**✅ Day 4.1 – Inline, Block & HTML Tags (Deep Notes)**

**🧾 What I Learned Today**

Today I focused on understanding different types of HTML elements, especially the **difference between inline and block elements**, and how they behave inside the HTML document.

**🔹 HTML Code Used:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Inline</title>

<link rel="stylesheet" href="style\_Day\_4\_1.css">

</head>

<body>

<p>Hey I am a para</p>

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

<div>I am a block</div>

<span>I am a span and a inline</span>

<a href="">Inline</a>

</body>

</html>

**🔍 Detailed Explanation of Concepts:**

**1. ✅ <!DOCTYPE html>**

* This declaration defines the **document type** and version of HTML.
* Here it declares that we are using **HTML5**.

**2. ✅ <html lang="en">**

* The root element of the HTML document.
* lang="en" specifies the **language** of the page content is English.

**3. ✅ <head> Section**

* Contains **metadata** like:
  + Character encoding: <meta charset="UTF-8">
  + Viewport settings for responsiveness: <meta name="viewport" content="width=device-width, initial-scale=1.0">
  + Page title: <title>Inline</title>
  + CSS Link: <link rel="stylesheet" href="style\_Day\_4\_1.css">

**4. ✅ <body> Section**

* The **visible content** of the webpage.

**🧱 Block vs Inline Elements**

**🔸 <p> (Paragraph Tag)**

* **Block-level element**
* Starts on a **new line**
* Takes **full width** of the container
* Example:
* <p>Hey I am a para</p>

**🔸 <div> (Division)**

* Generic **block-level container**
* Used for **grouping** elements for styling or scripting
* Starts on a **new line**
* Example:
* <div>I am a block</div>

**🔸 <span> (Span)**

* Generic **inline-level element**
* Does **not** start on a new line
* Used for styling small parts of text
* Example:
* <span>I am a span and a inline</span>

**🔸 <a href=""> (Anchor Tag)**

* **Inline element**
* Used for **hyperlinks**
* Does not break into a new line
* Example:
* <a href="https://google.com">google</a>
* <a href="">Inline</a>

**🎨 Linking External CSS**

<link rel="stylesheet" href="style\_Day\_4\_1.css">

* Connects the HTML file to a **separate CSS file**
* Makes styling easier and cleaner

**💡 Key Takeaways:**

* Block elements take full width and start on a new line.
* Inline elements stay within the same line and take only as much width as needed.
* Use <div> for structure and grouping.
* Use <span> for inline text styling.
* External CSS keeps your HTML neat and maintainable.

**📘 Assignment Ideas:**

* Try creating a page with both inline and block elements.
* Style your elements in style\_Day\_4\_1.css with borders to visualize block vs inline behavior.
* Try using display: inline, block, and inline-block in CSS to convert elements.

**✅ Day 4.2 – Understanding id, class, and Styling in HTML**

**🧾 What I Learned Today**

Today, I learned how to use **id** and **class** attributes in HTML to **identify** and **style elements**. I also practiced combining multiple classes and connecting CSS using an external file.

**🔹 HTML Code I Wrote:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>ID and Classes</title>

<link rel="stylesheet" href="style\_Day\_4\_2.css">

</head>

<body>

<div id="firstdiv" class="red bg-yellow">First</div>

<div id="seconddiv" class="red bg-yellow">Second</div>

<span class="red">that is a inline</span>

</body>

</html>

**🔍 Detailed Explanation of Key Concepts**

**1. ✅ id Attribute**

* **Unique identifier** for a single HTML element.
* Each id **must be unique** within a page.
* Typically used for **targeting a specific element** with CSS or JavaScript.

Example:

<div id="firstdiv">First</div>

🔹 In CSS:

#firstdiv {

color: blue;

}

**2. ✅ class Attribute**

* **Reusable identifier** used to apply styles to **multiple elements**.
* One element can have **multiple classes** separated by space.

Example:

<div class="red bg-yellow">Second</div>

🔹 In CSS:

.red {

color: red;

}

.bg-yellow {

background-color: yellow;

}

**3. ✅ Using Both id and class Together**

* You can use id to identify an element **uniquely**, and class to **share styling** with others.

<div id="seconddiv" class="red bg-yellow">Second</div>

**4. ✅ <span class="red">that is a inline</span>**

* The <span> is an **inline element**.
* Applying the class red changes the text color.

**🎨 CSS File Link**

<link rel="stylesheet" href="style\_Day\_4\_2.css">

* Connects the page to a separate **CSS file**.
* Keeps structure and style **separated** and clean.

