

HTML

HTML stands for HyperText Markup Language.

It is the basic language used to create websites.

HTML gives structure and meaning to web content.

What does HyperText mean?

HyperText means clickable links.

These links help you move from one page to another on the internet.

For example,

when you click on a word or picture and it takes you to a new page — that's HyperText!

What is Markup?

Markup means adding labels to parts of a web page.

It tells the browser what each part is, like a heading, paragraph, or image.

HTML uses tags (like `<p>` or `<h1>`) to do this.

These tags help the web browser know how to show the content.

What are HTML Tags?

Tags look like this: `<tagname>`.

Example: `<p>` for a paragraph.

Most tags come in pairs:

Opening tag: `<p>`

Closing tag: `</p>`

Tags are not case-sensitive, but it's best to write them in lowercase.

Common HTML Elements

<head> – Info about the page

<title> – Page title (seen on browser tab)

<body> – Main content

<p> – Paragraph

<div> – Container for content

 – Image

<a> – Link

, , – Lists (unordered, ordered, list items)

<video>, <audio> – Media

<!DOCTYPE html>

This line is called the DOCTYPE declaration.

It tells the browser that the page is written in HTML5 (the latest version of HTML).

It must be the first line in every HTML document.

It helps browsers display the page correctly.

What is Boilerplate Code in HTML?

Boilerplate code is the basic structure of an HTML page.

You use it as a starting point for every new webpage.

Explanation of Each Part:

<!DOCTYPE html>

This tells the browser:

"This is an HTML5 document."

<html>

The root element.

All content of the web page goes inside this.

<head>

Contains information about the page (not shown to users).

Examples: title, links to CSS, meta tags.

<title>My First Page</title>

Sets the title shown in the browser tab.

</head>

<body>

Everything inside <body> is visible on the webpage.

You put text, images, buttons, etc., here.

<h1>Hello, world!</h1>

This is a main heading.

It appears big and bold on the page.

</body>

</html>

Closes the <body> and <html> tags.

This is the end of the HTML page.

HTML Comments

Used to add notes in the code.

Not shown on the webpage.

Example:

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

Attributes in HTML

Attributes give extra information about an element.

Written inside the opening tag.

Example:

```

```

src = image source

alt = alternative text

HTML + CSS + JavaScript

HTML = Structure

CSS = Appearance (colors, layout)

JavaScript = Behavior (what the page does)

Forms and Media

Forms help with actions like login, register, or send info.

HTML supports media like audio and video using tags like <audio> and <video>.

Block vs Inline Elements

Block-level: Takes full width (e.g., <div>, <p>)

Inline level: Only takes up as much space as needed (e.g., , <a>)

Extra Tips

Use <link rel="preload"> to load important files early.

Use responsive images for better performance on different devices.

HTML Text Formatting Tags – Summary with Examples

Tag	Use	Example	output
<p>	Paragraph	<p>This is a paragraph.</p>	This is a paragraph.
 	Line break	Line one Line two	Line one Line two
<hr>	Horizontal line	<hr>	-----
<h1>to<h6>	Headings (big to small)	<h2>Subheading</h2>	Subheading
	Bold (important)	This is important	This is important
	Bold (styling only)	This is bold	This is bold
	Italic (emphasis)	This is emphasized	This is <i>emphasized</i>
<i>	Italic (styling)	This is <i>italic</i>	This is <i>italic</i>
<u>	Underline	This is <u>underlined</u>	This is <u>underlined</u>
<mark>	Highlight	This is <mark>highlighted</mark>	This is highlighted
<small>	Smaller text	This is <small>small text</small>	This is small text
	Strikethrough means drawing a horizontal line through text to show that it has been removed	Old Price	Old Price
<ins>	Inserted (underlined)	<ins>New Text</ins>	<u>New Text</u>
<sub>	Subscript	H₂O	H ₂ O
<sup>	Superscript	X²	X ²
<code>	Code snippet	Use <code> alert () </code>	Use alert ()
<pre>	Preformatted text	<pre>Line 1\n Line2</pre>	Line 1 Line 2
<kbd>	Keyboard input	<kbd>Ctrl</kbd> + <kbd>C</kbd>	Ctrl + C
<samp>	Program output	<samp>Error404</samp>	Error 404
<var>	Variable name	<var>x</var> = 5	x = 5

<q>	Short quote	<q>Hello</q>	“Hello”
<blockquote>	Long quote	<blockquote>This is quoted</blockquote>	(Indented quote)
<abbr>	Abbreviation with tooltip	<abbr title="HyperText Markup Language">HTML</abbr>	HTML (hover shows full)
<time>	Time/date info	<time datetime="2025-07-17">July 17</time>	July 17
<cite>	Cites a source	<cite>Wikipedia</cite>	Wikipedia
<dfn>	Defines a term	<dfn>HTML</dfn> is a markup language.	HTML is a markup language.

