



WEB TECH CLA1

🕒 Created @February 16, 2025 11:10 AM

1. a) Explain the purpose of text formatting tags.
b) Write a html program to get the following output using various text formatting tags.
Article Title (Bold) This is an HTML tutorial where we cover bold text and strong emphasis. Use italicized text or emphasized text to highlight ideas. This is an underlined word in HTML. Old price: \$100, New price: \$80. H₂O represents water, and $x^2 + y^2 = r^2$ is a circle equation. Here is a sample code: `print("Hello, World!")`

Ans)

a) Purpose of Text Formatting Tags in HTML:

Text formatting tags in HTML are used to enhance the appearance of text by applying styles such as bold, italics, underlining, superscripts, and subscripts. These tags help to structure and highlight important content, making web pages more readable and visually appealing. Some common formatting tags include:

- `` and `` for bold text.

- `<i>` and `` for italicized text.
- `<u>` for underlined text.
- `<sub>` for subscript text (e.g., H₂O).
- `<sup>` for superscript text (e.g., x²).
- `` for strikethrough text (e.g., ~~Old price~~).
- `<code>` for displaying code snippets.
- `<mark>` : Highlights text with a yellow background.
- `<small>` : Reduces the font size of the text.
- `<ins>` : Represents inserted text, usually underlined.
- `<abbr>` : Defines an abbreviation or acronym, showing a tooltip when hovered.
- `<blockquote>` : Represents a block quotation.
- `<cite>` : Cites a reference to another source.
- `<kbd>` : Represents keyboard input (useful for showing key presses).
- `<pre>` : Preserves whitespace and displays text in a monospaced font.
- `<s>` : Represents strikethrough text (similar to `` but for stylistic purposes).
- `<dfn>` : Highlights a term that is being defined.

b) HTML Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Text Formatting </title>
</head>
<body>
  <h2><b>Article Title</b></h2>
```

```

<p>This is an HTML tutorial where we cover <b>bold text</b> and <strong>st
<p>Use <i>italicized text</i> or <em>emphasized text</em> to highlight idea
<p>This is an <u>underlined word</u> in HTML.</p>
<p>Old price: <s>$100</s>, New price: <b>$80</b>.</p>
<p>H<sub>2</sub>O represents water, and x<sup>2</sup> + y<sup>2</sup>
<p>Here is a sample code: <code>print("Hello, World!")</code></p>
</body>
</html>

```

2.a) Create an HTML page that displays an image using the tag. Ensure that:

- The image is responsive and scales according to different screen sizes.
- If the image is not available, an alternate text is displayed.
- When a user hovers over the image, a tooltip appears.

Hint: Use attributes like src, alt, title, and CSS properties like max-width and height.

b) Audio Tag (<audio>) - Custom Audio Player

Question:

Embed an audio file in an HTML page using the <audio> tag.

- The audio file should play automatically when the page loads (without sound).
- Provide controls for play, pause, and volume.
-
- Add a fallback message for browsers that do not support the <audio> tag.

Hint: Use attributes like autoplay, muted, controls, and preload.

c) Question:

Create a webpage that embeds a video using the <video> tag and:

- Supports multiple formats (MP4, WebM, Ogg) to ensure browser compatibility.
- Includes controls for play, pause, and fullscreen.
- Displays a custom thumbnail before playing.
- Shows an error message if the video format is not supported.

Hint: Use <source> tags inside <video>, and attributes like poster, controls, and fallback text.

a) HTML Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Responsive Image Example</title>
  <style>
    /* to make the image responsive */
    img {
      max-width: 100%;
      height: auto;
      display: block;
      margin: auto; /* Centering the image */
    }
  </style>
</head>
<body>

  <h2>Responsive Image Example</h2>
```

```


</body>
</html>
```

b) HTML Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Custom Audio Player</title>
</head>
<body>

  <h2>Embedded Audio Player</h2>

  <audio controls autoplay muted preload="auto">
    <source src="Luther.mpeg" type="audio/mpeg">
    <source src="audio-file.ogg" type="audio/ogg">
    Your browser does not support the audio element.
  </audio>

</body>
</html>
```

c) HTML Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Embedded Video Player</title>
</head>
<body>

<h2>Embedded Video Player</h2>

<video controls width="640" height="360" poster="blair.jpeg">
  <source src="Squid game.mp4" type="video/mp4">
  <source src="video.webm" type="video/webm">
  <source src="video.ogv" type="video/ogg">
  Your browser does not support the video tag.
</video>

</body>
</html>

```

3. **Compare the three types of CSS (inline, internal, external). Analyze their advantages and disadvantages in different scenarios using suitable example for each.**

Comparison of Three Types of CSS: Inline, Internal, and External

CSS (Cascading Style Sheets) can be applied to an HTML document in three different ways: **Inline, Internal, and External**. Each method has its advantages and disadvantages depending on the use case.

