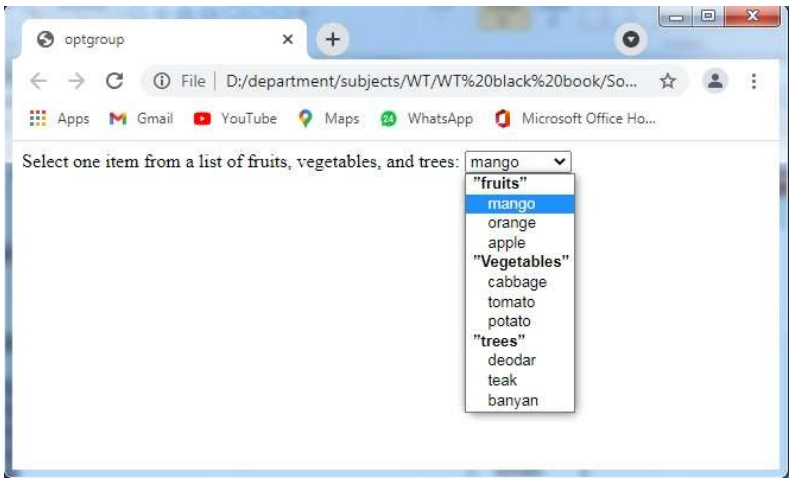


Example: Design a web page to demonstrate the OPTGROUP element.

Source code:

```
<!DOCTYPE  HTML >
<HTML>
    <HEAD>
        <TITLE>optgroup</TITLE>
    </HEAD>
    <BODY>
        <FORM>
        Select one item from a list of fruits, vegetables, and trees:
        <SELECT>
            <OPTGROUP label=”fruits”>
                <OPTION value="mango">mango</OPTION>
                <OPTION value="orange">orange</OPTION>
                <OPTION value="apple">apple</OPTION>
            </OPTGROUP>
            <OPTGROUP label=”Vegetables”>
                <OPTION value="cabbage">cabbage</OPTION>
                <OPTION value="tomato">tomato</OPTION>
                <OPTION value="potato">potato</OPTION>
            </OPTGROUP>
            <OPTGROUP label=”trees”>
                <OPTION value="deodar">deodar</OPTION>
                <OPTION value="teak">teak</OPTION>
                <OPTION value="banyan">banyan</OPTION>
            </OPTGROUP>
        </SELECT>
        </FORM>
    </BODY>
</HTML>
```

OUTPUT:



7. TEXTAREA Element:

- Text area is similar to the text box except that in the text box, you can enter single line of information whereas in textarea you can enter multiple lines of information.
- The content in TEXTAREA element should only plain text.
- You can adjust the size of textarea by using the two attributes: cols and rows.
- TEXAREA element covers entire web page, if it requires.
- Attributes of TEXTAREA element

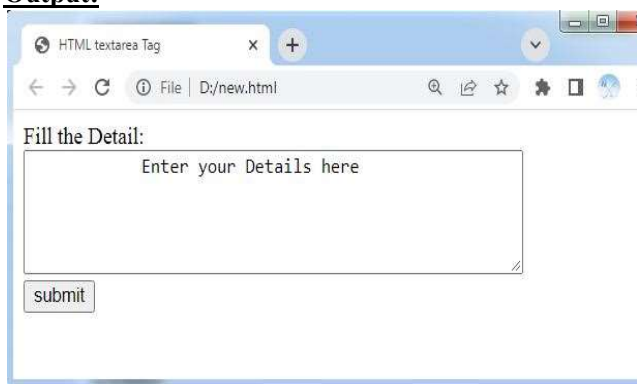
Attribute	Value	Description
autofocus	Autofocus	Specifies that a text area should automatically get focus when the page loads
cols	Number	Specifies the visible width of a text area
dirname	textareaname.dir	Specifies that the text direction of the textarea will be submitted
disabled	Disabled	Specifies that a text area should be disabled
form	form_id	Specifies which form the text area belongs to
maxlength	Number	Specifies the maximum number of characters allowed in the text area
name	Text	Specifies a name for a text area
placeholder	Text	Specifies a short hint that describes the expected value of a text area
readonly	Readonly	Specifies that a text area should be read-only
required	Required	Specifies that a text area is required/must be filled out
rows	Number	Specifies the visible number of lines in a text area
wrap	hard soft	Specifies how the text in a text area is to be wrapped when submitted in a form

Example: Design a web page to demonstrate the OPTGROUP element.

