

COMPLETE COURSE

HTML

Mastery

Handbook

From Zero to Hero · 17 Modules · Real Projects

HTML5

Web Dev

Beginner Friendly

LMS Ready

Table of Contents

Module 01 — Introduction to HTML

- What is HTML?
- History of HTML
- How Websites Work
- Frontend vs Backend
- Role of HTML

Module 02 — Setting Up Environment

- Installing VS Code
- Best Extensions
- Project Folder
- First HTML File

Module 03 — HTML Document Structure

- Basic Structure
- DOCTYPE
- html / head / body Tags

Module 04 — Headings & Paragraphs

- H1–H6 Headings
- Paragraphs
- Line Break
- Horizontal Rule

Module 05 — Text Formatting Tags

- Bold & Strong
- Italic & Emphasis
- Underline & Mark

Module 06 — Links & Navigation

- Basic Link
- New Tab Links
- Email & Phone Links
- Anchor Links

Module 07 — Images in HTML

- Adding Images
- Alt Text
- Width & Height
- Linked Images

Module 08 — Lists in HTML

- Ordered Lists
- Unordered Lists
- Nested Lists

Module 09 — Tables in HTML

- Basic Table
- Table Tags
- thead / tbody / tfoot

Module 10 — Forms in HTML

- Basic Form
- Input Types
- Radio & Checkbox
- Select & Textarea

Module 11 — Semantic HTML

- What is Semantic HTML?
- Semantic Tags
- Why It Matters

Module 12 — Multimedia in HTML

- Audio Tag
- Video Tag
- YouTube Embed

Module 13 — HTML Entities & Symbols

- Why Entities?
- Common Entities
- Examples

Module 14 — Meta Tags & SEO Basics

- Essential Meta Tags
- Viewport Tag
- Open Graph Tags

Module 15 — Accessibility Basics

- Alt Text Rules
- Heading Order
- Form Accessibility

Module 16 — HTML Best Practices

- Clean Code
- Indentation
- File Naming
- Validation

Module 17 — Real HTML Projects

- Portfolio Website
- Blog Webpage
- Restaurant Site
- Final Project

MODULE 01

Introduction to HTML

Understanding the foundation of every webpage

What is HTML?

HTML stands for **HyperText Markup Language**. It is the backbone of every webpage on the internet — the structural layer that tells a browser what content to display and how to organise it.

| Word | Meaning |
|-----------|--|
| HyperText | Connects pages to each other via hyperlinks |
| Markup | Uses special tags to label and structure content |
| Language | A set of rules the browser understands and renders |

What Does HTML Actually Do?

HTML instructs the web browser where to place things on the page:

- Where the **heading** (main title) appears
- Where **paragraphs** of text are displayed
- Where **images** and media are embedded
- Where **buttons** and forms are placed
- Where **navigation links** lead to other pages

Your First HTML Code

```
<h1>Hello World</h1>
<p>This is my first webpage.</p>
```

■ Tip

HTML uses tags — words inside angle brackets like `<h1>` — to tell the browser what each piece of content represents. Every tag has an opening (`<h1>`) and a closing (`</h1>`) form.

History of HTML

| Year | Event |
|------|---|
| 1989 | Tim Berners-Lee conceives the idea of sharing documents over the internet |

| | |
|------|--|
| 1991 | HTML version 1 introduced — only 18 tags! |
| 1995 | HTML 2.0 released — forms support added |
| 1999 | HTML 4.01 with better structure and CSS integration |
| 2014 | HTML5 officially finalised — audio, video, and semantic tags |

■ Tip

Tim Berners-Lee, a British scientist at CERN, invented HTML in 1991 to allow researchers to share documents over the internet. He also created the World Wide Web itself.

How Websites Work — Step by Step

Step 1

You type a URL (e.g., <https://google.com>) in your browser's address bar.

Step 2

Your browser sends an HTTP request to Google's server: 'Please send me your homepage.'

Step 3

The server responds with three files: HTML (structure), CSS (styling), JavaScript (behaviour).

