HTML

What is HTML?

- HTML (**HyperText Markup Language**) is the standard language for creating web pages.

- It defines the meaning and structure of web content.

> 💡 A markup language is a system that defines the structure of a document using tags and annotations.

Basic HTML Structure

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>HTML</title>
  </head>
  <body>
    <!-- this is a comment →
    <h1>Learning HTML with Chaicode</h1>
  </body>
</html>
```

`<!DOCTYPE html>` → Tells the browser to use HTML5 standards.

`<html>` → Indicates that the content enclosed is an HTML document.

`<head>` → Contains metadata and links to external resources.

`<body>` → Contains visible content.

Tags

- Building blocks of HTML.
- Usually come in pairs: `<opening>` and `</closing>`.

📝 **Example:** `<p>` is an opening tag for a paragraph, and `</p>` is the corresponding closing tag.

> 💡 Self-Closing tags are tags that do not require a closing tag.
>
> 📝 **Example:** `<img>` , `<br>` , `<hr>`

Elements

- An element consists of an opening tag, content, and a closing tag.
- ▼ 📝 **Code Example**

```
<p>Hello there</p>
```

Attributes

- Provide additional information about an element.
- ▼ 📝 **Code Example**

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

💡 **Global attributes** are attributes that can be applied to any HTML elements.

📝 **Example:** `id` , `class` , `title` , `style` , etc.

Common HTML Tags

Tag	Description
<h1> - **<h6>**	Define headings, with `<h1>` being the most important and `<h6>` the least.
<p>	Defines a paragraph.
** **	Inserts a line break
<hr>	Creates a horizontal rule
<b>	Bold text (no semantic meaning)
<strong>	Bold text that conveys importance
<i>	Italicizes text (no semantic meaning)
<em>	Italicized and emphasized text

💡 HTML entities are special codes used to display characters that are **reserved in HTML.**

📝 **Example:** ` ` , `<` , `©`

Block and Inline Elements

Block Elements

- Take up the entire width of their parent container.

- Always start on a new line.

- 📝 **Example:** `<div>` , `<p>` , `<h1>`

Inline Elements

- Take only as much width as necessary.

- Do not force a new line.

- 📝 **Example:** `<a>` , `<img>` , `<span>`

> 💡 `<div>` **is a** block-level container for grouping content.
>
> `<span>` **is a i**nline container for styling or grouping elements.

Links in HTML

The `<a>` tag is used to create **hyperlinks**, allowing users to navigate to other web pages, sections within a page, or external resources.

▼ **External Link**

```
<a href="https://www.chaicode.com">Visit Chaicode</a>
```

▼ Internal Link

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

▼ Email Link

```
<a href="mailto:dontmailme@gmail.com">Send an Email</a>
```

▼ Phone Call Link

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

▼ Anchor Link (Jump to Section on Same Page)

```
<!-- Link to id of target section →
<a href="#contact">Go to Contact Section</a>

<!-- Give an id to the target section →
<h2 id="contact">Contact Us</h2>
```

target attribute

- Controlling How Links Open

Value	What It Does
_self	Opens the link in the **same tab** (default behavior).
_blank	Opens the link in a **new tab**.
_parent	If inside an **iframe**, opens in the **parent window**.
_top	If inside **nested iframes**, opens in the **full window**.

▼ 📝 Code Example

```
<a href="https://www.chaicode.com" target="_blank">Visit Chaicode</a>
```

download attribute

- Downloads a file instead of opening it.

▼ 📝 **Code Example**

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

Images in HTML

- Embeds images on a webpage
- `alt` → Displays alternative text if the image fails to load.
- ▼ 📝 **Code Example**

```html
<img src="image.jpg" alt="description">
```

<figure> & <figcaption>

- Embed images with caption
- ▼ 📝 **Code Example**

```html
<figure>
    <img src="sunset.jpg" alt="Sunset over the ocean">
    <figcaption>A beautiful sunset over the ocean.</figcaption>
</figure>
```

<picture>

- Loads different images based on screen size (great for responsive design).
- ▼ 📝 **Code Example**

```
<picture>
  <!-- If the screen width is at least 800px, use the larger image →
  <source media="(min-width: 800px)" srcset="large-image.jpg">
  <!-- If the screen width is less than 800px, use the smaller image →
  <source media="(max-width: 799px)" srcset="small-image.jpg">
  <!-- Fallback image if none of the above conditions are met →
  <img src="default-image.jpg" alt="A description of the image">
</picture>
```

Videos in HTML

- The `<video>` tag embeds video files.

Common Attributes

Attribute	Function
controls	Adds play, pause, volume controls.
controlslist	Select what controls to show.
autoplay	Starts playing automatically (generally blocked by browsers).
loop	Plays the video in a loop.
muted	Mutes the video.
poster	Shows an image before video plays.

▼ 📝 **Code Example**

```
<video
    src="video.mp4"
    width="320"
    height="240"
    controls
    controlslist="nofullscreen nodownload"
    autoplay
```