**🖌️ Visual Example (CSS Idea for style\_Day\_4\_2.css):**

Here’s how the CSS file might look:

/\* style\_Day\_4\_2.css \*/

.red {

color: red;

font-weight: bold;

}

.bg-yellow {

background-color: yellow;

padding: 10px;

}

**💡 Key Takeaways:**

* Use id for **unique** elements.
* Use class for **reusable** styles.
* You can apply **multiple classes** to a single element.
* External CSS keeps your HTML organized and easier to manage.
* <span> is inline; <div> is block-level.

**📘 Practice Ideas:**

* Try making 3 divs with different ids and shared class.
* Style each with background, padding, and font.
* Add borders to see how block vs inline behaves visually.

**✅ Day 4.3 – Audio, Video, SVG & iframe in HTML**

**🧾 What I Learned Today**

Today I learned how to embed **multimedia** elements in HTML including **videos, audio tracks, vector graphics (SVG), and YouTube videos using iframe**. These tags are essential for creating rich, interactive, and engaging web content.

**🔹 HTML Code I Used:**

<video src="matrix\_-\_47802 (1440p).mp4" height="355" controls poster="20230623\_160558.jpg"></video>

<!-- <video src="matrix\_-\_47802 (1440p).mp4" controls autoplay></video> -->

<audio src="Tomar Jonno Nilche Tara Arnob female cover by Barisha Khan @musiclovers4027.mp3" controls></audio>

<svg height="100" width="100">

<circle cx="50" r="40" stroke="black" stroke-width="3" fill="red" >

</svg>

<iframe width="560" height="315" src="https://www.youtube.com/embed/XZwBNDGuWGU?si=SInENISzXTz-4CsS&amp;start=794" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share" referrerpolicy="strict-origin-when-cross-origin" allowfullscreen></iframe>

**🎥 1. <video> Tag**

* Used to embed **video files** directly into the webpage.
* Common attributes:
  + src – video file path
  + controls – adds play, pause, volume controls
  + height / width – sets video dimensions
  + poster – displays an image before video plays
  + autoplay – starts playing automatically (use with caution!)

✅ Example:

<video src="video.mp4" height="355" controls poster="thumbnail.jpg"></video>

⚠️ Note: Some browsers restrict autoplay unless muted.

**🔊 2. <audio> Tag**

* Used to embed **audio** (music, speech) into a web page.
* Common attributes:
  + src – audio file path
  + controls – shows audio player (play, pause)
  + autoplay, loop, muted – additional behaviors

✅ Example:

<audio src="song.mp3" controls></audio>

**🖼️ 3. <svg> (Scalable Vector Graphics)**

* Used to create **shapes and drawings** using XML syntax.
* It’s **resolution independent**, perfect for icons and diagrams.
* <circle> attributes:
  + cx – x-position of the center
  + r – radius
  + stroke, stroke-width – outline color and thickness
  + fill – inside color

✅ Example:

<svg height="100" width="100">

<circle cx="50" r="40" stroke="black" stroke-width="3" fill="red" />

</svg>

**📺 4. <iframe> Tag**

* Used to embed **external web content**, like YouTube videos, maps, websites.
* Common attributes:
  + src – URL of the content to embed
  + width, height – dimensions
  + frameborder – border (usually set to 0)
  + allowfullscreen – lets the content be viewed full screen
  + allow – allows features like autoplay, encrypted media, etc.

✅ Example:

<iframe width="560" height="315"

src="https://www.youtube.com/embed/VIDEO\_ID"

frameborder="0"

allowfullscreen>

</iframe>

📝 Your src also includes a start=794 parameter which tells the video to begin at 794 seconds (~13:14).

**💡 Key Takeaways**

| **Tag** | **Purpose** | **Key Attributes** |
| --- | --- | --- |
| <video> | Embed video | src, controls, poster, autoplay |
| <audio> | Embed audio | src, controls, loop, autoplay |
| <svg> | Vector graphics (shapes) | cx, r, fill, stroke |
| <iframe> | Embed YouTube/webpages | src, allowfullscreen, allow |

**🎨 Optional CSS Ideas (style\_Day\_4\_3.css):**

video, audio, iframe {

display: block;

margin: 20px auto;

border: 2px solid #333;

border-radius: 10px;

}

svg {

display: block;

margin: 20px auto;

}

**📘 Practice Ideas**

* Try using <source> tag inside <video> or <audio> for multiple formats.
* Create your own SVG shapes like rectangle, ellipse, or line.
* Embed a Google Map using <iframe>.