Step 4

Your browser reads all three files and renders the finished, interactive webpage.

■ Tip

Analogy: Think of it like ordering food. You (browser) place an order → the kitchen (server) prepares it → the waiter delivers it (HTML/CSS/JS) → you enjoy the meal (webpage).

Frontend vs Backend

| Aspect | Frontend | Backend |
|--------------|---------------------------------------|------------------------------------|
| What it is | What the user sees and interacts with | Works behind the scenes |
| Examples | Buttons, images, menus, text | Databases, login systems, payments |
| Technologies | HTML, CSS, JavaScript | PHP, Python, Node.js, MySQL |

■ Tip

Analogy: A restaurant has a dining room (frontend — what customers experience) and a kitchen (backend — where everything is prepared). HTML is how you build the dining room.

MODULE 02

Setting Up Your Environment

Install the tools every professional web developer uses

Installing VS Code

VS Code (Visual Studio Code) is a free, open-source code editor made by Microsoft. It is the most widely used editor in the world for web development.

Step 1

Open your browser and navigate to: <https://code.visualstudio.com>

Step 2

Click the large Download button — it automatically detects your OS (Windows / Mac / Linux).

Step 3

Open the downloaded installer file and follow the setup wizard with default settings.

Step 4

Launch VS Code — you will see a Welcome tab with tips for getting started.

Must-Have Extensions

Extensions turbocharge VS Code. Install these 4 from the Extensions panel (Ctrl+Shift+X):

Live Server

Shows your HTML in the browser in real time. Every save auto-refreshes the browser.

Prettier

Auto-formats your code for consistent style. Press Shift+Alt+F to format.

Auto Rename Tag

When you rename an opening tag, the closing tag updates automatically.

HTML CSS Support

Provides intelligent autocomplete suggestions as you type HTML and CSS.

Creating Your First HTML File

Step 1

Create a folder named MyWebsite on your Desktop or Documents.

Step 2

Open VS Code → File → Open Folder → select MyWebsite.

Step 3

Click the New File icon in the Explorer panel and name it index.html.

Step 4

Type the code below, save with Ctrl+S, then right-click → Open with Live Server.

```
<h1>Hello</h1>
```

■ ■ Note

Always name your main HTML file index.html. Browsers automatically look for this file first when a website is opened.

MODULE 03

HTML Document Structure

The required skeleton every HTML page must have

The Basic Structure

Every HTML page must follow this exact structure. Think of it as a formal letter — it has a required format that cannot be skipped:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>My Website</title>
</head>
<body>
<h1>Hello World</h1>
<p>Welcome to my first webpage.</p>
</body>
</html>
```

Understanding Each Part

<!DOCTYPE html>

Declaration (not a tag) that tells the browser this file uses HTML5. Must always be the very first line. Never change it.

<html>

The root element — every other element must be nested inside it. Always add lang="en" for accessibility and SEO.

<head>

Contains invisible page metadata: character set, page title, links to CSS, meta tags for SEO, viewport settings.

<title>

The text that appears on the browser's tab and in Google's search results. Keep it descriptive and under 60 characters.

<body>

Everything visible on the webpage lives here: text, images, buttons, forms, tables, videos — all of it.

■ ■ Note

*For every opening tag <tag> you must have a matching closing tag </tag>. The only exceptions are self-closing tags like
, <hr>, and .*

MODULE 04

Headings & Paragraphs

Structuring text content clearly and semantically

HTML Headings — H1 to H6

HTML provides six levels of headings. H1 is the largest and most important; H6 is the smallest. Use them in order — never skip a level.

```
<h1>Main Page Title – Use Only Once Per Page</h1>
<h2>Major Section Heading</h2>
<h3>Sub-Section Heading</h3>
<h4>Sub-Sub-Section</h4>
<h5>Minor Heading</h5>
<h6>Smallest Heading</h6>
```

■ Tip

SEO Rule: Use exactly ONE <h1> per page — it is your primary page title. Google uses it to understand the main topic. H2s are for major sections; H3s for subsections.

Paragraphs — <p>

The <p> tag wraps a block of text into a paragraph. Browsers automatically add spacing above and below each paragraph.