Source code:

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML textarea Tag</title>
  </head>
  <body>
    <form>
       Fill the Detail:  <br />
       <textarea rows = "5" cols = "50" name = "description">
         Enter your Details here
       </textarea><br />
       <input type = "submit" value = "submit" />
    </form>
  </body>
</html>
```

Output:



8 FIELDSET Element

- The FIELDSET Element (<fieldset> tag) is used to group related elements in a form.
- The <fieldset> tag draws a box around the related elements.

9. LEGEND Element

- The <legend> tag defines a caption for the FIELDSET Element.
- By using the fieldset tag and the legend tag, you can make your forms much easier to understand for your users.
- The HTML <fieldset> tag also supports the following additional attributes −

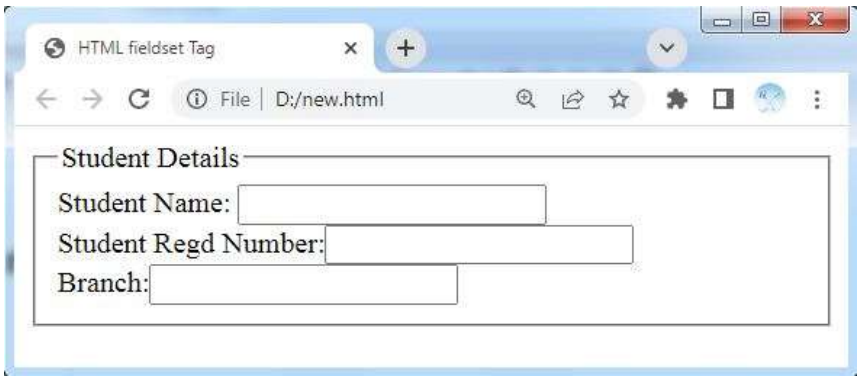
Attribute	Value	Description
align	left right center top bottom	Deprecated − Specifies the content alignment.
disabled	Disabled	Specifies that a group of related form elements should be disabled.
form	form_id	Specifies forms which belong to fieldset.
name	Text	Specifies a name for fieldset.

Example: Design a web page to demonstrate the OPTGROUP element.

Source code:

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML fieldset Tag</title>
  </head>
  <body>
    <form>
      <fieldset>
        <legend>Student Details</legend>
        Student Name: <input type = "text"><br />
        Student Regd Number:<input type = "text"><br />
        Branch:<input type = "text" >
      </fieldset>
    </form>
  </body>
</html>
```

Output:



10. DATALIST Element:

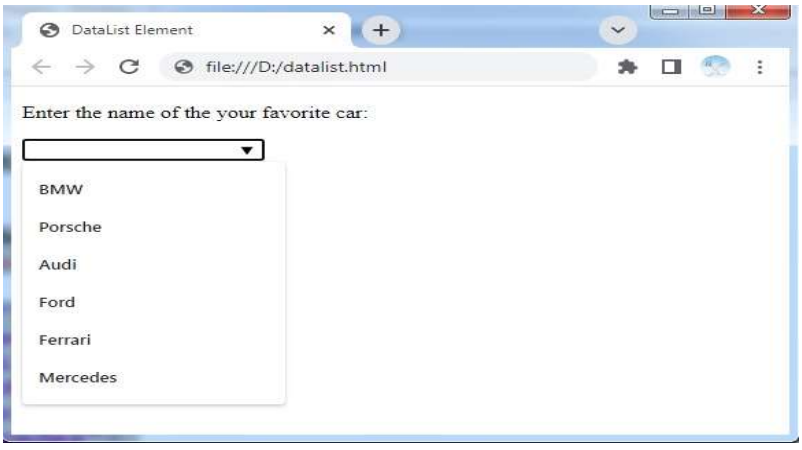
- The DATALIST Element used to display the **list of the predefined options** that the user may want to select as input.
- The DATALIST Element enables auto complete feature on the forms i.e., a list of predefined words is dropped down to choose.
- The DATALIST Element is used with **INPUT element**, in which the **list attribute** is specified.
- The value of list and id attributes must same to link the INPUT element.
- The OPTION element specifies the list of the options that are to be displayed.

Example: Design a web page to demonstrate the OPTGROUP element.