1) Inline CSS

Inline CSS is applied directly to an HTML element using the `style` attribute.

Example:

`<p style="color: blue; font-size: 18px;">This is a paragraph with inline CSS.</p>`

Advantages:

- ✓ Useful for quick styling changes.
- ✓ No need for an external file, making it easy to apply CSS to a single element.
- ✓ Overrides internal and external styles due to higher specificity.

Disadvantages:

- ✗ Not reusable—must be written separately for each element.
- ✗ Makes HTML code messy and harder to maintain.
- ✗ Not ideal for large projects due to poor organization.

Best Use Case:

Inline CSS is best for quick fixes or when modifying a small section without affecting other elements.

2) Internal CSS

Internal CSS is written inside a `<style>` tag in the `<head>` section of an HTML document.

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Internal CSS Example</title>
<style>
p {
color: green;
font-size: 20px;
}
</style>
```

```
</head>
<body>
<p>This paragraph uses internal CSS.</p>
</body>
</html>
```

Advantages:

- ✓ Useful for styling a single HTML page.
- ✓ Keeps CSS separate from content while still being in the same file.
- ✓ Allows class and ID selectors for better styling control.

Disadvantages:

- ✗ Not reusable across multiple pages.
- ✗ Can make the HTML file large if there are too many styles.
- ✗ Does not provide the best performance compared to external CSS.

Best Use Case:

Internal CSS is best for single-page applications or when unique styles are needed for a specific HTML page.

3) External CSS

External CSS is written in a separate `.css` file and linked to the HTML document using the `<link>` tag.

Example:

style.css (External CSS file)

```
p {
color: red;
font-size: 22px;
}
```

index.html (HTML File)


```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>External CSS Example</title>
<link rel="stylesheet" href="style.css">
</head>
<body>
<p>This paragraph uses external CSS.</p>
</body>
</html>
```

Advantages:

- ✓ Best for maintaining a consistent style across multiple pages.
- ✓ Keeps HTML clean and readable.
- ✓ Improves page load speed due to caching.

Disadvantages:

- ✗ Requires an extra file, increasing HTTP requests.
- ✗ Styles will not be applied if the CSS file is missing or the link is broken.
- ✗ Debugging can be difficult when multiple stylesheets are involved.

Best Use Case:

External CSS is best for large projects with multiple pages, as it ensures consistency and easier maintenance.

Conclusion:

- **Use inline CSS** for quick fixes or small, one-time changes.
- **Use internal CSS** when styling a single HTML page.
- **Use external CSS** for large-scale projects requiring consistent styling across multiple pages.

4. **Analyze the role of CSS selectors like Class and ID selectors in targeting elements for styling. Give examples to demonstrate Class and ID selectors.**

Role of CSS Selectors: Class and ID Selectors

CSS **selectors** are used to target specific HTML elements and apply styles to them. Among the different types of CSS selectors, **Class (.)** and **ID (#)** selectors play a crucial role in selecting and styling elements efficiently.

1) Class Selector (.)

The **class selector** targets multiple elements that share the same class name. It is defined using a dot (.) before the class name in CSS.

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Class Selector Example</title>
<style>
.highlight {
color: blue;
font-weight: bold;
font-size: 20px;
}
</style>
</head>
<body>

<p class="highlight">This paragraph is styled using a class selector.</p>
<p class="highlight">This is another paragraph with the same class applied.</p>
```

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

Key Features of Class Selectors:

- ✓ Can be applied to **multiple** elements.
 - ✓ Helps in **grouping** elements with the same styling.
 - ✓ Increases **code reusability** and makes styling more efficient.
-

2) ID Selector (#)

The **ID selector** targets a single, unique element using a # symbol before the ID name in CSS. Each ID must be unique within a webpage.