```
<p>This is the first paragraph. It can contain many sentences.</p>
<p>This is a separate paragraph. Notice the automatic spacing between them.</p>
```

Line Break
 and Horizontal Rule <hr>

```
<!-- Force a new line within a paragraph -->
<p>First line.<br>Second line on same paragraph.</p>
<!-- Draw a horizontal dividing line -->
<hr>
```

■■ Note

*
 and <hr> are self-closing — they have no closing tag. Use
 sparingly; prefer separate <p> tags for new paragraphs.*

MODULE 05

Text Formatting Tags

Making text bold, italic, highlighted, and more

All Text Formatting Tags

| Tag | Effect | Best Used For |
|-----------------------------|--------------------------------|---|
| <code></code> | Bold (visual only) | Stylistic emphasis — keywords, product names |
| <code></code> | Bold + important (semantic) | Critical warnings, key information |
| <code><i></code> | Italic (visual only) | Book titles, foreign words, technical terms |
| <code></code> | Italic + emphasised (semantic) | Stressed words, important phrases |
| <code><u></code> | Underline | Proper nouns, spelling mistakes (use carefully) |
| <code><mark></code> | Yellow highlight | Search results, important definitions |

```
<p>
Price is <b>£999</b> only.
<strong>Warning:</strong> Do not click unknown links.
The movie <i>Inception</i> was brilliant.
You should <em>always</em> save your work.
The most important formula is <mark>E = mc2</mark>.
</p>
```

■ Tip

Prefer `` over `` and `` over `<i>` whenever the text is genuinely important or emphasised — screen readers and search engines recognise the semantic meaning.

Links & Navigation

Connecting pages and building navigation menus

The Anchor Tag — <a>

The <a> (anchor) tag creates clickable hyperlinks. The href attribute specifies the destination URL.

```
<!-- Basic link -->
<a href="https://www.google.com">Visit Google</a>
<!-- Open in new browser tab -->
<a href="https://www.google.com" target="_blank">Google (new tab)</a>
<!-- Email link - opens email app -->
<a href="mailto:hello@mysite.com">Send an Email</a>
<!-- Phone link - opens dialler on mobile -->
<a href="tel:+919876543210">Call Us</a>
<!-- Link to another page on your own website -->
<a href="about.html">About Us</a>
<!-- Anchor link - jumps to section on same page -->
<a href="#contact">Jump to Contact</a>
<h2 id="contact">Contact Section</h2>
```

■ Tip

Always use target="_blank" when linking to external websites so users do not leave your site. Add rel="noopener noreferrer" for security.

MODULE 07

Images in HTML

Embedding, sizing, and linking images properly

The Tag

The tag embeds an image. It is self-closing — no closing tag is needed. The src attribute points to the image file.

```
<!-- Basic image -->


<!-- With alt text (required for accessibility + SEO) -->


<!-- With fixed width (height scales automatically) -->


<!-- Image inside a folder -->


<!-- Clickable image (image wrapped in anchor tag) -->
<a href="https://google.com">

</a>
```

■ ■ Note

Always write meaningful alt text for every image. Without it, visually impaired users and search engines cannot understand the image. A missing alt attribute is an accessibility failure.

MODULE 08

Lists in HTML

Organising content with ordered and unordered lists

Ordered List —

Use an ordered list when sequence matters — steps, instructions, rankings.

```
<ol>
<li>Open VS Code</li>
<li>Create index.html</li>
<li>Write your HTML</li>
<li>Save the file</li>
<li>Open with Live Server</li>
</ol>
```

Unordered List —

Use an unordered list when order does not matter — features, options, ingredients.

