Introduction to HTML

What is HTML?

HTML (HyperText Markup Language) is the standard markup language used to create and structure content on the Web.

It is not a programming language.

What HTML Does:

- Defines **structure** (headings, paragraphs, lists, tables)
- It uses **elements (tags)** to describe the structure of web pages.
- Embeds **media** (images, audio, video)
- Creates **hyperlinks** between pages

Example Structure:

```html
<!DOCTYPE html>
<html>
<head>
    <title>My First Page</title>
</head>
<body>
    <h1>Welcome</h1>
    <p>This is HTML content.</p>
</body>
</html>
```

Key Concept:

HTML provides **structure**, CSS provides **styling**, and JavaScript provides **behavior**.

History and Evolution of HTML

HTML was created by **Tim Berners-Lee** in 1991.

Why HTML Was Created

To:

- Share scientific documents electronically
- Link research papers using hyperlinks
- Standardize information exchange over the internet

Evolution Timeline

Year	Version	Key Features
1991	HTML 1.0	Basic tags (headings, paragraphs, links)
1995	HTML 2.0	Standardization by IETF
1997	HTML 3.2	Tables, scripting support
1999	HTML 4.01	CSS support, better structure
2014	HTML 5	More Markup less scripting

HTML Versions (HTML 2.0 → HTML 4.01)

Let's examine the important versions before HTML5.

- **HTML 2.0 (1995)**
  - First official standard
  - Basic structural tags
  - Forms introduced

Limitation:

- No styling capabilities

- **HTML 3.2 (1997)**
  - Introduced tables
  - Early scripting support

Problem:

- Mixing content with presentation
- Poor separation of concerns

- **HTML 4.01 (1999)**

Major milestone version.

Key Improvements:

- Better support for CSS
- Introduction of semantic structure
- Separation of structure and presentation
- Frames and scripting enhancements

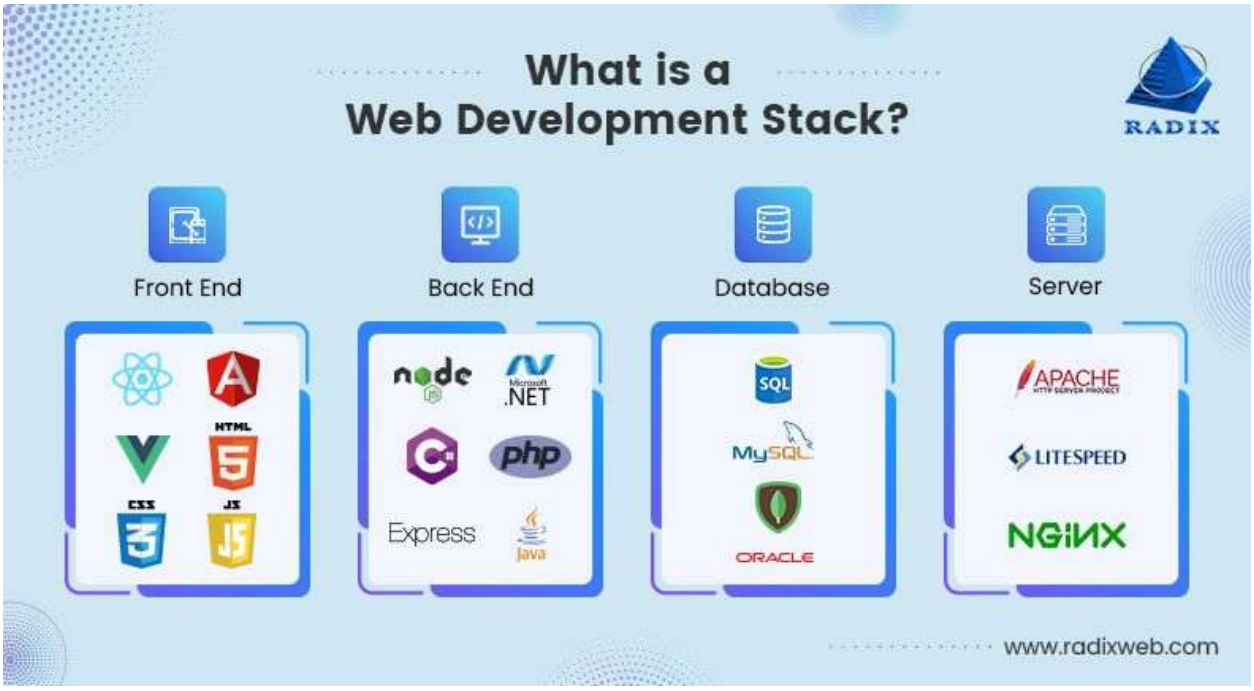
- HTML5 (2014 – Present Standard)

The modern version.