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>ID Selector Example</title>
<style>
#main-heading {
color: red;
font-size: 24px;
text-align: center;
}
</style>
</head>
<body>

<h1 id="main-heading">This is a heading styled using an ID selector.</h1>

</body>
</html>
```

Key Features of ID Selectors:

- ✓ Used to style a **single** unique element.
- ✓ **Higher specificity** than class selectors.
- ✓ Useful for **targeting specific elements** that need unique styling.

Conclusion:

- **Use class selectors (.)** when you need to style multiple elements.
- **Use ID selectors (#)** when you need to style a single, unique element.
- **Combine both** when necessary to achieve efficient and organized styling.

5. Explain the importance of responsive web design. How do CSS features like media queries contribute to creating adaptable layouts?

Importance of Responsive Web Design (RWD)

What is Responsive Web Design?

Responsive Web Design (RWD) ensures websites **adapt to different screen sizes**, providing a **better user experience (UX)**, **improved SEO**, and **reduced maintenance effort**. It enhances usability on **desktops, tablets, and mobile devices** without needing multiple site versions.

Role of CSS Features Like Media Queries in Responsive Design

What are Media Queries?

CSS **media queries** allow developers to apply different styles based on the screen width, height, resolution, or device type. This helps create adaptable layouts for various screen sizes.

Example of Media Queries:

```
/* Default styles for larger screens */
body {
font-size: 18px;
background-color: lightblue;
}

/* Styles for tablets (width up to 768px) */
@media (max-width: 768px) {
body {
font-size: 16px;
background-color: lightgreen;
}
}

/* Styles for mobile phones (width up to 480px) */
@media (max-width: 480px) {
body {
font-size: 14px;
background-color: lightcoral;
}
}

/* Default styles for larger screens */
body {
font-size: 18px;
background-color: lightblue;
}

/* Styles for tablets (width up to 768px) */
@media (max-width: 768px) {
body {
font-size: 16px;
background-color: lightgreen;
}
}

/* Styles for mobile phones (width up to 480px) */
@media (max-width: 480px) {
```

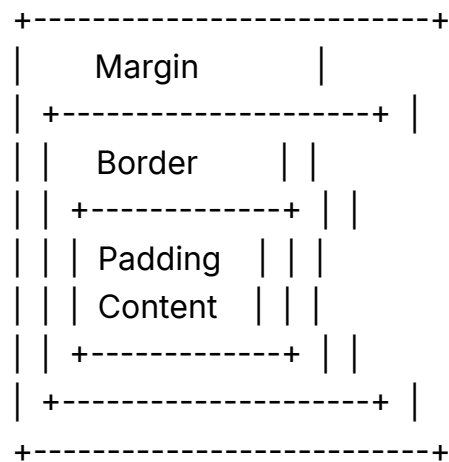
```
body {
font-size: 14px;
background-color: lightcoral;
}
}
```

6. Discuss the importance of box model properties such as margin, border, padding, and width in layout design.

The **CSS Box Model** controls element spacing and layout using **margin, border, padding, and width**.

- **Padding** – Space **inside** the border, prevents text from touching edges.
- **Border** – Defines the **element's boundary**, can be styled.
- **Margin** – Space **outside** the border, controls element positioning.
- **Width/Height** – Sets **content size**, helps with responsive design.

Visual Representation of Box Model:



Why It Matters?

- ✓ Organizes layout & spacing.
- ✓ Improves readability & aesthetics.
- ✓ Essential for responsive design.

Example:

```
div {  
width: 80%;  
padding: 10px;  
border: 2px solid black;  
margin: 20px;  
}
```

7) Use appropriate html list tags to display the following output.

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>HTML List Example</title>  
</head>  
<body>  
  
  <h2>Steps to Prepare a Sandwich</h2>  
  <ol>  
    <li>Gather Ingredients  
      <ol type="a">  
        <li>Bread</li>  
        <li>Butter</li>  
        <li>Vegetables (Tomato, Lettuce, Cucumber)</li>  
      </ol>  
    </li>  
    <li>Spread butter on the bread</li>  
    <li>Add vegetables</li>  
    <li>Close the sandwich</li>  
    <li>Serve and enjoy</li>  
  </ol>  
  
  <h2>Daily To-Do List</h2>
```

```

<ul>
  <li>Morning Tasks
    <ul>
      <li>Wake up</li>
      <li>Exercise</li>
      <li>Have breakfast</li>
    </ul>
  </li>
  <li>Work Tasks
    <ul>
      <li>Attend meeting</li>
      <li>Complete project report</li>
    </ul>
  </li>
  <li>Evening Tasks
    <ul>
      <li>Read a book</li>
      <li>Go for a walk</li>
    </ul>
  </li>
</ul>

<h2>Glossary</h2>
<dl>
  <dt><strong>HTML</strong></dt>
  <dd>A markup language for structuring web content.</dd>

  <dt><strong>CSS</strong></dt>
  <dd>A stylesheet language used for designing web pages.</dd>

  <dt><strong>JavaScript</strong></dt>
  <dd>A programming language used to add interactivity to websites.</dd>
</dl>

```



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

8. Create an HTML webpage to demonstrate the usage of the target attribute in hyperlinks. Your page should include:

- 1. A text link that opens "<https://www.example.com>" in the same tab (_self).**
- 2. A text link that opens "<https://www.google.com>" in a new tab (_blank).**
- 3. A text link that opens "<https://www.wikipedia.org>" in a new window (_blank).**
- 4. A text link that loads "<https://www.youtube.com>" inside an iframe on the same page (use a named target).**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Hyperlink Target Attribute Demo</title>
</head>
<body>

  <h2>Demonstrating the Target Attribute in Hyperlinks</h2>

  <!-- Link that opens in the same tab →
  <p><a href="https://www.example.com" target="_self">Open Example.com in

  <!-- Link that opens in a new tab →
  <p><a href="https://www.google.com" target="_blank">Open Google in a new

  <!-- Link that opens in a new window (same as _blank in most cases) →
```

```

<p><a href="https://www.wikipedia.org" target="_blank">Open Wikipedia in a

<!-- Named target for iframe →
<p><a href="https://www.youtube.com" target="youtubeFrame">Load YouTu

<!-- Iframe for loading YouTube →
<iframe name="youtubeFrame" width="600" height="400" style="border: 1px

</body>
</html>

```

9) Design the table using HTML as shown below:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Sales Data Table</title>
  <style>
    table {
      width: 60%;
      border-collapse: collapse;
      margin: 20px auto;
      text-align: center;
    }
    th, td {
      border: 1px solid black;
      padding: 10px;
    }
    th {
      background-color: #f2f2f2;
    }
    .bold {

```

```

        font-weight: bold;
    }
</style>
</head>
<body>

<h2 style="text-align: center;">Sales Data Table</h2>

<table>
    <tr>
        <th rowspan="2">Product</th>
        <th colspan="3">January</th>
        <th colspan="3">February</th>
        <th rowspan="2">Total Sales</th>
    </tr>
    <tr>
        <th>Week 1</th>
        <th>Week 2</th>
        <th>Week 3</th>
        <th>Week 1</th>
        <th>Week 2</th>
        <th>Week 3</th>
    </tr>
    <tr>
        <td rowspan="2">Laptop</td>
        <td>150</td>
        <td>200</td>
        <td>180</td>
        <td>220</td>
        <td>250</td>
        <td>230</td>
        <td rowspan="2">1230</td>
    </tr>
    <tr>
        <td>130</td>
        <td>190</td>

```

```

        <td>170</td>
        <td>210</td>
        <td>240</td>
        <td>220</td>
    </tr>
    <tr>
        <td>Tablet</td>
        <td colspan="2">140</td>
        <td>160</td>
        <td>180</td>
        <td>200</td>
        <td>190</td>
        <td>870</td>
    </tr>
    <tr>
        <td rowspan="2">Smartphone</td>
        <td>300</td>
        <td>320</td>
        <td>310</td>
        <td>330</td>
        <td>350</td>
        <td>340</td>
        <td rowspan="2">1950</td>
    </tr>
    <tr>
        <td>280</td>
        <td>310</td>
        <td>290</td>
        <td>320</td>
        <td>340</td>
        <td>330</td>
    </tr>
    <tr class="bold">
        <td colspan="7">Grand Total</td>
        <td>4050</td>
    </tr>