```
<ul>
<li>HTML - Page structure</li>
<li>CSS - Visual styling</li>
<li>JavaScript - Interactivity</li>
</ul>
```

Nested Lists

```
<ul>
<li>Frontend
<ul>
<li>HTML</li>
<li>CSS</li>
</ul>
</li>
<li>Backend
<ul>
<li>Node.js</li>
<li>Python</li>
</ul>
</li>
</ul>
```

MODULE 09

Tables in HTML

Displaying structured data in rows and columns

Basic Table Structure

| Tag | Full Name | Purpose |
|---------|--------------|---|
| <table> | Table | Creates the table container |
| <tr> | Table Row | Creates one horizontal row |
| <th> | Table Header | Header cell — bold and centred by default |
| <td> | Table Data | Regular data cell |
| <thead> | Table Head | Groups header rows (semantic) |
| <tbody> | Table Body | Groups data rows (semantic) |
| <tfoot> | Table Footer | Groups footer rows — totals, summaries |

```
<table border="1">
<thead>
<tr>
<th>Name</th>
<th>Subject</th>
<th>Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rahul</td>
<td>Mathematics</td>
<td>92</td>
</tr>
<tr>
<td>Priya</td>
<td>Science</td>
<td>88</td>
</tr>
</tbody>
<tfoot>
<tr>
<td colspan="2">Class Average</td>
<td>90</td>
</tr>
</tfoot>
</table>
```

■ **Tip**

colspan lets a cell span multiple columns. rowspan lets a cell span multiple rows. Both accept a number: colspan="3" spans 3 columns.

MODULE 10

Forms in HTML

Collecting user input with forms and input fields

Basic Form Structure

```
<form action="submit.php" method="POST">
<label for="name">Full Name:</label>
<input type="text" id="name" name="name" placeholder="Enter your name">
<label for="email">Email:</label>
<input type="email" id="email" name="email" placeholder="Enter your email">
<label for="pass">Password:</label>
<input type="password" id="pass" name="pass">
<button type="submit">Submit</button>
</form>
```

Input Types Reference

| type | Purpose | Behaviour |
|----------|------------------|---------------------------------------|
| text | Single-line text | Plain text input |
| email | Email address | Validates email format automatically |
| password | Secret entry | Characters shown as dots |
| number | Numeric entry | Shows up/down arrows; accepts min/max |
| date | Date picker | Opens a calendar widget |
| file | File upload | Opens file browser |
| radio | Single choice | Only one can be selected in the group |
| checkbox | Multiple choice | Any number can be selected |
| submit | Submit button | Sends the form data |

```
<!-- Radio buttons - only one can be selected -->
<input type="radio" id="male" name="gender" value="male">
<label for="male">Male</label>
<input type="radio" id="female" name="gender" value="female">
<label for="female">Female</label>

<!-- Checkboxes - multiple can be selected -->
<input type="checkbox" id="html" name="skill" value="html">
<label for="html">HTML</label>
<input type="checkbox" id="css" name="skill" value="css">
<label for="css">CSS</label>

<!-- Dropdown menu -->
<select name="city">
<option value="mumbai">Mumbai</option>
<option value="delhi">Delhi</option>
<option value="ahmedabad">Ahmedabad</option>
</select>

<!-- Multi-line textarea -->
<textarea name="message" rows="5" cols="40"></textarea>
```

MODULE 11

Semantic HTML

Writing meaningful code that search engines and users understand

What is Semantic HTML?

Semantic means *meaningful*. Semantic HTML uses tags that describe the **purpose** of the content — not just how it looks. Both examples below look the same in the browser, but only one tells the browser what the content *means*.

Non-Semantic (Poor Practice)

```
<div id="header">...</div>
<div id="nav">...</div>
<div id="content">...</div>
<div id="footer">...</div>
```

Semantic (Correct Practice)

