Introduction to HTML

What is HTML?

HTML stands for **HyperText Markup Language**.
It is the standard language used to create and structure webpages.

Web browsers (like Chrome, Firefox, Safari) read HTML code and display it as websites.

Why Learn HTML?

- It's the **foundation** of all websites.
- You need HTML to build pages before adding CSS (styling) or JavaScript (functionality).
- Knowing HTML helps you understand how websites work behind the scenes.

Basic Terminology

- **Element**: A piece of content in a webpage (like a paragraph, heading, or image).
- **Tag**: Special keywords inside angle brackets like `<p>` or `<h1>` that define elements.
- **Attribute**: Extra information added to tags, like `href` in links or `src` in images.

Basic Example

```html
<!DOCTYPE html>
<html>
  <head>
    <title>My First Page</title>
  </head>
  <body>
    <h1>Hello, world!</h1>
    <p>This is my first HTML page.</p>
  </body>
</html>
```

Explanation:

- `<!DOCTYPE html>` tells the browser this is an HTML5 document.
- `<html>` is the root of the HTML page.
- `<head>` contains information about the page (not shown on screen).
- `<title>` sets the title seen on the browser tab.
- `<body>` contains everything visible on the webpage.

Key Points

- HTML is made up of **tags**.
- Tags usually come in **pairs**: an opening tag `<p>` and a closing tag `</p>`.
- The content goes **between** the tags.
- Indentation helps make code easier to read, but it's not required.

Basic HTML Structure

What is the Basic Structure of an HTML Page?

Every HTML document follows a basic structure. This structure tells the browser how to read and display the content.

Template of a Basic HTML Page

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <!-- Your content goes here -->
  </body>
</html>
```

Explanation of Each Part

1. `<!DOCTYPE html>`

- Declares that this is an HTML5 document.
- Must be the first line in the file.

2. `<html>...</html>`

- The root element of the page.
- Wraps all the content of your HTML document.

3. `<head>...</head>`

- Contains meta-information **about** the page.
- This can include:
  - The page `<title>`
  - Links to CSS files
  - Meta tags (like keywords, description, etc.)

4. `<title>...</title>`

- Sets the name shown on the browser **tab**.

5. `<body>...</body>`

- Contains everything visible on the page.
- You'll place text, images, links, forms, etc. here.

Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>My First Web Page</title>
  </head>
  <body>
    <h1>Welcome!</h1>
    <p>This is a simple HTML page with basic structure.</p>
  </body>
</html>
```

Tips

- Always start with `<!DOCTYPE html>`.
- Make sure `<html>`, `<head>`, and `<body>` are properly opened and closed.
- Use indentation to keep your code clean and readable.

Headings and Paragraphs

Headings in HTML

Headings help you organize content into sections.
HTML provides **6 levels** of headings:

- `<h1>` – Main heading (biggest)
- `<h2>` – Subheading
- `<h3>` – Smaller subheading
- `<h4>`, `<h5>`, `<h6>` – Even smaller headings

Example of Headings

```
<h1>This is a Heading 1</h1>
<h2>This is a Heading 2</h2>
<h3>This is a Heading 3</h3>
<h4>This is a Heading 4</h4>
<h5>This is a Heading 5</h5>
<h6>This is a Heading 6</h6>
```


📌 Tip:

- Use **only one** `<h1>` per page (usually for the page title).
- Use headings to **structure your content**, not to make text look big (that's CSS's job).

Paragraphs in HTML

Paragraphs are written using the `<p>` tag.

Example:

```
<p>This is a paragraph. It can contain multiple sentences of text.</p>
```

Notes:

- Browsers automatically add **space** before and after each paragraph.
- You don't need to press Enter manually for new lines. Use a new `<p>` tag instead.

Line Breaks

If you want to break a line **without starting a new paragraph**, use the `<br>` tag.

Example:

```
<p>This is line one.<br>This is line two.</p>
```

> `<br>` is a **self-closing tag**, which means it doesn't need a closing `</br>`.