Major Features:

- **Semantic Elements**
- **Multimedia Support**
- **New Input Types**
- **APIs**

Role of HTML in Web Development



HTML is the **foundation layer** of every website.

Web Stack Relationship

Layer	**Technology**	**Purpose**
Structure	HTML	Defines layout & content
Styling	CSS	Design & presentation
Behavior	JavaScript	Interactivity
Backend	Server-side language	Business logic & data

Without HTML:

- No structure
- No DOM
- No webpage rendering

Real-World Example

When you visit a website:

1. Browser requests HTML from server
2. Server responds with HTML document
3. Browser parses HTML → builds DOM
4. CSS & JS are applied
5. Page is rendered

Basic Web Terminologies

🌐 URL (Uniform Resource Locator)

A URL is the **address of a resource** on the internet.

Example:

https://www.example.com/about.html

Structure of URL:

- Protocol → https
- Domain → example.com
- Path → /about.html

🌎 WWW (World Wide Web)

World Wide Web

The WWW is:

- WWW is a service that runs on the Internet.
- A collection of interconnected web pages
- Runs on top of the Internet
- Uses HTTP/HTTPS protocols

📱 Browser

Examples:

- Google Chrome
- Mozilla Firefox
- Microsoft Edge

What Browser Does:

- Sends HTTP request
- Receives HTML
- Parses HTML
- Renders webpage visually

🖧 Server

A server is:

- A machine that stores websites
- Responds to browser requests
- Sends HTML, CSS, JS files

Example Server Software:

- Apache HTTP Server
- Nginx

📌 Summary (Concept Map)

HTML:

- Is a markup language
- Created in 1991
- Evolved from HTML 1.0 → 5.01
- Forms the structure of web pages
- Works with CSS & JavaScript
- Delivered by server
- Rendered by browser
- Accessed using URLs

Structure of an HTML Document

Every HTML document follows a **standard structure**.

```
<!DOCTYPE html>
<html>
    <head>
        <title></title>
    </head>
    <body>
    </body>
</html>
```

🔹 1. <!DOCTYPE html>

- Declares the document type
- Tells browser: "This is HTML5"
- Must be the first line

- Not a tag, but a declaration

🔹 **2. <html> Tag**

- Root element
- All HTML content must be inside it
- Can include lang attribute <html lang="en">

🔹 **3. <head> Tag**

Contains **metadata** (not visible on webpage).

Examples:

- <title>
- <meta>
- <link>
- <style>
- <script>

🔹 **4. <title> Tag**

- Defines page title
- Appears in browser tab
- Important for SEO
- <title>Home Page</title>

🔹 **5. <body> Tag**

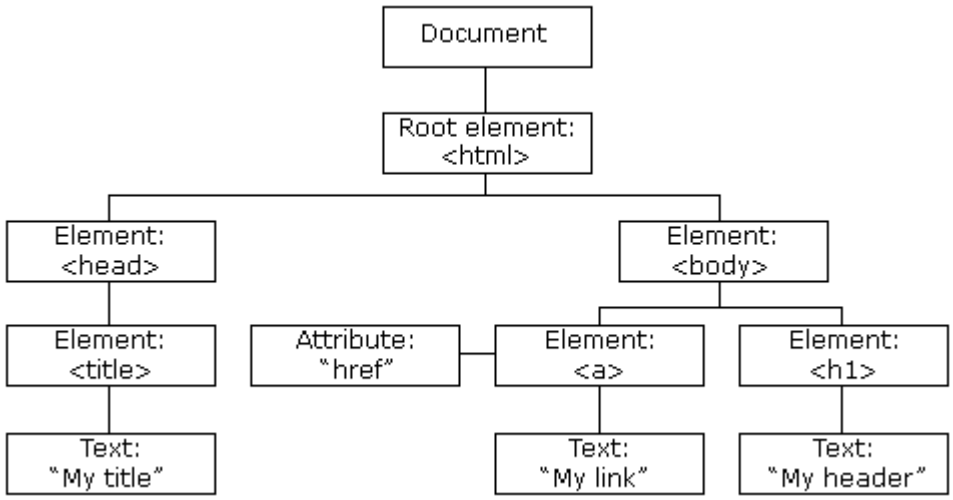
- Contains visible content
- Text, images, buttons, forms, tables etc.

  <body>

    <h1>Hello</h1>

  </body>

Visual Representation of Structure



HTML Editors

To write HTML, we need a **text editor**.