Source code:

```
<!DOCTYPE HTML>
<HTML>
	<HEAD>
		<TITLE>DataList Element</TITLE>
	</HEAD>
	<BODY>
		<P>Enter the name of the your favorite car:</P>
		<FORM>
		<INPUT type="text" name="favCar" list="cars">
		<DATALIST  id="cars">
			<OPTION value="BMW">
			<OPTION value="Porsche">
			<OPTION value="Audi">
			<OPTION value="Ford">
			<OPTION value="Ferrari">
			<OPTION value="Mercedes">
		</DATALIST>
		</FORM>
	</BODY>
</HTML>
```

Output:



11. KEYGEN Element:

- The KEYGEN Element used to generate the key pair.
- Which contains private and public keys generated using the KEYGEN element to secure the content of the form.
- The private key is encrypted and stored in the key database on local computer.
- The public key is encrypted and submitted to the server along with the form.
- Attributes of the KEYGEN element-

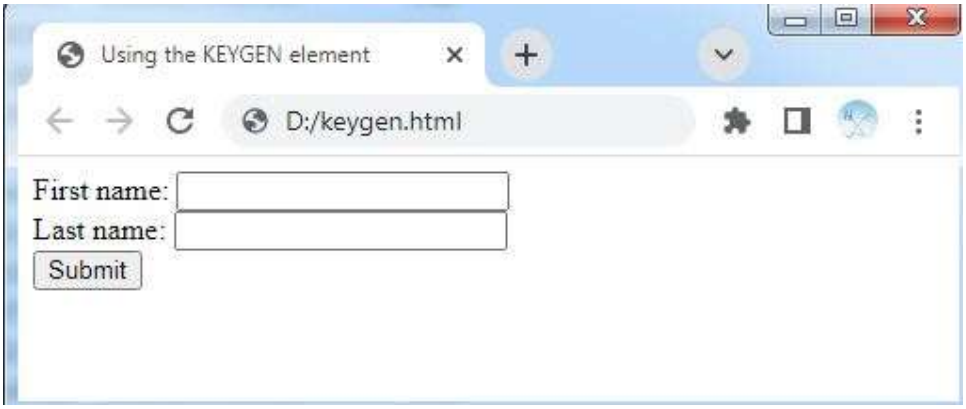
Attribute	Description	New in HTML5
autofocus	Allows the control to get the focus as soon as the page load	YES
challenge	Specify the string that is used for the verification at the time of submission of the form.	YES
disabled	Disable the input control	YES
form	Refers to the id of FORM element	YES
keytype	Specify the type of the key to generate	YES
name	Provide the name to the input control	

Example: Design a web page to demonstrate the OPTGROUP element.

Source code:

```
<!DOCTYPE HTML>
<HTML>
    <HEAD>
        <TITLE>Using the KEYGEN element</TITLE>
    </HEAD>
    <BODY>
        <FORM action="keygen.html" method="post" enctype="text/plain">
            First name: <INPUT type="text" name="fname" /><BR />
            Last name: <INPUT type="text" name="lname" /><BR />
            <KEYGEN name="key" challenge="0987654321" keytype="RSA">
            <INPUT type="submit" value="Submit" />
        </FORM>
    </BODY>
</HTML>
```

Output:



12. OUTPUT Element:

- The OUTPUT Element is used to display the result of the calculation, which can be written using the JavaScript
- The OUTPUT Element having following attributes

Attribute	Description
for	Specifies the id of the form element the label should be bound to
form	Refers to the id of the form.
name	Specifies name of the OUTPUT element.

Example: Design a web page to demonstrate the OPTGROUP element.

Source code:

```
<!DOCTYPE HTML >
<HTML>
<HEAD>
<TITLE>Using the OUTPUT Element</TITLE>
        <SCRIPT type="text/javascript">
                function add()
                {
                document.forms["form"]["resultadd"].value=23+123;
                }
        </SCRIPT>
</HEAD>
<BODY onload="add()">
        <H1>OUTPUT Element.</H1>

        <FORM name="form">
                Adding the two numbers:
                <OUTPUT name="resultadd">
                </OUTPUT>
        </FORM>
        </BODY>
</HTML>
```

Output:

Working with multimedia

- HTML helps to add multimedia files on your website by providing various multimedia elements.
- The multimedia elements are **AUDIO**, **VIDEO**, **EMBED** and **OBJECT**.