Complete Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>Headings and Paragraphs</title>
  </head>
  <body>
```

```
    <h1>My Blog</h1>
    <h2>Introduction</h2>
    <p>Welcome to my first HTML blog post!</p>

    <h2>Why I Love Coding</h2>
    <p>Coding lets you build websites, apps, and games.<br>It’s fun and creative!</
p>
  </body>
</html>
```

Formatting Text in HTML

HTML allows you to format your text using different tags. These tags help make your content easier to read and visually appealing.

Bold Text

Use the `<b>` or `<strong>` tag to make text bold.

```
<p>This is <b>bold</b> text.</p>
<p>This is <strong>important</strong> text.</p>
```

- `<strong>` also means the text is important (for screen readers and SEO).

Italic Text

Use the `<i>` or `<em>` tag to italicize text.

```
<p>This is <i>italic</i> text.</p>
<p>This is <em>emphasized</em> text.</p>
```

- `<em>` gives extra emphasis and has meaning, especially for accessibility.

Underlined Text

Use the `<u>` tag to underline text.

```
<p>This is <u>underlined</u> text.</p>
```

Strikethrough Text

Use the `<s>` or `<del>` tag to show deleted or crossed-out text.

```
<p>This is <s>wrong</s> text.</p>
<p>Old price: <del>$100</del> New price: $80</p>
```

Superscript and Subscript

Use `<sup>` for superscript (above line), `<sub>` for subscript (below line).

```
<p>Water is H<sub>2</sub>O.</p>
<p>E = mc<sup>2</sup></p>
```

Combining Formats

You can combine formatting tags.

```
<p>This is <b><i>bold and italic</i></b> text.</p>
```

Summary of Formatting Tags

Tag	Purpose
`<b>`	Bold (no meaning)
`<strong>`	Bold (important)
`<i>`	Italic (no meaning)
`<em>`	Italic (emphasis)
`<u>`	Underline
`<s>`	Strikethrough
`<del>`	Deleted text
`<sub>`	Subscript
`<sup>`	Superscript

Comments and Whitespace in HTML

HTML Comments

Comments are notes in your HTML code that are ignored by the browser.
They are useful for explaining code or leaving reminders.

Syntax:

```
<!-- This is a comment -->
<p>This is visible content.</p>
<!-- <p>This line will not show on the webpage.</p> -->
```

Comments do not appear on the webpage. They're only visible in the source code.

Whitespace in HTML

Whitespace includes spaces, tabs, and newlines (Enter key). HTML treats multiple spaces as a **single space**.

Example:

```
<p>This        is        spaced.</p>
```

This will be displayed as:

```
This is spaced.
```

If you want to preserve spaces and line breaks, use the `<pre>` tag.

Example:

```
<pre>
This    is     preformatted
   text.
</pre>
```

Tip

- Use comments to explain your HTML when needed.
- Use proper indentation (whitespace) to make your code readable, even though HTML doesn't require it.

Links and Anchor Tags

Creating Links in HTML

HTML uses the `<a>` tag to create links.
The `href` attribute tells the browser **where** the link should go.

Basic Syntax:

```
<a href="https://www.example.com">Visit Example</a>
```

This creates a clickable link that takes you to the specified URL.

Opening Links in a New Tab

Use the `target="_blank"` attribute to open the link in a new tab.

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

Linking to Other Pages (Internal Links)

You can also link to other pages of your own website.

```
<a href="about.html">About Us</a>
```

Link to a Section on the Same Page

Use the `id` attribute and a hash (`#`) to jump to a section.

```
<a href="#contact">Go to Contact</a>

...

<h2 id="contact">Contact Section</h2>
```

Email and Phone Links

- **Email Link:**

```
<a href="mailto:hello@example.com">Send Email</a>
```

- **Phone Link:**

```
<a href="tel:+1234567890">Call Us</a>
```

Styling Links (Default Behavior)

- **Normal**: Blue and underlined
- **Visited**: Purple
- **Hover**: Changes color when mouse is over it
- **Active**: Red while clicking

These styles can be changed with CSS later.

Images in HTML

Adding Images

Use the `<img>` tag to display images in HTML.
It is a **self-closing tag**, meaning it doesn't need a closing `</img>`.

Basic Syntax:

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

- `src` (source): The path or URL to the image file.
- `alt` (alternative text): Text shown if the image doesn't load, also used by screen readers.

Example with Local Image

```
<img src="my-photo.jpg" alt="My Photo">
```

(Assumes `my-photo.jpg` is in the same folder as your HTML file)

Example with Online Image

```
<img src="https://example.com/image.png" alt="Online Image">
```

Image Size: Width and Height

You can set the size of the image using `width` and `height` attributes.