📝 1. Notepad (Basic Editor)

💻 2. VS Code (Professional Editor)

What is an HTML Element?

An **HTML Element** is the fundamental building block of a web page. It defines structure and meaning.

Structure of an Element

<tagname>Content</tagname>

Example:

This is a paragraph

- → Opening tag
- This is a paragraph → Content
- → Closing tag

👉 Together, this forms a **paragraph element**.

Element Syntax Breakdown

Visit Site

- <a> → Anchor element
- href="https://example.com" → Attribute
- Visit Site → Content

Empty vs Container Elements

HTML elements are categorized based on whether they contain content.

🔹 **A) Container Elements (Paired Elements)**

These have **opening and closing tags**.

Content here

Examples:

-
-
-
- <h1></h1>

They can contain:

- Text
- Other elements
- Both

🔹 **B) Empty Elements (Void Elements)**

These **do NOT have closing tags** and cannot contain content.

<img src="image.jpg" alt="Image">

<hr>

Examples:

-
 → Line break
- <hr> → Horizontal line
- <img> → Image
- <input> → Form input

Visual Comparison

Attributes and Attribute Values

An **attribute** provides additional information about an element.

Syntax:

<tag attribute="value">

Example:

<img src="logo.png" alt="Company Logo">

- src → Attribute
- "logo.png" → Attribute value
- alt → Attribute
- "Company Logo" → Attribute value

Key Points:

- Attributes are always written inside the **opening tag**
- Attribute values are placed inside **quotes**
- Multiple attributes are separated by spaces

Example:

```
<a href="https://google.com" target="_blank" title="Open Google">
  Google
</a>
```

Global Attributes

Global attributes can be used on **almost all HTML elements**.

- **id**
  - Uniquely identifies an element
  - Used in CSS and JavaScript
  - Must be unique in a page

```
<p id="intro">Welcome</p>
```

- **class**
  - Used to group multiple elements
  - Reusable
  - Mainly used for CSS styling

```
<p class="highlight">Text 1</p>
<p class="highlight">Text 2</p>
```

- **title**
  - Shows tooltip when hovering

```
<p title="This is a tooltip">Hover over me</p>
```

- **style**
  - Inline CSS styling
  - Not recommended for large projects (better use CSS files)

```
<p style="color:red; font-size:20px;">
  Styled Text
</p>
```

Example Demonstrating All Global Attributes

```
<div id="main" class="container" title="Main Section" style="background-color:lightgray;">
  Content goes here
</div>
```

Comments in HTML

Comments are used to:

- Explain code
- Temporarily disable code
- Improve readability

Syntax:

```
<!-- This is a comment -->
```

Example:

```
<!-- Navigation Section -->
<nav>
  <a href="#">Home</a>
</nav>
```

Important Rules:

- Comments are **not visible** in the browser
- Cannot be nested
- Used only for developers

🔎 Quick Summary Table

Concept	Definition
HTML Element	Tag + content + closing tag
Container Element	Has opening & closing tag
Empty Element	No closing tag
Attribute	Extra information inside opening tag
Global Attributes	id, class, title, style
Comment	Developer note (<!-- -->)

HTML Text Formatting Tags – Instructor Explanation

In HTML, **text formatting tags** are used to structure and style textual content on a web page.

Headings (<h1> to <h6>)

Headings define **titles and sub-titles** of content.
They create **document hierarchy** (important for SEO and accessibility).

📌 Syntax

<h1>Main Heading</h1>

<h2>Sub Heading</h2>

<h3>Section Heading</h3>

...

<h6>Smallest Heading</h6>

Key Points

- <h1> → Most important (usually page title)
- <h6> → Least important
- Use headings for **structure**, not just size.

Paragraph ( )

Defines a **block of text**.

📌 **Syntax**

This is a paragraph.

Key Points

- Automatically adds spacing before and after.
- Used for normal content text.

**Line Break and Horizontal Rule (
, <hr>)**

🔹 **Line Break (<br>)**

Breaks the line without starting a new paragraph.

📌 **Syntax**

First Line

Second Line

🔹 **Horizontal Rule (<hr>)**

Creates a horizontal line to separate content sections.

📌 **Syntax**

<hr>

🎯 Summary Table

Tag	Purpose	Type
<h1>–<h6>	Headings	Structural
<p>	Paragraph	Structural
 	Line Break	Formatting
<hr>	Horizontal Line	Formatting
<b>	Bold	Visual
	Important Text	Semantic
<i>	Italic	Visual
	Emphasized	Semantic
<small>	Smaller Text	Visual
<big>	Bigger Text	Obsolete
<sub>	Subscript	Formatting
<sup>	Superscript	Formatting
<pre>	Preserves formatting	Structural

Certainly. I will explain each topic in a structured, instructor-led manner with clear examples and practical usage.

