

LECTURE 3

HTML and HTML5 Fundamentals

The raw ingredients of every website: structure, text, links, images, tables, and lists.

CSC336 Web Technologies

In This Lecture

- 1 Understand the role of markup languages, and the anatomy of an HTML tag
- 2 HTML's document structure, elements, attributes, nesting rules, and void elements
- 3 Headings, text formatting, and why the old `<font>` tag is obsolete
- 4 `<pre>/<code>` and other technical text tags, plus HTML entities
- 5 Hyperlinks (internal vs. external), images (including image links), audio, video
- 6 Tables, all three list types (including nested lists), and what HTML5 added

What Is a Markup Language?

- 1 HyperText — text that contains links to other text or pages
- 2 Markup — the text is annotated with tags describing its structure
- 3 Language — a shared grammar every browser agrees to follow
- 4 HTML is NOT a programming language — no variables, loops, or conditions; it only describes structure and content

Anatomy of an HTML Tag

Part	Example	Meaning
Opening tag	<p>	Marks the start of an element
Attribute	class="intro"	Extra info, written inside the opening tag only
Content	Hello, World!	What sits between the opening and closing tag
Closing tag	</p>	Marks the end — note the forward slash
Comment	<!-- note -->	Ignored by the browser, visible only in the source
Void (self-closing) tag	 	No content, and so no closing tag at all

START HERE

Your First Example: Hello, World!

```
<!-- This is a comment: the browser ignores it -->  
<p class="intro" id="greeting">Hello, World!</p>
```

Hello, World!

HTML Document Structure

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>My First Page</title>
</head>
<body>
  <h1>Hello, Web!</h1>
  <p>This is my first HTML page.</p>
</body>
</html>
```

Hello, Web!

This is my first HTML page.

Void (Self-Closing) Elements

Element	What It Does
<code>
</code>	A single line break within text
<code><hr></code>	A thematic divider line between sections
<code></code>	Embeds an image
<code><input></code>	A form control
<code><meta> / <link></code>	Document metadata / an external resource, used inside <code><head></code>

Void elements have no content and therefore no closing tag — `<div></div>` is NOT void, it just happens to be empty.

Void Elements, Rendered

```
<br>  
<hr>  
  
<input type="text">
```

Text before a line break.
Text after the break.



Block-Level vs. Inline Elements

	Block-Level	Inline
Starts a new line?	Yes	No — flows with content
Full available width?	Yes, by default	No — only as wide as content
Examples	h1-h6, p, div, ul, table	a, strong, em, span, img

Block vs. Inline, Rendered

```
<h1>Page Title</h1>
<p>This is a paragraph.</p>
<p>This is another paragraph, on its own line below.</p>

<p>This word is <strong>important</strong>, and this is a
<a href="#">link</a> in the middle of a sentence.</p>
```

Page Title

This is a paragraph.

This is another paragraph, on its own line below.

This word is **important**, and this is a [link](#) in the middle of a sentence.

Global Attributes

Attribute	Purpose
id	A unique identifier for one specific element
class	Category name(s) shared by many elements, for CSS/JS
title	Extra info shown as a tooltip on hover
style	Inline CSS applied to just this one element
lang	The (human) language of this element's content
data-*	A custom attribute for attaching your own data

Boolean attributes (required, checked, disabled) need no ="value" — present means true, absent means false.

Attributes, Rendered

```
<a href="https://www.comsats.edu.pk" target="_blank">Visit COMSATS</a>
<p>Founded in 1998, <abbr title="World Wide Web Consortium">W3C</abbr> sets web standards.</p>
<p style="color: red;">Urgent</p>
<input type="text" required placeholder="required field">
<input type="checkbox" checked> Subscribe
<button disabled>Can't click me</button>
```

[Visit COMSATS](https://www.comsats.edu.pk)

Founded in 1998, W3C sets web standards.

Urgent



Subscribe

Can't click me

Headings