AUDIO element:

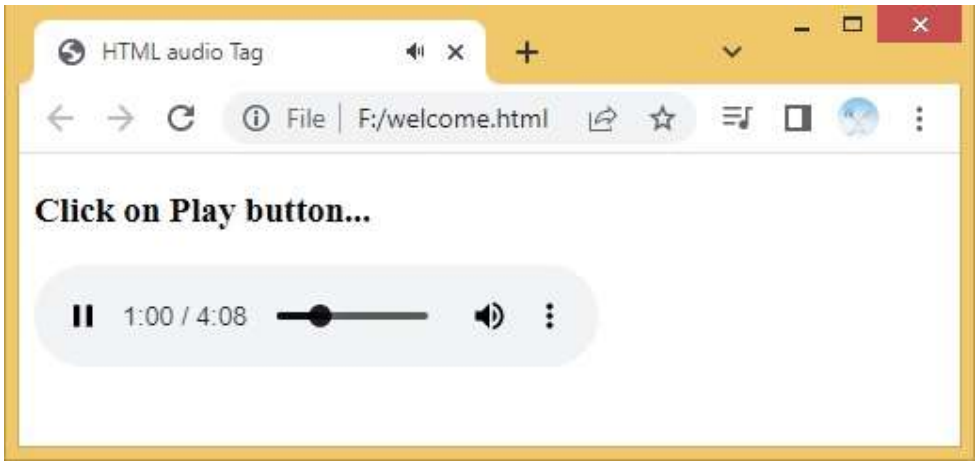
- An audio file used to store audio data on various devices.
- Commonly used audio file formats are mp3, aac, aiff, au, iff, m3u etc.
- Attributes of AUDIO element

Attribute	Description
autoplay	Specifies that the audio will start playing as soon as it is ready
controls	Specifies that audio controls should be displayed (such as a play/pause button etc)
loop	Specifies that the audio will start over again, every time it is finished
preload	Specifies if and how the author thinks the audio should be loaded when the page loads
src	Specifies the URL of the audio file

Example: Design web page that demonstrate the AUDIO element

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML audio Tag</title>
  </head>
  <body>
        <h3>Click on Play button...</h3>
        <audio src = "Komuram.mp3" type = "audio/mp3" controls>
                        Your browser does not support
        </audio>
  </body>
</html>
```

Output:



VIDEOE element:

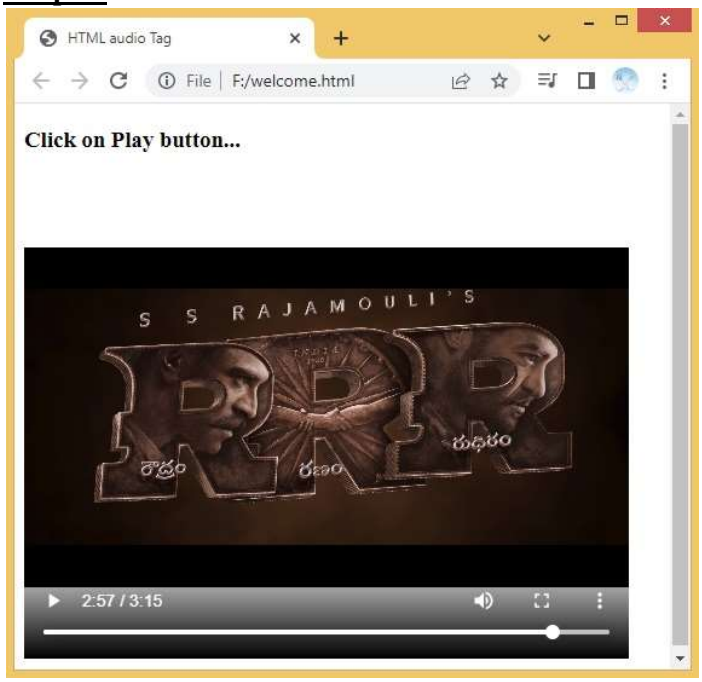
- A video file is a collection of images that displayed in sequence representing sequence in motion.
- Commonly used video file formats are mp4,mpeg, avi, wmv etc
- Attributes of VIDEO element

Attribute	Description
audio	Controls the default state of the video's audio channel
autoplay	Specifies that the video will start playing as soon as it is ready
controls	Specifies that video controls should be displayed (such as a play/pause button etc).
height	Sets the height of the video player
loop	Specifies that the video will start over again, every time it is finished
poster	Specifies an image to be shown while the video is downloading, or until the user hits the play button
preload	Specifies if and how the author thinks the video should be loaded when the page loads
src	Specifies the URL of the video file
width	Sets the width of the video player

Example: Design web page that demonstrate the VIDEO element

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML audio Tag</title>
  </head>
  <body>
    <h3>Click on Play button...</h3>
    <video src = "rrr.mp4" type = "video/mp4" controls width="500" height="400">
                    Your browser does not support
    </audio>
  </body>
</html>
```