HTML Links (Hyperlinks)

Anchor Tag (<a>)

The **anchor tag** is used to create hyperlinks in HTML.

Syntax:

Link Text

Example:

Visit Google

- href = Hypertext Reference (destination address)

Absolute vs Relative URLs

Absolute URL

- Full web address
- Includes protocol (http/https) and domain

Microsoft

Use when linking to external websites.

Relative URL

- Path relative to current file
- Used within same project

About Us

Contact

Use when linking between pages in your website.

Target Attribute

Controls where the link opens.

Open in New Tab

Common values:

- _self → Same tab (default)
- _blank → New tab
- _parent
- _top

Linking to External Pages

Visit Wikipedia

Always use target="_blank" for external sites.

Linking Within the Same Page (Bookmarks)

Step 1: Create an ID

<h2 id="section1">Section 1</h2>

Step 2: Link to it

Go to Section 1

Used for long pages and navigation menus.

HTML Images

Image Tag (<img>)

Used to display images.

Syntax:

<img src="image.jpg" alt="Description">

⚠ <img> is an empty tag (no closing tag).

Important Attributes

Attribute	Purpose
src	Image path
alt	Alternative text (SEO + accessibility)
width	Width in px
height	Height in px

Example:

```
<img src="logo.png" alt="Company Logo" width="200" height="100">
```

Image Paths

Same Folder

```
<img src="photo.jpg">
```

Inside Images Folder

```
<img src="images/photo.jpg">
```

Parent Folder

```
<img src="../photo.jpg">
```

HTML Lists

Ordered List (<ol>)

Numbered list.

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

Unordered List (<ul>)

Bulleted list.

```
<ul>
  <li>Apple</li>
  <li>Banana</li>
  <li>Mango</li>
</ul>
```

List Items (<li>)

Used inside <ol> or <ul>.

Nested Lists

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

Definition Lists (<dl>, <dt>, <dd>)

Used for terms and descriptions.

```
<dl>
  <dt>HTML</dt>
```

```
  <dd>HyperText Markup Language</dd>

  <dt>CSS</dt>
  <dd>Cascading Style Sheets</dd>
</dl>
```

HTML Tables

Table Structure

```
<table>
  <tr>
    <th>Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>John</td>
    <td>25</td>
  </tr>
</table>
```

Tags Explained

Tag	Meaning
<table>	Table container
<tr>	Table row
<th>	Header cell

Tag	Meaning
<td>	Data cell

Table Attributes

<table border="1" cellpadding="10" cellspacing="5">

- border → Border thickness
- cellpadding → Space inside cell
- cellspacing → Space between cells

(Note: In modern HTML5, CSS is preferred.)

Table Sections

```
<table border="1">
  <thead>
    <tr>
      <th>Name</th>
      <th>Marks</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Alice</td>
      <td>90</td>
    </tr>
  </tbody>
  <tfoot>
```

```
      <tr>
        <td>Total</td>
        <td>90</td>
      </tr>
    </tfoot>
</table>
```

Rowspan and Colspan

```
<td rowspan="2">Merged Row</td>
<td colspan="2">Merged Column</td>
```

- rowspan → Merge rows
- colspan → Merge columns

---

9. HTML Forms

Introduction to Forms

Forms collect user input and send data to a server.

Form Tag (<form>)

```
<form action="submit.php" method="post">
```

Form Attributes

Attribute	**Purpose**
action	Server URL
method	GET or POST

GET

- Data visible in URL
- Used for search forms

POST

- Data hidden
- Used for login, registration

Input Controls

Text & Password

<input type="text" name="username">

<input type="password" name="password">

Radio Buttons

<input type="radio" name="gender" value="male"> Male

<input type="radio" name="gender" value="female"> Female

(Same name → single selection)

Checkbox

<input type="checkbox" name="hobby" value="reading"> Reading

(Multiple selections allowed)

Submit, Reset, Button

<input type="submit" value="Submit">

<input type="reset" value="Clear">

<button type="button">Click Me</button>

Hidden Fields

<input type="hidden" name="userId" value="101">

Used to send invisible data.

Textarea

<textarea rows="4" cols="30"></textarea>

Select Box

<select name="city">

  <option value="chennai">Chennai</option>

  <option value="delhi">Delhi</option>

</select>

Labels and Fieldsets

<label for="username">Username:</label>

<input type="text" id="username">

<fieldset>

  <legend>Personal Info</legend>

  ...