```
<h1>Chapter Title</h1>  
<h2>Section Heading</h2>  
<h3>Sub-section Heading</h3>
```

Chapter Title

Section Heading

Sub-section Heading

Text Formatting Elements

```
<strong>Warning</strong>: this is <em>really</em> important.  
<b>Bold text</b> and <i>Italic text</i>.  
<mark>highlighted</mark> text, and <small>terms apply</small>.  
H<sub>2</sub>O and x<sup>2</sup>  
Line one<br>Line two  
<hr>
```

Warning: this is *really* important.

Bold text and *Italic text*.

highlighted text, and terms apply.

H₂O and x²

Line one
Line two

The Tag

```
<font color="red" size="5" face="Arial">  
  This text is red, size 5, Arial font.  
</font>
```

This text is red, size 5, Arial font.

Why We Don't Use

- Mixes structure with appearance — HTML should describe what something is, not how it looks
- Doesn't scale — changing 50 pages means editing 50 pages, not one CSS rule
- No responsive or interactive ability — can't change on hover, on a smaller screen, or in dark mode
- Fails validation — flagged as an error by the W3C Markup Validator

<pre>, <code>, and Related Tags

Tag	Meaning
<pre>	Preformatted text — preserves every space and line break, monospace
<code>	An inline snippet of computer code, monospace
<kbd>	Keyboard input — a key the user should press
<samp>	Sample output from a program or command
<var>	A variable name or placeholder value
<blockquote>	A longer, block-level quotation

Technical Text Tags, Rendered

```
<pre>function greet() {  
    console.log("Hello!");  
}</pre>  
<p>Use the <code>console.log()</code> function.</p>  
<p>Press <kbd>Ctrl</kbd> + <kbd>C</kbd> to copy.</p>  
<p>Printed: <samp>Build succeeded</samp></p>  
<p>Replace <var>filename</var> with your own.</p>  
<blockquote>The Web connects people.</blockquote>
```

```
function greet() {  
    console.log("Hello!");  
}
```

Use the `console.log()` function to print output.

Press `Ctrl + C` to copy.

The command printed: `Build succeeded`

Replace *filename* with your own file's name.

| The Web does not just connect machines, it connects people.

HTML Entities

Entity	Renders As	Why You Need It
&lt;	<	A literal < would start a tag
&gt;	>	A literal > would end a tag
&amp;	&	A literal & would start another entity
&nbsp;	(a space)	A non-breaking space — never wraps a line
&copy;	©	Not on most keyboards
&mdash;	—	An em dash, longer than a hyphen

Entities, Rendered

```
<p>5 &lt; 10 and 10 &gt; 5</p>  
<p>Copyright &copy; 2025 &mdash; All rights reserved</p>  
<p>Click&nbsp;Here&nbsp;Now (no line break allowed)</p>
```

5 < 10 and 10 > 5

Copyright © 2025 — All rights reserved

Click Here Now (no line break allowed between these words)

Internal and External Links

EXTERNAL (ABSOLUTE URL)

- The complete address, including https:// and the domain
- Use for a page on a DIFFERENT website
- `target="_blank" + rel="noopener noreferrer"` opens safely in a new tab

INTERNAL (RELATIVE URL)

- A path relative to the current page — no domain name
- Use for pages within YOUR OWN site
- Survives a domain move, since it never contained the domain

Links, Rendered

```
<a href="https://en.wikipedia.org/wiki/HTML">HTML on Wikipedia</a>  
<a href="https://www.comsats.edu.pk" target="_blank" rel="noopener noreferrer">Visit COMSATS (new tab)</a>  
<a href="about.html">About Us</a>  
<a href="pages/contact.html">Contact (in a subfolder)</a>  
<a href=" ../index.html">Back to home (one folder up)</a>
```

[HTML on Wikipedia](https://en.wikipedia.org/wiki/HTML)

[Visit COMSATS \(new tab\)](https://www.comsats.edu.pk)

[About Us](about.html)

[Contact \(in a subfolder\)](pages/contact.html)

[Back to home \(one folder up\)](../index.html)

The Element