✅ HTML Lists — Summary with Examples

1. Ordered List ()

A list with numbers or letters.

Used when the order matters (steps, rankings).

Example:

```
<ol>
```

```
<li>First item</li>
```

```
<li>Second item</li>
```

```
<li>Third item</li>
```

```
</ol>
```

Output:

1.First item

2.Second item

3.Third item

2. Unordered List ()

A list with bullet points.

Used when order doesn't matter.

Example:

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

Output:

```
.Apples  
.Oranges  
.Bananas
```

3. Description List (<dl>)

A list of terms and their descriptions.

Terms use <dt> and descriptions use <dd>.

Example:

```
<dl>  
  <dt>HTML</dt>  
  <dd>HyperText Markup Language</dd>  
  <dt>CSS</dt>  
  <dd>Cascading Style Sheets</dd>  
</dl>
```

Output:

HTML

HyperText Markup Language

CSS

Cascading Style Sheets

Bonus: Nesting Lists

You can put one list inside another.

Example:

```
<ul>
```

```
<li>Fruits
```

```
<ul>
```

```
<li>Apple</li>
```

```
<li>Banana</li>
```

```
</ul>
```

```
</li>
```

```
<li>Vegetables</li>
```

```
</ul>
```

Output:

Fruits

Apple

Banana

Vegetables

HTML Links

What is the <a> tag?

The <a> tag defines a hyperlink in HTML.

It's used to link from one page to another, to different parts of the same page, or to external resources like email or phone.

The href attribute inside <a> specifies the destination URL or link target.

Example:

1. Basic Link

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

href = "Hypertext Reference" — tells where the link goes.

The text between <a> and is what the user clicks on.

Clicking "Visit Example" opens <https://www.example.com>.

2. Open Link in New Tab

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

target="_blank" opens the link in a new browser tab.

3. Email Link

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

Clicking opens your email client to send an email to someone@example.com.

The mailto: link is used in an <a> tag to let users send an email by opening their default email app (like Gmail, Outlook, or Mail)

Opens the user's email client to send an email to someone@example.com.



Note: Works only if an email client is set up on the device.

4. Phone Link

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

On mobile or supported devices, this starts a phone call to the number.

5. Link with Tooltip

```
<a href="https://www.example.com" title="Visit Example Site">Hover me!</a>
```

When you hover over “Hover me!”, a small tooltip shows “Visit Example Site”.

6. Link that Downloads a File

```
<a href="file.pdf" download>Download PDF</a>
```

Instead of opening file.pdf, the browser downloads it when clicked.

Images in HTML

Use the tag to display images. It is a self-closing tag and has no end tag.

◆ Syntax:

```

```

◆ Attributes:

src: (required) URL or path of the image file.

alt: (recommended) Text shown if the image cannot be displayed.

width & height: (optional) Specify the size of the image.

✅ Example:

```

```

◆ Output:

Displays an image with the specified size and an alternate text if it fails to load.

Tables in HTML

Tables are created using the <table> element. Rows are defined by <tr>, headers by <th>, and data cells by <td>.

◆ Basic Structure:

```
<table>

<tr>

  <th>Header 1</th>

  <th>Header 2</th>

</tr>

<tr>

  <td>Row1 Cell1</td>

  <td>Row1 Cell2</td>

</tr>

<tr>

  <td>Row2 Cell1</td>

  <td>Row2 Cell2</td>

</tr>

</table>
```

◆ Description:

<table>: Wraps the whole table.

<tr> (table row): Defines a row.

<th> (table heading): Defines a heading (bold & centered).

<td> (table data): Defines a standard data cell.

◆ **Output:**

Header 1	Header 2
Row1 Cell1	Row1 Cell2
Row2 Cell1	Row2 Cell2

Forms in HTML

HTML forms are used to collect user input.

1. <form>

The <form> tag wraps all the input elements.

◆ **Syntax:**

```
<form action="submit.php" method="post">  
  <!-- form elements go here -->  
</form>
```

◆ **Attributes:**

action: Where to send form data (usually a backend file like .php or .js).

method:

"get": Appends data in URL (for simple searches).