</fieldset>

- label improves accessibility
- fieldset groups related fields

HTML Meta Information

Meta information in HTML provides **data about the document**, not visible to users on the webpage but essential for:

- Browsers
- Search engines
- Other web services

In **pre-HTML5 (HTML 4.01 and earlier)**, meta tags were heavily used for SEO and browser control.

Meta Tag (<meta>)

The <meta> tag is placed inside the <head> section of an HTML document.

It provides:

- Character encoding
- Page description
- Keywords
- Author info
- Auto-refresh

📌 **Basic Syntax**

```
<head>
  <meta name="attribute-name" content="value">
</head>
```

OR

```
<meta http-equiv="property" content="value">
```

📌 **Example**

```
<head>
  <meta name="description" content="Learn HTML basics with examples">
```

```
  <meta name="keywords" content="HTML, Web Design, Tutorial">
  <meta name="author" content="John Doe">
</head>
```

Charset (Character Encoding)

Character encoding defines how characters are stored and displayed.

In older HTML versions:

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

Why Charset is Important?

- Prevents garbled text
- Supports special characters (₹, €, ñ, etc.)
- Ensures cross-browser compatibility

Common Character Sets:

- UTF-8 (most recommended)
- ISO-8859-1 (older websites)

Keywords and Description

These were **very important for SEO in early web development**.

🔹 **Description**

Provides summary of the page.

<meta name="description" content="Free HTML tutorial for beginners.">

✓ Appears in search engine result snippets
✓ Helps improve click-through rate

🔹 **Keywords**

List of important words related to the page.

<meta name="keywords" content="HTML, CSS, Web Development, Tutorial">

⚠ In modern SEO, keywords meta tag is mostly ignored by search engines.

Author and Refresh

- **Author**

Specifies the document author.

<meta name="author" content="Santhosh Kumar">

Useful for:

- Documentation
- Academic pages
- Company websites

- **Refresh**

Automatically refreshes or redirects the page.

Example – Refresh every 5 seconds

<meta http-equiv="refresh" content="5">

Example – Redirect after 3 seconds

<meta http-equiv="refresh" content="3; url=https://example.com">

⚠ Not recommended for SEO. Use server-side redirects instead.

HTML Character Entities & Symbols

What Are HTML Character Entities?

In HTML, some characters have **special meanings**.
For example:

- < and > are used to define tags.

- & is used to define entities.

If we directly type these characters in content, the browser may interpret them as **HTML code instead of text**.

👉 To display such characters correctly, we use **HTML Character Entities**.

General Syntax:

&entity_name;

or

&#entity_number;

Example:

5 < 10

Output:

5 < 10

Special Characters

These are the most commonly used HTML entities.

Character	Entity Name	Entity Number	Purpose
<	<	<	Less than
>	>	>	Greater than
&	&	&	Ampersand
"	"	"	Double quotes
'	'	'	Single quote
Space			Non-breaking space

🔹 **Example: < and >**

HTML stands for <Hyper Text Markup Language>

Output:

HTML stands for <Hyper Text Markup Language>

- **Example: (Non-Breaking Space)**

Normally, HTML removes extra spaces.

Hello World

Output:

Hello World

To add extra space:

Hello   World

Output:

Hello World

👉   prevents line break between words.

Currency Symbols

HTML provides predefined entities for various currency symbols.

Currency	Symbol	Entity Name	Entity Number
Dollar	$	&#36;	&#36;
Euro	€	&euro;	&#8364;
Pound	£	&pound;	&#163;
Yen	¥	&yen;	&#165;
Rupee	₹	&#8377;	&#8377;

- **Example:**

Product Price: £50

Price in India: ₹5000

Price in Europe: €100

Output:

Product Price: £50

Price in India: ₹5000

Price in Europe: €100

💡 In Indian websites, ₹ is commonly used for ₹.

Copyright & Trademark Symbols

These symbols are frequently used in websites and applications.

Symbol	Meaning	Entity Name	Entity Number
©	Copyright	©	©
®	Registered Trademark	®	®
™	Trademark	™	™

🔹 **Example:**

© 2026 MyCompany. All rights reserved.

ChatPro™

BrandName®

Output:

© 2026 MyCompany. All rights reserved.

ChatPro™

BrandName®

Why Are Character Entities Important?

✅ **Avoid HTML Parsing Errors**

Example:

10 < 20

This will cause error.

Correct way:

10 < 20

✅ **Improve Readability & Compatibility**

Some browsers may not render special symbols correctly without entities.

✅ **Professional Website Footer Usage**

<footer>

© 2026 San Technologies. All rights reserved.

</footer>