```

```

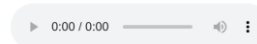
The image shows the COMSATS logo, which consists of the word "COMSATS" in white, bold, uppercase letters centered on a dark blue rectangular background.

Audio and Video

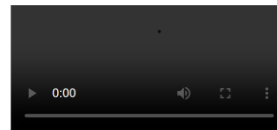
```
<audio controls>
  <source src="song.mp3" type="audio/mpeg">
  Your browser does not support audio.
</audio>
```

```
<video width="480" controls>
  <source src="movie.mp4" type="video/mp4">
  Your browser does not support video.
</video>
```

Audio player:



Video player:



Tables: <table>, <tr>, <th>, <td>

```
<table>
  <thead>
    <tr><th>Name</th><th>Course</th><th>Grade</th></tr>
  </thead>
  <tbody>
    <tr><td>Ali</td><td>CSC336</td><td>A</td></tr>
    <tr><td>Sara</td><td>CSC336</td><td>A+</td></tr>
  </tbody>
</table>
```

Name	Course	Grade
Ali	CSC336	A
Sara	CSC336	A+

Unordered, Ordered, and Description Lists

```
<ul>
  <li>HTML</li><li>CSS</li><li>JavaScript</li>
</ul>

<ol>
  <li>Write the HTML</li><li>Add CSS styling</li>
</ol>

<dl>
  <dt>HTML</dt><dd>The markup language...</dd>
</dl>
```

- HTML
 - CSS
 - JavaScript
1. Write the HTML
 2. Add CSS styling
 3. Add JavaScript behavior

HTML
The markup language used to structure web pages.

CSS
The language used to style web pages.

A Closer Look: Description Lists

```
<dl>
  <dt>HTML</dt>
  <dd>The markup language used to structure web pages.</dd>
  <dt>CSS</dt>
  <dd>The language used to style web pages.</dd>
</dl>
```

HTML

The markup language used to structure web pages.

CSS

The language used to style web pages.

Nested Lists

```
<ul>
  <li>Front End
    <ul>
      <li>HTML</li>
      <li>CSS</li>
    </ul>
  </li>
  <li>Back End
    <ul>
      <li>Node.js</li>
      <li>Express</li>
    </ul>
  </li>
</ul>
```

- Front End
 - HTML
 - CSS
- Back End
 - Node.js
 - Express

Nested List Rules

- A nested `<ul>/<ol>` goes inside the `<li>` of the item it belongs under
- A nested `<ol>` restarts its own numbering from 1, independent of the level above
- Browsers change bullet style (disc → circle → square) at each nesting depth

What HTML5 Added

- Semantic layout: header, nav, main, section, article, aside, footer
- Media: audio, video, source, track — no plugins required
- Graphics: canvas and native svg support
- New attributes: placeholder, required, pattern, data-*, contenteditable, draggable

Progressive Enhancement

TIP

Not every browser supports every HTML5 feature. Check caniuse.com before relying on a newer one, and design your page so it still works reasonably well in older browsers while enhancing the experience for those that support more.

KEY TAKEAWAYS

Lecture 3 in Six Points

- 1 A tag has an opening tag, content, and a closing tag; attributes live inside the opening tag; comments and attributes never appear in the rendered page.
- 2 Void elements (`br`, `img`) have no closing tag; nesting must close in reverse order.
- 3 Block-level elements stack and fill width; inline elements flow within text.
- 4 The `<font>` tag is obsolete — use CSS for all styling instead.
- 5 `<pre>/<code>` handle technical text; entities (`<` ` `) represent characters HTML can't take literally.
- 6 HTML has three list types — unordered, ordered, description (`<dl>`) — and any of them can be nested.