```

```
</table>

</body>
</html>
```

11. Design the HTML form as shown below:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Registration Form</title>
  <style>
    body {
      font-family: Arial, sans-serif;
    }
    table {
      width: 60%;
      border-collapse: collapse;
      margin: 20px auto;
    }
    td {
      padding: 10px;
    }
    fieldset {
      border: 1px solid black;
      padding: 10px;
      margin-bottom: 15px;
    }
    legend {
      font-weight: bold;
    }
  </style>
```

```

</head>
<body>

<h2>Registration Form</h2>

<form>

  <fieldset>
    <legend>Personal Information</legend>
    <table>
      <tr>
        <td>First Name:</td>
        <td><input type="text" name="firstname"></td>
      </tr>
      <tr>
        <td>Last Name:</td>
        <td><input type="text" name="lastname"></td>
      </tr>
      <tr>
        <td>Gender:</td>
        <td>
          <input type="radio" name="gender" value="male"> Male
          <input type="radio" name="gender" value="female"> Female
          <input type="radio" name="gender" value="other"> Other
        </td>
      </tr>
      <tr>
        <td>Date of Birth:</td>
        <td><input type="date" name="dob"></td>
      </tr>
      <tr>
        <td>Upload Photo:</td>
        <td><input type="file" name="photo"></td>
      </tr>
    </table>
  </fieldset>

```

```

<fieldset>
  <legend>Contact Details</legend>
  <table>
    <tr>
      <td>Email:</td>
      <td><input type="email" name="email"></td>
    </tr>
    <tr>
      <td>Phone Number:</td>
      <td><input type="tel" name="phone"></td>
    </tr>
    <tr>
      <td>Address:</td>
      <td><textarea name="address"></textarea></td>
    </tr>
    <tr>
      <td>Country:</td>
      <td>
        <select name="country">
          <option>Select</option>
          <option>USA</option>
          <option>UK</option>
          <option>India</option>
          <option>Australia</option>
        </select>
      </td>
    </tr>
  </table>
</fieldset>

<fieldset>
  <legend>Account Preferences</legend>
  <table>
    <tr>
      <td>Username:</td>

```

```

        <td><input type="text" name="username"></td>
    </tr>
    <tr>
        <td>Password:</td>
        <td><input type="password" name="password"></td>
    </tr>
    <tr>
        <td>Confirm Password:</td>
        <td><input type="password" name="confirm_password"></td>
    </tr>
    <tr>
        <td>Subscription Preferences:</td>
        <td>
            <input type="checkbox" name="newsletter"> Subscribe to Newsle
            <input type="checkbox" name="offers"> Receive Offers
        </td>
    </tr>
</table>
</fieldset>

<fieldset>
    <legend>Other Information</legend>
    <table>
        <tr>
            <td>Favorite Color:</td>
            <td><input type="color" name="color"></td>
        </tr>
        <tr>
            <td>Personal Website:</td>
            <td><input type="url" name="website"></td>
        </tr>
        <tr>
            <td>Rate Us:</td>
            <td><input type="range" name="rating" min="1" max="5"></td>
        </tr>
    </table>

```



```

        <tr>
        <td>Preferred Contact Time:</td>
        <td><input type="time" name="contact_time"></td>
        </tr>
    </table>
</fieldset>

<fieldset>
    <legend>Submit</legend>
    <div>
        <input type="submit" value="Register">
        <input type="reset" value="Reset">
    </div>
</fieldset>
</form>

</body>
</html>

```

12. Design the HTML table :

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Student Marks Table</title>
    <style>

        table, th, tr, td{
            border: 1px solid black;
        }

    </style>

```

```

</head>
<body>

<h2>Student Mark Table</h2>
<table cellspacing="5px" cellpadding="5px">
  <tr>
    <th>Student Name</th>
    <th>Subject</th>
    <th>Marks</th>
  </tr>
  <tr>
    <td>Alice</td>
    <td>Math</td>
    <td>85</td>
  </tr>
  <tr>
    <td>Bob</td>
    <td>Science</td>
    <td>90</td>
  </tr>
  <tr>
    <td>Charlie</td>
    <td>English</td>
    <td>88</td>
  </tr>
</table>

</body>
</html>