```
        loop
        muted
        poster="poster.png"
    ></video>
```

Adding Subtitles

- The `<track>` element provides subtitles, captions, or descriptions for media elements like `<video>` and `<audio>`

- Each `<track>` element represents a set of subtitles or captions.

- ▼ 📝 **Code Example**

```
<video src="chaicode.mp4" controls>
    <track
      src="chaicode.en_US.vtt"
      label="English"
      kind="subtitles"
      srclang="en"
    />
    <track
      src="chaicode.hi_IN.vtt"
      label="Hindi"
      kind="subtitles"
      srclang="hi"
      default
    />
</video>
```

Multiple Media Formats

- Using multiple `<source>` elements in `<audio>` or `<video>` ensures compatibility. If a browser can't play the first format, it tries the next.

- ▼ 📝 **Code Example**

```
<video controls>
  <source src="chaicode.mp4" type="video/mp4" />
  <source src="chaicode.webm" type="video/webm" />
  Your browser does not support the video tag.
</video>
```

Audio in HTML

- The `<audio>` tag embeds an audio file.

- ▼ 📝 **Code Example**

```
<audio src="audio.mp3" controls></audio>
```

Embedding External Content

- The `<iframe>` tag is used to embed another webpage or media (e.g. YouTube videos, Google Maps, external content).

- ▼ 📝 **Code Example**

```
<iframe
  width="560"
  height="315"
  src="https://www.youtube.com/embed/KZtQQQARWQE"
  title="YouTube Video"
></iframe>
```

Lists in HTML

- Lists are used to organize and display multiple items in a structured format.

Ordered List

- Used when the order of items **matters**, such as **steps in a process**.

▼ 📝 **Code Example**

```
<ol>
 <li>Boil water</li>
 <li>Add tea leaves</li>
 <li>Serve tea</li>
</ol>
```

Unordered List

- Used when the order of items **does NOT** matter, such as a **shopping list**.

▼ 📝 **Code Example**

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

Description List

- Used for **term-description pairs**, like glossaries or FAQs.
  - `<dt>` → **Description Term**
  - `<dd>` → **Description Details**

▼ 📝 **Code Example**

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

```
<dt>CSS</dt>
<dd>A stylesheet language used for styling HTML documents.</dd>

<dt>JavaScript</dt>
<dd>A programming language that makes web pages interactive.</dd>
</dl>
```

Tables in HTML

- Tables are used to display **structured/tabular data**.

Table Elements & Attributes

Element/Attribute	Function
`<caption>`	Provides a **table title**.
`<thead>`	Groups the **header** content.
`<tbody>`	Groups the **main table data**.
`<tfoot>`	Groups the **footer** content (summary data).
`<tr>`	Defines a **table row**.
`<th>`	Represents a **header cell.**
`<td>`	Represents a **data cell**.
`rowspan`	Merges multiple **rows** into a single cell.
`colspan`	Merges multiple **columns** into a single cell.

▼ 📝 **Code Example**

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" /
  <title>HTML Table</title>
```

```html
<style>
  body {
    display: flex;
    align-items: center;
    justify-content: center;
    height: 100vh;
  }
  table {
    width: 60%;
    border-collapse: collapse;
    text-align: center;
  }
  th,
  td {
    border: 1px solid black;
    padding: 8px;
  }
  th {
    background-color: #f2f2f2;
  }
  tfoot {
    font-weight: bold;
  }
</style>
</head>
<body>
<table>
  <caption>
    Student Score Table
  </caption>
  <thead>
    <tr>
      <th rowspan="2">ID</th>
      <th rowspan="2">Name</th>
      <th colspan="2">Scores</th>
      <th rowspan="2">Total</th>
```

```html
    </tr>
    <tr>
      <th>Math</th>
      <th>Science</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Dhruv</td>
      <td>90</td>
      <td>85</td>
      <td>175</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Sahil</td>
      <td>88</td>
      <td>92</td>
      <td>180</td>
    </tr>
    <tr>
      <td>3</td>
      <td>Rahul</td>
      <td>76</td>
      <td>80</td>
      <td>156</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td colspan="4">Average Score</td>
      <td>170</td>
    </tr>
  </tfoot>
</table>
```

```
    </body>
  </html>
```

Forms in HTML

- Forms are used to **collect user input** and send it to a server.

Basic Form Structure

```
<form action="/submit" method="post">
  <!-- Form content →
</form>
```

`action` : Specifies the URL where the form data is sent

`method` : Defines the HTTP method (`GET` or `POST`) used to send the data.

Common Form Elements

`<input>`

- Used to collect user input.
- ▼ 📝 **Code Example**

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

💡 When the form is submitted, the value of input with a `name` attribute are sent as key-value pairs.