"post": Sends data securely (for login, signup, etc.).

2. <input>

Used to take various types of input.

◆ **Common Types:**

```
<input type="text" name="username" placeholder="Enter username">  
<input type="email" name="email" placeholder="Enter email">  
<input type="password" name="password" placeholder="Enter password">  
<input type="checkbox" name="subscribe"> Subscribe
```

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

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

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

◆ **Attributes:**

type: text, email, password, checkbox, radio, submit, etc.

name: the key used when sending data.

value: the value that gets sent to server.

placeholder: shows light grey hint text inside the box.

3. **<textarea>**

For multiline text input (like comments or messages).

◆ **Syntax:**

```
<textarea name="message" rows="4" cols="30" placeholder="Type your  
message"></textarea>
```

4. **<select> and <option>**

Dropdown list with multiple choices.

◆ **Syntax:**

```
<select name="country">
```

```
<option value="india">India</option>
```

```
<option value="usa">USA</option>
```

```
<option value="uk">UK</option>
```

```
</select>
```

5. <button>

Clickable button for submitting or other actions.

◆ Syntax:

```
<button type="submit">Submit</button>
```

```
<button type="reset">Reset</button>
```

✅ Full Example Form:

```
<form action="/submit" method="post">
```

```
<label>Username:</label>
```

```
<input type="text" name="username" placeholder="Enter username"><br><br>
```

```
<label>Email:</label>
```

```
<input type="email" name="email" placeholder="Enter email"><br><br>
```

```
<label>Message:</label><br>
```

```
<textarea name="message" rows="4" cols="30"></textarea><br><br>
```

```
<label>Gender:</label>
```

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

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

```
<label>Country:</label>
```

```
<select name="country">
```

```
<option value="india">India</option>
```

```
<option value="usa">USA</option>
```

```
</select><br><br>
```

```
<button type="submit">Send</button>
```

```
</form>
```

What is Semantic HTML?

Semantic HTML uses meaningful tags that clearly describe their purpose in the page.

It helps search engines, screen readers, and developers understand the structure of a webpage.

Common Semantic Tags:

1. <header>

Defines the top section of a webpage or a section. Usually includes logos, titles, or navigation links.

◆ **Example:**

```
<header>  
  
  <h1>My Website</h1>  
  
  <p>Welcome to my world</p>  
  
</header>
```

2. <nav>

Defines the navigation bar. It holds links to other pages or sections.

◆ **Example:**

```
<nav>  
  
  <a href="/">Home</a>  
  
  <a href="/about">About</a>  
  
  <a href="/contact">Contact</a>  
  
</nav>
```

3. <section>

Defines a thematic section of content. Like a chapter or group of related content.

◆ **Example:**

```
<section>

<h2>Our Services</h2>

<p>We offer web development, design, and more.</p>

</section>
```

4. <article>

Used for independent content like blog posts, news articles, or user comments.

◆ **Example:**

```
<article>

<h2>How to Learn HTML</h2>

<p>Start with basic tags and build projects.</p>

</article>
```

5. <aside>

Defines side content like ads, tips, or extra info. Often used in sidebars.

◆ **Example:**

```
<aside>

<h3>Tip:</h3>

<p>Use semantic tags to improve SEO.</p></aside>
```

6. <footer>

Defines the bottom section of a page or section. Usually contains copyright, contact, or links.

◆ **Example:**

```
<footer>

<p>&copy; 2025 My Website. All rights reserved.</p>

</footer>
```


✅ Full Example Using Semantic Tags:

```
<header>
```

```
  <h1>My Blog</h1>
```

```
  <nav>
```

```
    <a href="#">Home</a>
```

```
    <a href="#">Posts</a>
```

```
    <a href="#">About</a>
```

```
  </nav>
```

```
</header>
```

```
<section>
```

```
  <article>
```

```
    <h2>Semantic HTML</h2>
```

```
    <p>Semantic tags make your HTML meaningful and easier to read.</p>
```

```
  </article>
```

```
</section>
```

```
<aside>
```

```
  <p>Subscribe for more HTML tips!</p>
```

```
</aside>
```

```
<footer>
```

```
  <p>Copyright &copy; 2025 My Blog</p>
```

```
</footer>
```

<audio> Tag – For Playing Sound

The <audio> tag is used to play audio files like music or voice.

◆ Basic Syntax:

<audio controls>

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

Your browser does not support the audio element.