```
<img src="pic.jpg" alt="Sample" width="200" height="150">
```

You can also use CSS later for better control.

Tip

- Always include the `alt` text for accessibility.
- Use proper image sizes to improve page loading speed.
- Avoid stretching images using incorrect width/height ratios.

Lists in HTML

Types of Lists

HTML supports three main types of lists:

1. **Unordered List** – Bulleted list
2. **Ordered List** – Numbered list
3. **Description List** – List of terms and their descriptions

Unordered List

Use the `<ul>` tag for unordered lists. Each item goes inside an `<li>` tag.

```
<ul>
  <li>Apples</li>
  <li>Bananas</li>
  <li>Oranges</li>
</ul>
```

This will display:

- Apples
- Bananas
- Oranges

Ordered List

Use the `<ol>` tag for ordered lists.

```
<ol>
  <li>Wake up</li>
  <li>Brush teeth</li>
  <li>Go to work</li>
</ol>
```

This will display:

1. Wake up
2. Brush teeth
3. Go to work

Description List

Use the `<dl>` tag for description lists. Terms go inside `<dt>`, and descriptions go inside `<dd>`.

```
<dl>
  <dt>HTML</dt>
  <dd>A markup language for creating web pages.</dd>

  <dt>CSS</dt>
  <dd>Used for styling HTML pages.</dd>
</dl>
```

Nesting Lists

You can put one list inside another.

```
<ul>
  <li>Fruits
    <ul>
      <li>Apple</li>
      <li>Mango</li>
    </ul>
  </li>
  <li>Vegetables</li>
</ul>
```

Tip

- Use unordered lists for things without order (like a shopping list).
- Use ordered lists when the **sequence matters**.
- Use description lists for definitions or Q&A-style content.

Tables in HTML

Creating Tables

Use the `<table>` tag to create tables. Inside a table, use:

- `<tr>` for table rows
- `<td>` for table data (cells)
- `<th>` for table headers

Basic Table Example

```
<table>
  <tr>
    <th>Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Alice</td>
    <td>24</td>
  </tr>
  <tr>
    <td>Bob</td>
    <td>30</td>
  </tr>
</table>
```


This will display:

Name	Age
Alice	24
Bob	30

Adding Borders

By default, tables have no border. Use the `border` attribute to add one.

```
<table border="1">
  ...
</table>
```

Table Headings vs Data

- `<th>` is usually bold and centered.
- `<td>` is regular table data.

Spanning Columns and Rows

Use `colspan` and `rowspan` to merge cells.

Column Span Example:

```
<tr>
  <th colspan="2">Employee Details</th>
</tr>
```

Row Span Example:

```
<tr>
  <td rowspan="2">John</td>
  <td>Manager</td>
</tr>
<tr>
  <td>IT Department</td>
</tr>
```

Tip

- Keep your tables organized and easy to read.
- Use headers (`<th>`) for important rows or columns.
- Avoid very complex tables for layout—use CSS for page design.

HTML Forms: Inputs, Labels, and Buttons

What is a Form?

Forms allow users to **input data** and send it to a server.
Use the `<form>` tag to create a form.

```
<form>
  <!-- form elements go here -->
</form>
```

Text Input Field

Use `<input type="text">` to get a single line of text from the user.

```
<form>
  <label for="name">Name:</label>
  <input type="text" id="name" name="name">
</form>
```

- `<label>` is used to describe the input.
- The `for` attribute should match the input's `id`.

Password Field

```
<label for="password">Password:</label>
<input type="password" id="password" name="password">
```

This hides the characters as you type.

Submit Button

Use `type="submit"` to create a button that submits the form.

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

Placeholder Text

You can show a hint inside the input using the `placeholder` attribute.

