



Introduction to

HTML

Lecture Flow



- *What is HTML?*
- *Tags, Elements, and Attributes*
- *Block vs. Inline Elements*
- *Essential Tags*
- *Q&A*

HTML

HyperText Markup Language

What is HTML?

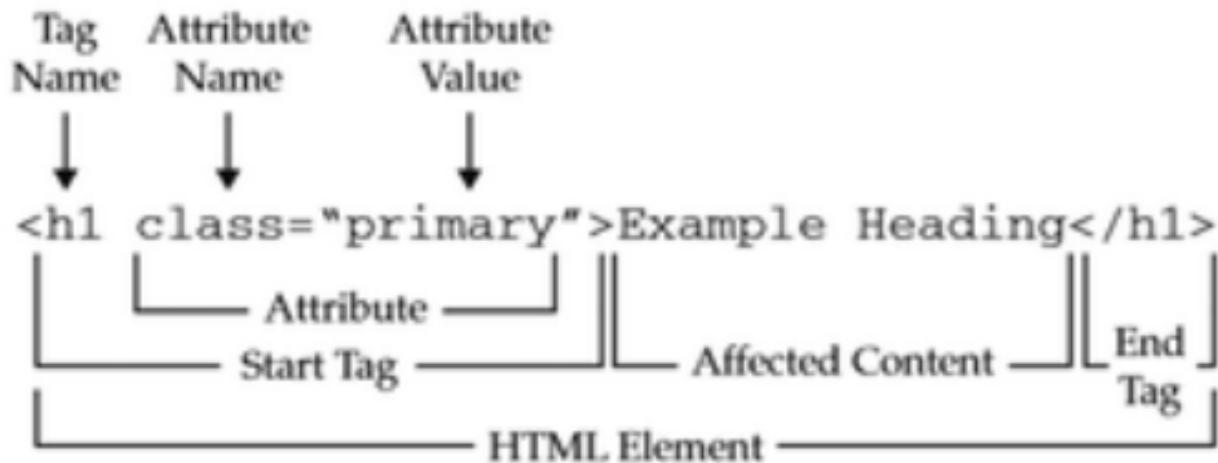
- It's the foundation or structure of every webpage.
- It uses tags to markup (describe) content.

	HTML (HyperText Markup Language)	Programming Language (e.g., Python, JavaScript)
Feature	Structures and describes content. It defines <i>what</i> the content is.	Executes actions and implements logic. It defines <i>how</i> a program should behave
Core Function	Presentation and Organization. It acts like the skeleton or blueprint of a web page.	Computation and Automation. It's the brain that makes decisions and performs tasks.
Key Characteristics	Uses tags (<code><h1></code> , <code><p></code> , <code><a></code>) to wrap and categorize static content.	Uses functions , variables , conditionals (<code>if/else</code>), and loops (<code>for/while</code>).
Example Analogy	The Blueprint of a House: It shows where the walls are, where the kitchen is, and where the doors go.	The Electrician's Wiring: It determines <i>if</i> the light switch is pressed, <i>then</i> the light bulb should turn on.

HTML was initially created by Tim Berners-Lee in 1991 as a way to share and structure documents on the web.



Tags, Elements and Attributes



An **Element** is everything from the opening tag to the closing tag (e.g., <p>...</p>).

A **Tag** is just the opening or closing marker (e.g., <p> or </p>). Tags tell the browser what a piece of content is (e.g., this is a link, this is a paragraph).

eg. <h1>Hello World!</h1>

Attributes: Extra information inside the opening tag that modifies the element (e.g., `link`).

Void/Self-Closing Tags: Tags that don't wrap content and don't have a closing tag (e.g., `
` for a line break).