Output:



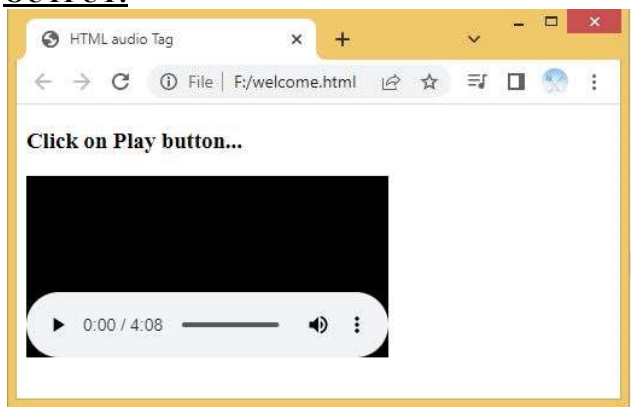
EMBEDDED element:

- HTML allows you to embed plug-ins in a web page using the EMBEDE element.
- The attributes are scr, width, height, and type.

Example: Design web page that demonstrate the EMBEDED element to add audio file.

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML audio Tag</title>
  </head>
  <body>
    <h3>Click on Play button...</h3>
    <embed  src = "Komuram.mp3"></embed>
  </body>
</html>
```

OUTPUT:



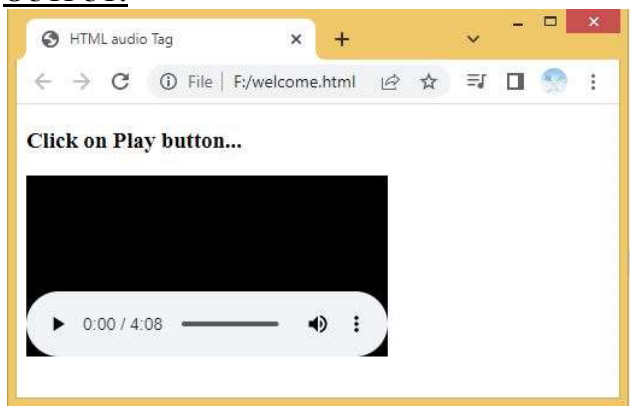
OBJECT element:

- HTML uses the OBJECT element to include objects, such as images, audios, videos ect.
- The attributes are data (specify the object), width, height, type, name, usemap, and form.

Example: Design web page that demonstrate the OBJECT element to add audio file.

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML audio Tag</title>
  </head>
  <body>
    <h3>Click on Play button...</h3>
    <object  data = "Komuram.mp3"></object>
  </body>
</html>
```

OUTPUT:



One mark Questions:

1. What is HTML?
2. What are the features of HTML5?
3. Give the Basic Structure of an HTML5 program.
4. Syntax for character entity.
5. List out the various Empty tags in HTML5
6. Differentiate the <br> and <wbr> tags.
7. List and define any 4 physical formatting tags.
8. List and define any 4 logical formatting tags.
9. Define ruby text.
10. Write the advantage of preformatted text.
11. Write the purpose of DIV element.
12. What is hyperlink and write it purpose with syntax.
13. What is the use of LINK element?
14. Define row spanning and column spanning.
15. Define frame.
16. Write the syntax for IMG element.
17. What is image map and write its syntax.
18. List the various ways of applying the colors in HTML.
19. Define form and write its syntax.
20. List and define attributes of FORM element.
21. What is the use of DATALIST element?
22. Is it possible to select multiple options in dropdown list? Justify your answer.

Essay Questions:

1. Explain structure of HTML with an example.
2. Explain different formatting tags in HTML with an example.
3. List and write a short note on ways of organizing text in HTML.
4. Describe different types of list used in HTML with an example.
5. What is hyperlink? Explain href, target, and id attributes with an example.
6. Describe various table elements and attributes with an example.
7. Explain in detail about INPUT element and their attributes in HTML with an example.

Programming Exercises:

1. Design a web page that displays multiple frames in window.
2. Design a web page that demonstrates the ID attribute in hyperlink.
3. Design a web page that demonstrates the image map
4. Design a web page that demonstrates the TABLE element.
5. Design a web page that demonstrates the nested TABLE element.
6. Create an HTML page that designs the following form.
   Form having at least one text, number, radio, checkbox, dropdown, file, textarea, date input fields.