```
<input type="text" placeholder="Enter your name">
```

Complete Example

```
<form>
  <label for="email">Email:</label>
  <input type="email" id="email" name="email"
placeholder="you@example.com"><br><br>

  <label for="pass">Password:</label>
  <input type="password" id="pass" name="pass"><br><br>
```

```
  <input type="submit" value="Login">
</form>
```

Tip

- Always label your inputs for better accessibility.
- Use `name` attributes if the form is going to submit data.
- You'll learn more input types in the next lesson.

Form Elements: Radio, Checkbox, Select, Textarea

Radio Buttons

Use radio buttons when users need to **select only one** option from a group.

```
<p>Choose your gender:</p>
<input type="radio" id="male" name="gender" value="male">
<label for="male">Male</label><br>

<input type="radio" id="female" name="gender" value="female">
<label for="female">Female</label>
```

- All radio buttons in a group should have the **same** `name`.
- Only one option can be selected.

Checkboxes

Use checkboxes when users can **select multiple** options.

```
<p>Select your hobbies:</p>
<input type="checkbox" id="reading" name="hobby" value="reading">
<label for="reading">Reading</label><br>

<input type="checkbox" id="sports" name="hobby" value="sports">
<label for="sports">Sports</label><br>
```

```
<input type="checkbox" id="music" name="hobby" value="music">
<label for="music">Music</label>
```

Dropdown List (Select Menu)

Use the `<select>` tag with `<option>` to let users pick one option from a dropdown.

```
<label for="city">Choose a city:</label>
<select id="city" name="city">
  <option value="delhi">Delhi</option>
  <option value="mumbai">Mumbai</option>
  <option value="bangalore">Bangalore</option>
</select>
```

Textarea (Multiline Input)

Use the `<textarea>` tag to let users type multiple lines of text.

```
<label for="message">Your Message:</label><br>
<textarea id="message" name="message" rows="4" cols="30"></textarea>
```

Tip

- Use radio buttons for single choice questions.
- Use checkboxes for multiple selections.
- Use `select` for compact dropdowns.
- Use `textarea` for longer user input like comments or messages.

HTML5 Semantic Tags

Inline vs Block Elements in HTML

In HTML, elements are broadly categorized as **inline** or **block** based on how they behave in the document flow.

Block Elements

- Start on a **new line**.
- Take up the **full width** available.
- Can contain **other block** and **inline** elements.

Common Block Elements:

- `<div>`
- `<p>`
- `<h1>` to `<h6>`
- `<section>`
- `<article>`
- `<ul>`, `<ol>`, `<li>`

Example:

```
<div>
  <h2>This is a heading</h2>
  <p>This is a paragraph inside a div.</p>
</div>
```

Inline Elements

- Do **not start on a new line**.

- Only take up as much **width as necessary**.
- Usually used to style **small portions** of content within block elements.

Common Inline Elements:

- `<span>`
- `<a>`
- `<strong>`, `<em>`
- `<img>`
- `<code>`

Example:

```
<p>This is a <strong>bold</strong> word and <a href="#">this is a link</a>.</p>
```

Summary

Feature	Block Elements	Inline Elements
New Line	Yes	No
Width	Full width	Width of content only
Nesting	Can contain any elements	Only other inline elements

What Are Semantic Tags?

Semantic tags clearly describe the **meaning** of the content they contain.
They help both developers and browsers understand the structure of the page.

Example:
- `<div>` says nothing about its content.
- `<header>` clearly means it's a page or section header.

Common Semantic Tags

Tag	Purpose
`<header>`	Top section of a page or section
`<nav>`	Navigation links
`<main>`	Main content of the page
`<section>`	A standalone section
`<article>`	Self-contained content like a blog
`<aside>`	Sidebar or extra info
`<footer>`	Bottom section of a page or section

Example Usage

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

    <header>
      <h1>My Website</h1>
    </header>

    <nav>
      <a href="#">Home</a> |
      <a href="#">About</a> |
      <a href="#">Contact</a>
    </nav>

    <main>
      <section>
```

```
        <h2>Welcome</h2>
        <p>This is the welcome section.</p>
      </section>

      <article>
        <h2>Blog Post</h2>
        <p>This is a blog post inside an article tag.</p>
      </article>
    </main>

    <aside>
      <p>This is a sidebar with related links.</p>
    </aside>

    <footer>
      <p>Copyright © 2025</p>
    </footer>

  </body>
</html>
```

Tip

- Semantic tags improve accessibility and SEO.
- Use them instead of generic `<div>` and `<span>` wherever possible.