```
<header>...</header>
<nav>...</nav>
<main>
<section>...</section>
<article>...</article>
<aside>...</aside>
</main>
<footer>...</footer>
```

Semantic Tags Reference

| Tag | Purpose |
|-----------|--|
| <header> | Site or section header — logo, branding, top navigation |
| <nav> | Navigation menu — links between pages |
| <main> | The primary unique content of the page (only one per page) |
| <section> | A thematic grouping of content with its own heading |
| <article> | Self-contained content: blog post, news story, card |
| <aside> | Secondary content: sidebar, related links, advertisements |

<footer>

Bottom of page — copyright, links, contact info

■ **Tip**

Three key benefits of semantic HTML: (1) Better SEO — Google understands your page structure. (2) Accessibility — screen readers navigate semantically. (3) Maintainability — cleaner, self-documenting code.

MODULE 12

Multimedia in HTML

Embedding audio, video, and YouTube directly in pages

Audio — <audio>

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

Video — <video>

```
<video controls width="640" height="360" poster="thumbnail.jpg">
<source src="lesson.mp4" type="video/mp4">
<source src="lesson.webm" type="video/webm">
Your browser does not support video.
</video>
```

| Attribute | Effect |
|-----------|---|
| controls | Displays play/pause/volume/scrub controls |
| autoplay | Plays automatically on page load |
| loop | Repeats the media continuously |
| muted | Starts with sound off (required for autoplay in Chrome) |
| poster | Image shown before the video starts playing |

YouTube Embed — <iframe>

```
<iframe
width="560"
height="315"
src="https://www.youtube.com/embed/VIDEO_ID_HERE"
title="YouTube video player"
frameborder="0"
allowfullscreen>
</iframe>
```

■ **Tip**

To get the embed code: open any YouTube video → Share → Embed → copy the <iframe> code. Replace 'watch?v=' with 'embed/' in the URL.

MODULE 13

HTML Entities & Symbols

Displaying special characters and reserved symbols

Why Entities Are Needed

Some characters have special meaning in HTML — like < and > which define tags. To display these characters as visible text on the page, you use HTML entities.

Common Entities Reference

| Symbol | Entity Code | Description |
|--------|-------------|----------------------|
| < | < | Less than sign |
| > | > | Greater than sign |
| & | & | Ampersand |
| | | Non-breaking space |
| © | © | Copyright symbol |
| ® | ® | Registered trademark |
| ™ | ™ | Trademark symbol |
| ₹ | ₹ | Indian Rupee symbol |
| € | € | Euro symbol |
| £ | £ | British Pound |

```
<p>Copyright &copy; 2025 ThinkEdu. All rights reserved.</p>
<p>Price: &#8377;999 only.</p>
<p>5 &lt; 10 is a true statement.</p>
<p>Bread &amp; Butter</p>
```

MODULE 14

Meta Tags & SEO Basics

Optimising your page for search engines and mobile devices

Essential Meta Tags

```
<!DOCTYPE html>
<html lang="en">
<head>
<!-- Character encoding: supports all world languages -->
<meta charset="UTF-8">
<!-- Makes website mobile-responsive (MANDATORY) -->
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<!-- Description shown in Google search results (150-160 chars) -->
<meta name="description"
content="Learn HTML from scratch with this complete course.">
<!-- Page author -->
<meta name="author" content="ThinkEdu">
<!-- Browser tab title + Google headline (50-60 chars) -->
<title>Complete HTML Course | ThinkEdu</title>
</head>
```

Open Graph Tags — Social Media Sharing

These tags control how your page looks when shared on Facebook, WhatsApp, LinkedIn, and Twitter.

```
<meta property="og:title" content="Complete HTML Course">
<meta property="og:description" content="Learn HTML step by step.">
<meta property="og:image" content="https://site.com/thumb.jpg">
<meta property="og:url" content="https://site.com/html-course">
```

■ Tip

Write your title between 50–60 characters and description between 150–160 characters for best Google display. These are what users see before they click your page in search results.

Accessibility Basics

Building websites that everyone can use

Why Accessibility Matters

About 15% of the world's population has some form of disability. Accessible websites are not only ethical — they reach more people, rank better in Google, and are legally required in many countries.

Alt Text Rules for Images

```
<!-- GOOD: descriptive and specific -->

<!-- BAD: too vague -->

<!-- EMPTY alt for decorative-only images -->

```

Correct Heading Order — Never Skip Levels