</audio>

◆ Attributes:

controls: shows play, pause, volume buttons.

autoplay: plays the audio automatically when the page loads.

loop: repeats the audio after it ends.

<video> Tag – For Playing Videos

The <video> tag is used to play video files.

◆ Basic Syntax:

<video width="320" height="240" controls>

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

Your browser does not support the video tag.

</video>

◆ Attributes:

controls: shows play, pause, volume, etc.

autoplay: plays the video automatically.

loop: repeats the video.

poster: image shown before video plays.

muted: video starts with no sound.

<source> Tag – For Multiple Media Formats

The <source> tag is used inside <audio> or <video> to define media files and their formats.

◆ Why use <source>?

Some browsers support only certain file types.

You can give multiple formats — the browser picks the one it supports.

◆ Example:

<video controls>

<source src="video.mp4" type="video/mp4">

<source src="video.ogg" type="video/ogg">

Your browser does not support this video.

</video>

✅ Summary Table

Tag	Use for Inside	Notes
<audio>	Sound/music	Needs <source> or src
<video>	Video clips	Needs <source> or src
<source>	Add file formats <audio>, <video>	Use multiple formats for safety

<iframe> – Inline Frame

The <iframe> tag is used to embed another webpage inside your current webpage.

It's like a window inside your page that shows another site or file.

```
<iframe  
src="https://www.wikipedia.org"  
width="600"  
height="400"  
title="Wikipedia Page"  
frameborder="0">  
</iframe>
```

This will show Wikipedia's homepage inside your webpage in a 600x400 box.

What are **<meta>** tags?

<meta> tags give information about the webpage, called metadata.

They go inside the **<head>** tag and are not visible on the page.

They help:

Browsers understand how to show the page

Search engines know what your page is about

Responsive design on mobile devices

✅ Common **<meta>** Tags:

1. **<meta charset="UTF-8">**

Sets the character encoding — how text is stored and displayed.

♦ Example:

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

✅ UTF-8 supports all languages and symbols (safe default for all websites).

2. <meta name="viewport">

Makes your website responsive (mobile-friendly).

♦ Example:

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

✅ This makes your page fit the screen on phones and tablets.

3. <meta name="description">

Gives a short summary of your webpage.

This may show up in Google search results.

♦ Example:

```
<meta name="description" content="This is a blog about HTML tutorials in simple language.">
```

✅ Helps improve SEO (search engine optimization).

4. <meta name="keywords"> (less used today)

A list of words related to your page (for search engines).

Most modern search engines ignore this tag now.

♦ Example:

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

5. <meta name="author">

Tells who created the webpage.

♦ Example:

```
<meta name="author" content="Akash Shinde">
```

6. <meta http-equiv="refresh">

Refreshes the page after some time or redirects to another URL.

- ◆ Example (refresh after 5 seconds):

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

- ◆ Example (redirect to another page after 3 seconds):

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

✓ Full Example in <head>:

```
<head>
```

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

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

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

```
<meta name="author" content="Akash Shinde">
```

```
<title>My HTML Page</title>
```

```
</head>
```

● 3. Advanced HTML Concepts

✓ 1. Accessibility (a11y)

Accessibility means making your website usable for everyone, including people with disabilities.

◆ Best Practices:

Use proper alt text for images.

Use semantic HTML (<header>, <nav>, etc.).

Use labels for form elements.

Ensure good keyboard navigation (Tab, Enter, etc.).

Use proper color contrast and font sizes.

✅ 2. ARIA Roles (Accessible Rich Internet Applications)

ARIA helps screen readers understand elements better when HTML alone is not enough.

◆ Example:

```
<div role="navigation">Menu</div>
```

```
<div role="button" tabindex="0">Click me</div>
```

ARIA Role	Meaning
-----------	---------

role="button"	Acts like a button
---------------	--------------------

role="navigation"	Marks navigation section
-------------------	--------------------------

role="dialog"	Marks popup dialogs
---------------	---------------------

Use ARIA only when needed. Prefer semantic HTML first.

✅ 3. Using Proper Heading Structure

Headings (<h1> to <h6>) should follow a logical order, like an outline.

✅ Good Example:

```
<h1>Main Title</h1>
```

```
<h2>Section 1</h2>
```

```
<h3>Subsection</h3>
```

```
<h2>Section 2</h2>
```

❌ Avoid skipping levels (like jumping from <h1> to <h4>).

✅ 4. Labeling Form Elements

Use <label> to connect text with form inputs, so screen readers can read them.