HTML Entities and Special Characters

What Are HTML Entities?

Some characters have **special meaning** in HTML (like `<`, `>`, `&`).
To display these characters on a webpage, you need to use **HTML entities**.

An entity starts with `&` and ends with `;`.

Common HTML Entities

Character	Entity Code	Description
<	`<`	Less than
>	`>`	Greater than
&	`&`	Ampersand
"	`&quot;`	Double quote
'	`&apos;`	Single quote
©	`&copy;`	Copyright symbol
®	`&reg;`	Registered symbol
₹	`&#8377;`	Indian Rupee sign
→	`&rarr;`	Right arrow

Example

```
<p>5 &lt; 10</p>
<p>Use &amp; to join strings</p>
<p>Price: &#8377;499</p>
```

This will display as:

5 < 10 Use & to join strings Price: ₹499

Non-Breaking Space

Use ` ` to add extra space that the browser won't collapse.

```
<p>Hello   World</p>
```

Tip

- Use entities when you want to **show special characters** as text.
- HTML automatically converts most symbols when needed, but using entities ensures correct display.

Audio and Video Embedding in HTML

Embedding Audio

Use the `<audio>` tag to add sound or music to your webpage.

Basic Example:

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

- `controls` adds play, pause, and volume controls.
- The `<source>` tag specifies the audio file and type.

Audio Formats

Format	MIME Type
MP3	`audio/mpeg`
OGG	`audio/ogg`
WAV	`audio/wav`

To support all browsers, you can include multiple sources:

```
<audio controls>
  <source src="audio.mp3" type="audio/mpeg">
```

```
  <source src="audio.ogg" type="audio/ogg">
</audio>
```

Embedding Video

Use the `<video>` tag to add videos to your page.

Basic Example:

```
<video width="320" height="240" controls>
  <source src="video.mp4" type="video/mp4">
  Your browser does not support the video tag.
</video>
```

- `width` and `height` control the video size.
- `controls` adds playback controls.

Video Formats

Format	MIME Type
MP4	`video/mp4`
WebM	`video/webm`
OGG	`video/ogg`

You can include multiple sources to ensure browser compatibility:

```
<video controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
</video>
```

Tip

- Always provide `controls` so users can interact with media.
- Use multiple formats for broader browser support.
- Include fallback text for unsupported browsers.

IFrames and Embedding Content

What is an IFrame?

An `<iframe>` (inline frame) is used to embed another webpage or external content inside your HTML page.

Basic Syntax

```
<iframe src="https://example.com" width="600" height="400"></iframe>
```

- `src` specifies the URL of the page to embed.
- `width` and `height` set the size of the frame.

Example: Embed a Website

```
<iframe src="https://www.wikipedia.org" width="800" height="500"></iframe>
```

Example: Embed a YouTube Video

YouTube provides embed code for each video.

```
<iframe width="560" height="315"
  src="https://www.youtube.com/embed/dQw4w9WgXcQ"
```

```
  frameborder="0"
  allowfullscreen>
</iframe>
```

Attributes

Attribute	Description
`src`	URL of the page or content
`width , height`	Size of the iframe
`frameborder`	Border of the frame (0 = none)
`allowfullscreen`	Allows video to go full screen
`loading="lazy"`	Delays loading until iframe is visible

Security Note

Some websites may **block iframe embedding** for security reasons using headers like `X-Frame-Options` .

Tip

- Use `<iframe>` to embed maps, videos, forms, and external tools.
- Always set appropriate width and height for better layout control.

Using Meta Tags and SEO Basics

What Are Meta Tags?

Meta tags provide **information about the webpage** to browsers and search engines.
They go inside the `<head>` section and do not appear on the page itself.

Common Meta Tags

1. Charset

```
<meta charset="UTF-8">
```

- Defines the character encoding.
- UTF-8 covers most characters in all languages.

---

2. Viewport (Mobile Responsiveness)

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

- Makes your website mobile-friendly.
- Tells the browser to match the screen's width.