```

13. Design the form as shown below:

```

<!DOCTYPE html>
<html lang="en">

```

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Basic Registration Form</title>
  <style>

    body {
      align-items: center;
      padding: 20px;
      width: 70%;
    }
    h2 {
      text-align: center;
    }
    label {
      display: block;
      margin-top: 10px;
      font-weight: bold;
    }

    input, select, textarea {
      width: 100%;
      padding: 8px;
      margin-top: 5px;
      border: 1px solid #ccc;
      border-radius: 5px;
    }

    button {
      margin-top: 15px;
      padding: 10px;
      width: 48%;
      border: none;
      border-radius: 5px;
      cursor: pointer;
    }
  </style>
</head>
```

```

</style>
</style>
</head>
<body>

<h2><b>Basic Registration Form</b></h2>

<form>
  <label for="name">Name:</label>
  <input type="text" id="name" name="name">

  <label for="email">Email:</label>
  <input type="email" id="email" name="email">

  <label for="age">Age:</label>
  <input type="number" id="age" name="age">

  <label for="gender">Gender:</label>
  <select id="gender" name="gender">
    <option>Male</option>
    <option>Female</option>
    <option>Other</option>
  </select>

  <label for="message">Message:</label>
  <textarea id="message" name="message"></textarea>

  <div class="buttons">
    <input type="submit" value="Submit">
    <input type="reset" value="Reset">
  </div>
</form>

```

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

14) Differentiate <iframe>, <form>, <link> and <a> tags in HTML 5 with examples.

Differences Between <iframe>, <form>, <link>, and <a> Tags in HTML5

1) <iframe> (Inline Frame)

- Used to **embed another webpage** within the current page.

Example:

```
<iframe src="https://example.com" width="300" height="200"></iframe>
```

2) <form> (Form for User Input)

- Defines a form for **user input and data submission**.

Example:

```
<form action="/submit" method="POST">  
<input type="text" name="name" placeholder="Enter your name">  
<button type="submit">Submit</button>  
</form>
```

3) <link> (External Resource Linking)

- Links to **external stylesheets, icons, or resources**.

Example:

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

4) <a> (Anchor Link for Navigation)

- Creates **hyperlinks** to navigate between web pages.

Example:

`Visit Example`

Tag	Purpose	Example
<code><iframe></code>	Embeds another webpage within the current page.	<code><iframe src="https://example.com" width="300" height="200"></iframe></code>
<code><form></code>	Creates an input form for user data submission.	<code><form action="/submit" method="POST"><input type="text" name="name"><button type="submit">Submit</button></form></code>
<code><link></code>	Links external resources like CSS or icons.	<code><link rel="stylesheet" href="styles.css"></code>
<code><a></code>	Creates hyperlinks to navigate between pages.	<code>Visit Example</code>

15. Demonstrate on CSS link properties, background properties and animation properties with examples.

CSS Link, Background, and Animation Properties

1) CSS Link Properties

Used to style hyperlinks in different states (`normal` , `hover` , `active` , `visited`).

Example:

```
a {
color: blue;
text-decoration: none;
}
a:hover {
color: red;
text-decoration: underline;
}
a:active {
```

```
color: green;
}
a:visited {
color: purple;
}

<a href="https://example.com">Hover over this link</a>
```

2) CSS Background Properties

Control the background color, image, position, and more.

Example:

```
body {
background-color: lightblue;
background-image: url('background.jpg');
background-size: cover;
background-repeat: no-repeat;
}
```

3) CSS Animation Properties

Used to create smooth animations for elements.

Example:

```
@keyframes slide {
from { transform: translateX(0); }
to { transform: translateX(100px); }
}

.box {
width: 100px;
height: 100px;
background-color: red;
animation: slide 2s infinite alternate;
}

<div class="box"></div>
```

Summary Table

Property Type	Description	Example Property	Example Usage
Link Properties	Styles hyperlinks in different states.	<code>color</code> , <code>text-decoration</code> , <code>:hover</code> , <code>:visited</code>	<code>a:hover { color: red; }</code>
Background Properties	Controls background color, images, size, and positioning.	<code>background-color</code> , <code>background-image</code> , <code>background-size</code>	<code>background-image: url('bg.jpg');</code>
Animation Properties	Creates animations using keyframes.	<code>@keyframes</code> , <code>animation</code> , <code>animation-duration</code>	<code>animation: slide 2s infinite;</code>