◆ Example:

```
<label for="email">Email:</label>
```

```
<input type="email" id="email" name="email">
```

✅ Improves usability and accessibility.

✅ 5. HTML APIs (with JavaScript)

These allow HTML + JavaScript to add interactive features.

● a) Drag and Drop API

```
<div draggable="true">Drag me</div>
```

You use JS events like `ondragstart`, `ondrop`, etc., to move elements around.

● b) Geolocation API

Get the user's current location.

```
navigator.geolocation.getCurrentPosition((position) => {  
  console.log(position.coords.latitude, position.coords.longitude); });
```

● c) Web Storage API

Stores data in the browser:

Type	Description
<code>localStorage</code>	Saves data permanently
<code>sessionStorage</code>	Saves data until tab is closed

```
localStorage.setItem("username", "Akash");  
  
const user = localStorage.getItem("username"); // Akash
```

✅ 6. Microdata / Schema.org

Helps search engines understand your content using special attributes.

◆ Example:

```
<div itemscope itemtype="https://schema.org/Person">  
  <span itemprop="name">Akash Shinde</span>  
  <span itemprop="jobTitle">Web Developer</span>  
</div>
```


- ✔ Improves SEO and enables rich results (like reviews, events).

✔ 7. HTML5 Form Validation

Built-in way to check form fields without JavaScript.

◆ Common Attributes:

Attribute	Purpose	Example
required	Field must be filled	<input required>
pattern	Must match a regex pattern	<input pattern="[A-Za-z]{3,}">
min/max	Numeric/date input range	<input type="number" min="1" max="10">
type	Checks input type like email, number	<input type="email">

✔ Example:

```
<form>

<input type="email" required>

<input type="number" min="1" max="100">

<input type="text" pattern="[A-Za-z]{3,}">

<button type="submit">Submit</button>

</form>
```

✔ Developer Tools (Chrome DevTools)

◆ What it is:

Chrome DevTools is a built-in tool in Chrome (press F12 or right-click → "Inspect")

It helps debug, edit, and test your HTML, CSS, and JavaScript directly in the browser.

◆ Key Features:

Elements tab: See and edit HTML/CSS live.

Console: Run JavaScript or check errors.

Network: See how and when files load.

Responsive view: Simulate mobile/tablet screens.

✅ SEO Basics with HTML (Search Engine Optimization)

Search engines like Google use your HTML to understand your site.

◆ Key HTML Tips for SEO:

Use semantic tags (<header>, <article>, <footer>)

Use <title> and <meta name="description"> in <head>

Use alt attributes on images

Use proper heading structure (<h1>, <h2>, etc.)

Link pages with meaningful anchor text

✅ Example:

```
<head>
```

```
  <title>Learn HTML in Simple Steps</title>
```

```
  <meta name="description" content="This tutorial helps you learn HTML easily.">
```

```
</head>
```

```

```

✅ Responsive Design Foundations

Making your website look good on all devices: mobile, tablet, desktop.

◆ 1. Viewport Meta Tag

Tells the browser how to scale the page on mobile.

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

♦ 2. Mobile-First Design

Start designing for small screens first

Then use media queries to adjust for larger screens

✅ Example (CSS):

```
/* Base (mobile-first) */  
  
body {  
  
    font-size: 16px;  
  
}  
  
/* For larger screens */  
  
@media (min-width: 768px) {  
  
    body {  
  
        font-size: 18px;  
  
    }  
  
}
```

✅ **Linting HTML (W3C Validator)**

♦ What is Linting?

Linting = Checking your code for errors or warnings

Use tools like:

🔗 <https://validator.w3.org/>

Just paste your HTML or URL and get a report.

✅ It helps you fix:

Missing tags

Wrong nesting

Outdated attributes

✅ Code Organization & Formatting

Keeping your HTML clean, readable, and maintainable.

◆ Best Practices:

Indent properly (usually 2 or 4 spaces)

Close all tags

Group related elements

Use comments:

```
<!-- Main Navigation -->
```

```
<nav>...</nav>
```

Break code into sections (<header>, <main>, <footer>)

Name files clearly (about.html, contact.html)

💡 Summary Checklist

Topic	Key Point
Chrome DevTools	Inspect & debug HTML/CSS/JS
SEO with HTML	Use title, description, headings, alt text
Responsive Design	Use viewport tag & mobile-first CSS
Linting HTML	Use W3C Validator to check for errors
Code Formatting	Indent, comment, structure with semantic tags