---

3. Page Description

```
<meta name="description" content="Learn HTML from scratch with simple examples.">
```

- Summarizes the page content.
- Often shown in search engine results.

---

4. Keywords (Less important today)

```
<meta name="keywords" content="HTML, web development, coding">
```

- List of keywords related to your page.
- Not heavily used by modern search engines.

---

5. Author

```
<meta name="author" content="Your Name">
```

- Specifies the name of the content creator.

---

6. Refresh / Redirect (Optional)

```
<meta http-equiv="refresh" content="5; url=https://example.com">
```

- Redirects the page after 5 seconds.

SEO Basics

- Use meaningful page titles with the `<title>` tag.

- Include a clear meta description.
- Structure content using headings (`<h1>` , `<h2>` , etc.).
- Use semantic tags to describe content.
- Make sure the page loads fast and works well on mobile.

Example Head Section

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="description" content="Simple HTML tutorial for beginners.">
  <meta name="author" content="John Doe">
  <title>Learn HTML</title>
</head>
```

Tip

- Good meta tags improve search visibility and user experience.
- Always include the viewport meta tag for mobile responsiveness.

Internal vs External Links

Internal Links

Internal links connect one page of your website to another.
They help users **navigate** within your site.

Example:

```
<a href="about.html">About Us</a>
```

This opens the `about.html` page located in the same folder.

Linking to a Section on the Same Page

Use `#id` to jump to a specific section.

```
<a href="#contact">Go to Contact Section</a>

...

<h2 id="contact">Contact Us</h2>
```


External Links

External links take the user to a **different website**.

Example:

```
<a href="https://www.google.com">Visit Google</a>
```

Open External Links in a New Tab

Use the `target="_blank"` attribute.

```
<a href="https://www.example.com" target="_blank">Open Example</a>
```

Adding `rel="noopener noreferrer"` for Security

When using `target="_blank"`, it's recommended to add `rel="noopener noreferrer"` to prevent security risks.

```
<a href="https://external.com" target="_blank" rel="noopener noreferrer">
  Visit External Site
</a>
```

Summary

Link Type	Example
Internal Link	`href="about.html"`
Section Jump	`href="#section-id"`
External Link	`href="https://example.com"`
New Tab + Safe	`target="_blank" rel="noopener"`

Tip

- Use internal links to connect your content and improve navigation.
- Use external links to reference useful outside resources.

Best Practices for Writing Clean HTML

1. Use Proper Indentation

Indent nested elements for better readability.

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
</ul>
```

2. Always Close Your Tags

Even if some tags are optional, it's best to close them properly.

```
<p>This is correct.</p>
```

3. Use Meaningful Tag Structure

Use semantic tags like `<header>`, `<main>`, `<footer>` instead of relying only on `<div>`.

4. Include `alt` Text for Images

This improves accessibility and helps screen readers understand image content.

```
<img src="photo.jpg" alt="A smiling person">
```

---

5. Use Lowercase for Tags and Attributes

HTML is not case-sensitive, but using lowercase is the standard.

```
<!-- Good -->
<input type="text">

<!-- Avoid -->
<INPUT TYPE="TEXT">
```

---

6. Organize Your Code

Keep your HTML structured by grouping related elements together. Use comments to separate sections.

```
<!-- Navigation -->
<nav>...</nav>

<!-- Main Content -->
<main>...</main>
```


---

7. Don’t Use Inline Styles (if possible)

Avoid putting CSS styles directly into HTML tags. Use external CSS files instead.

```
<!-- Avoid -->
<p style="color: red;">Red text</p>

<!-- Prefer -->
<p class="red-text">Red text</p>
```

---

8. Validate Your HTML

Use tools like W3C HTML Validator to check for errors in your code.

---

9. Keep File Names Simple and Clear

Use lowercase letters, dashes instead of spaces, and meaningful names.

```
✔ about-us.html
✘ About Us!.html
```

---

10. Comment Your Code (When Needed)

Use comments to explain complex sections or to label page areas.

```
<!-- Contact Form -->
<form>...</form>
```

Tip

Clean HTML is easier to read, debug, maintain, and scale as your website grows.