```
<!-- CORRECT: sequential order -->
<h1>Main Title</h1>
<h2>Section One</h2>
<h3>Sub-section</h3>
<h2>Section Two</h2>
<!-- WRONG: jumping from H1 to H3 -->
<h1>Title</h1>
<h3>Jumped — missed H2!</h3>
```

■ Tip

Always connect <label> to <input> using matching for and id attributes. This lets users click the label to focus the input — essential for motor-impaired users.

HTML Best Practices

Writing professional, maintainable, and valid code

The 7 Golden Rules

1. Use Proper Document Structure

Always include DOCTYPE, html, head, and body tags with correct nesting.

2. Write Clean, Indented Code

Indent nested elements consistently (2 or 4 spaces). Never write everything on one line.

3. Use Semantic Tags

Choose header, nav, main, section, article, footer over generic div elements wherever possible.

4. Always Add Alt Text

Every img tag must have an alt attribute — even if it is an empty string for decorative images.

5. Use Meaningful File Names

Name files clearly: about.html, hero-banner.jpg, style.css — not page2.html or img123.jpg.

6. Validate Your HTML

Visit validator.w3.org and paste your code to find and fix errors automatically.

7. Organise Your Project Folder

Keep HTML files at root, images in /images/, CSS in /css/, JavaScript in /js/.

Recommended Folder Structure

```
MyWebsite/  
  ■■■ index.html ← Homepage  
  ■■■ about.html ← About page  
  ■■■ contact.html ← Contact page  
  ■■■ images/  
    ■ ■■■ logo.png  
    ■ ■■■ hero-banner.jpg  
  ■■■ css/  
    ■ ■■■ style.css  
  ■■■ js/  
    ■■■ script.js
```

MODULE 17

Real HTML Projects

Building complete, real-world websites from scratch

Project 1 — Personal Portfolio

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Rahul Sharma – Web Developer</title>
</head>
<body>
<header>
<h1>Rahul Sharma</h1>
<p>Web Developer | HTML & CSS Enthusiast</p>
<nav>
<a href="#about">About</a> |
<a href="#skills">Skills</a> |
<a href="#contact">Contact</a>
</nav>
</header>
<main>
<section id="about">
<h2>About Me</h2>

<p>Hello! I am a passionate web developer learning HTML,
CSS, and JavaScript.</p>
</section>
<section id="skills">
<h2>My Skills</h2>
<ul>
<li>HTML5</li>
<li>CSS3</li>
<li>JavaScript (Beginner)</li>
</ul>
</section>
<section id="contact">
<h2>Contact Me</h2>
<form>
<label for="name">Name:</label>
<input type="text" id="name" placeholder="Your name">
<label for="email">Email:</label>
<input type="email" id="email">
<button type="submit">Send Message</button>
</form>
</section>
</main>
<footer>
<p>&copy; 2025 Rahul Sharma. All rights reserved.</p>
</footer>
</body>
</html>
```

Final Project — Complete Multi-Page Website

For your final project, build a complete website with five linked pages:

| Page | File | Must Include |
|----------|---------------|--|
| Home | index.html | Hero section, welcome text, navigation |
| About | about.html | Your story, team photo, mission statement |
| Services | services.html | Services list with images and descriptions |
| Blog | blog.html | 2–3 articles with headings, images, dates |
| Contact | contact.html | Contact form, address, phone, map embed |

Checklist for Every Page

- Correct `<!DOCTYPE html>` structure
- `charset`, `viewport`, and meta description in `<head>`
- Semantic tags: `<header>`, `<nav>`, `<main>`, `<footer>`
- Navigation links connecting all five pages
- All images have descriptive alt text
- At least one form or data table
- Copyright notice and contact info in footer
- Code validated at validator.w3.org

Congratulations! You have completed the full ThinkEdu HTML Mastery Course.

You now know enough HTML to build real, professional websites. Your next steps: learn CSS to make them beautiful, then JavaScript to make them interactive.

— ThinkEdu | thinkedu.in —