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>My First Webpage</title>
  </head>
  <body>
    <h1>My very first web page</h1>
  </body>
</html>
```

!Tip Alert

Testing Your HTML in Real Time

- Install Live Server in VS Code.
- Right-click your .html file → “Open with Live Server”.
- The browser refreshes automatically when you save!

!Tip Alert

Extension: Live Server X



Live Server

Ritwick Dey | ⚡ 70,625,874 | ★★★★★(511)

Launch a development local Server with live reload feature for static & dynamic pages

[Disable](#) [Uninstall](#) Auto Update 

[DETAILS](#) [FEATURES](#) [CHANGELOG](#)

Mini-Exercise 1: "Hello File" (5 mins)

Task: Open your code editor and create my-first-page.html.



<!DOCTYPE html>

Tells the browser which version of HTML to use (HTML5). It is mandatory for all HTML documents

<html></html>

The root element, wraps everything.

<head></head>

store the data which actually doesn't appear on the webpage but it gives more information about the webpage. It also contain tags like:

`<title> </title>`: the content appears on the tab when opened by the browser

`<meta>` - information about the page to browsers search engines and social media platforms. It is useful in search engine optimization.

`<link rel ="stylesheet" href="file.css ">`-- to connect css file.

<body></body>

This tag is used to display all the information or data, i.e, text, images, hyperlinks videos, etc., on the webpage to the user.

<!-- comment -->

This tag is used to add comments in the HTML codes. These comments help the program to understand the code. The content inside the comment tag doesn't visible on the browser.

<style>

Embeds internal CSS styles

<script></script>

It is used for including javascript code. The external javascript can also be linked using the src attribute in the opening script tag. It can be included in the head or body tag.

!Tip Alert

- In Visual Studio Code, you can quickly generate an HTML boilerplate by typing ! in a blank HTML file and pressing **Tab or Enter**.
- This expands into the full <!DOCTYPE html> structure with <html>, <head>, and <body> tags – saving you time when starting a new project.

Essential Tags

Heading: HTML provides six types of headings, i.e., H1, H2, H3, H4, H5, and H6. H1 is the highest level heading and H6 is the lower level heading.

Paragraphs (<p>): Used to group blocks of text content.

<p>This is a block of text.</p>

Heading 1 ----- h1

Heading 2 ----- h2

Heading 3 ----- h3

Heading 4 ----- h4

Heading 5 ----- h5

Heading 6 ----- h6

- **Line Break Tag (
)**: Forces a line break within a paragraph or other element. (It is a void tag).
 - eg: *<p>Line one.
Line two starts here.</p>*
- **<hr> (Horizontal Rule)**: Creates a horizontal line to visually separate content and signal a thematic break in the page.
 - eg: *<h1>Section A</h1> <hr> <h1>Section B</h1>*

Mini-Exercise 2: "Layout Guess" (5 mins)

Task: Predict which element takes up the full width: `<h1>` vs. ``. Test it by adding a `<h1>` and `` in your file.

Block vs. Inline Elements

Every HTML element is essentially a box.
These boxes behave in one of two ways:
Block or Inline.

Block-Level Elements

- They always start on a new line.
- They take up the full available width of their container (even if the content is short).
- Common Tags: <h1>, <h2>, <p>, , , , <div> (a general container).

Inline Elements

- They do not start on a new line.
- They only take up as much width as necessary to contain their content.
- Common Tags: <a> (Links), , , (a general container for text).

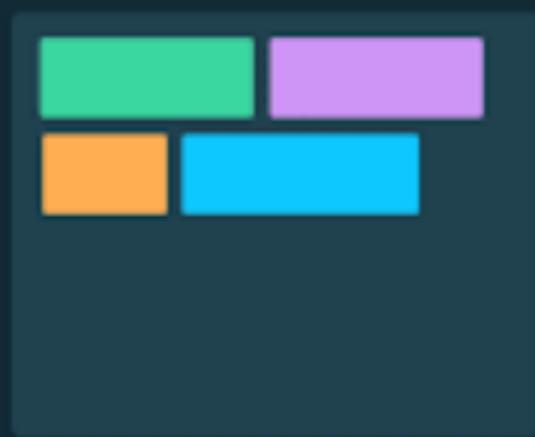
Inline Elements examples

- : Renders text in bold.
- <i>: Renders text in italics.
- <u>: Renders text with an underline.

Difference between block and inline elements



`display: block;`



`display: inline;`

Adding Links

< a > Defines a hyperlink

Key Attribute: href (Hypertext Reference) -
Specifies the destination URL

eg: < a href="#" > About Me </ a >

Type of Links

Absolute: Links to an external website (starts with http:// or https://).

Relative: Links to another page within your own website (e.g.,)

What if we wanted it to
open the link in a new
page?

Use target="_blank". This tells the browser to open the linked URL in a new browser window or tab.

eg: *<a href="https://example.com"
target="_blank">Open Example in New Tab*

List Tags (, ,)

Unordered Lists ():

For lists where the order doesn't matter (uses bullet points).

```
<ul>
    <li>Milk</li>
    <li>Eggs</li>
</ul>
```

Ordered Lists ()

For lists where the order does matter (uses numbers or letters).

```
<ol>
  <li>Step 1</li>
  <li>Step 2</li>
</ol>
```

List Item (): Must be used inside both and .

Images



- <Void tag → no closing tag.
- Attributes:
 - src: image path (required)
 - alt: text shown if image doesn't load (for accessibility)

eg: **

You can control image size using width and height, but it's better to style images with CSS for flexibility and consistency across your site. (you will learn css in the coming weeks)

Media Tags: <video> & <audio>

Used to embed and play media files (like music or videos) directly on a webpage no external player needed.

- Tags:<video> → for videos (.mp4, .webm)
<audio> → for sounds or music (.mp3, .wav)

Common Attributes:

- src → file path or URL to the media
- controls → adds built-in play, pause, and volume buttons
- autoplay → starts automatically when the page loads
- loop → restarts automatically after finishing.

eg: `<video src="videos/intro.mp4" controls></video>`
`<audio src="sounds/theme.mp3" controls loop></audio>`

Embedding External Content

- **<iframe>** - Inline Frame creates a rectangular window (a frame) inside your web page that loads and displays a completely separate HTML document (the external content).
- Controls provided by the external site (e.g., YouTube's branding and controls).

eg:<*iframe src="URL-of-external-content" width="560" height="315"*></*iframe*>

Grouping Tags

`<div></div>`

The most common block-level container.

Used to group large sections of content together (e.g., a header, a column, a card).

It has no semantic meaning on its own; it's purely for structure and styling.

The most common inline-level container.

Used to apply styling or target a small piece of text within a larger element (e.g., highlighting a single word within a paragraph).

It has no semantic meaning on its own.

Mini-Exercise 3: "The Simple CV" (15 mins)

Task: Create a page that includes:

- An **<h1>** for your name.
- A **<p>** for a short bio.
- An **<h2>** titled "Skills."
- An unordered list (****) of 3 skills.
- A link (**<a>**) to your favorite tech website that opens in a new tab.

✿ Meet Chapi!

Golden Retriever Extrateniare

Chapi is playful, energetic Golden Retriever who loves to explore! He enjoys chasing balls, swimming, getting belly rubs & H's very smart **boy** and a loyal * companion*



Skills & Tricks

- Sit & Stay
- Fetch The Ball
- Shake Hands
- Jump High

Chapi's Favorite Things

- 1 Squeaky Toys
- 2 Peanut Butter
- 4 Long Walks
- 6 His Human

Learn more about https://dogbreedinfo.com/target_trail_Click here



Q/A





Resources

- GeeksforGeeks
- HTML Tags
- Learn HTML - video Link
- cheat sheet



Thank You!!!