<textarea>

- Used for **multi-line** text input.
- ▼ 📝 **Code Example**

```html
<textarea
  name="message"
  rows="4"
  cols="50"
  placeholder="Enter your message"
></textarea>
```

<select>

- Creates a **dropdown list** using `<option>` .
- ▼ 📝 **Code Example**

```html
<select name="country">
  <option value="in">India</option>
  <option value="us">United States</option>
  <option value="ca">Canada</option>
</select>
```

> 💡 The `value` attribute determines what data is sent to the server.

<button>

- Used to **submit or reset** the form.

  - `submit` : Submits the form.

  - `reset` **:** Clears all input fields.

- ▼ 📝 **Code Example**

```html
<button type="submit">Submit</button>
<button type="reset">Reset</button>
```

<label>

- Associates text with form elements.
- ▼ 📝 **Code Example**

```html
<label for="username">Username:</label>
<input type="text" id="username" name="username">
```

> 💡 The `for` attribute links the label to an input element by its `id` .

Common Input types

type="text"

- For single-line text input.
- ▼ 📝 **Code Example**

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

type="email"

- For email addresses.
- ▼ 📝 **Code Example**

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

type="password"

- For passwords, hides the entered text.
- ▼ 📝 **Code Example**

```html
<input type="password" name="password" placeholder="Enter password">
```

type="checkbox"

- Allows users to select multiple options.
- ▼ 📝 **Code Example**

```html
<form>
 <fieldset>
  <legend>Select your interests:</legend>

  <label>
   <input type="checkbox" name="interests" value="coding" />
   Coding
  </label>

  <label>
   <input type="checkbox" name="interests" value="music" />
   Music
  </label>

  <label>
```

```
      <input type="checkbox" name="interests" value="sports" />
      Sports
    </label>
  </fieldset>
</form>
```

💡 Wrapping the `<input>` inside `<label>` removes the need for a `for` attribute.

---

type="radio"

- Allows users to select **only one** option from a group of choices.

▼ 📝 **Code Example**

```
<form>
  <fieldset>
    <legend>Favorite Programming Language</legend>

    <input type="radio" id="C++" name="fav_language" value="C++">
    <label for="C++">C++</label><br>

    <input type="radio" id="Python" name="fav_language" value="Python">
    <label for="Python">Python</label><br>

    <input type="radio" id="JavaScript" name="fav_language" value="JavaScri
    <label for="JavaScript">JavaScript</label><br>

    <input type="submit" value="Submit">
  </fieldset>
</form>
```

💡 `<fieldset>` groups related fields, and `<legend>` provides a title.

type="search"

- Allows users to enter search queries.
- ▼ 📝 **Code Example**

```html
<input type="search" id="search" name="search" placeholder="Search...">
```

type="url"

- Allows users to enter a URL.
- ▼ 📝 **Code Example**

```html
<input type="url" id="website" name="website" placeholder="https://xyz.co
```

type="file"

- Allows users to upload files.
- ▼ 📝 **Code Exampl**

```html
<input type="file" name="profile_pic">
```

type="month"

- Allows users to select a month & year.
- ▼ 📝 **Code Example**

```
<input type="month" id="dob" name="dob">
```

type="color"

- Lets users select a **color**.
- ▼ 📝 **Code Example**

```
<input type="color" id="favcolor">
```

Semantic HTML

Semantic HTML uses elements that **describe the meaning** of their content, rather than just their appearance. It enhances:

- **Accessibility** (helps screen readers & assistive tech).
- **SEO** (improves search engine rankings).
- **Maintainability** (makes code easier to read & manage).

Common Semantic Elements

Element	Purpose
`<header>`	Defines introductory content, such as a logo or navigation links.
`<nav>`	Contains navigation links for the website.
`<main>`	Wraps the primary content of the page (**only one per page**).
`<section>`	Groups related content with a heading.
`<article>`	Represents independent, self-contained content (e.g., blog post, news article).
`<aside>`	Contains side content, such as a sidebar or ads.
`<footer>`	Defines footer content, like copyright info, links, or contact details.

Additional Topics

ARIA (Accessible Rich Internet Applications)

- ARIA is a set of HTML attributes that enhance web accessibility for users with disabilities, especially those using screen readers.

- ▼ 📝 **Code Example**

  ```
  <button aria-label="Close menu">X</button>
  ```

Meta Tags

- Meta tags provide **metadata** about a webpage (SEO, character set, responsiveness).

Purpose	Example	Description
Character Set	`<meta charset="UTF-8">`	Defines text encoding (UTF-8 supports most characters).
Viewport (Responsive Design)	`<meta name="viewport" content="width=device-width, initial-scale=1.0">`	Ensures proper scaling on mobile devices.
SEO - Description	`<meta name="description" content="chaicode">`	Helps search engines understand the page content.
Author	`<meta name="author" content="Aman Gupta">`	Defines the author of the webpage.
Refresh/Redirect	`<meta http-equiv="refresh" content="5; url=newpage.html">`	Redirects the page after 5 seconds.

`data-*` Attributes

- **Custom attributes** for storing extra information, which can be accessed dynamically via **JavaScript**.

- ▼ 📝 **Code Example**

```html
<button data-user-id="123">Click Me</button>
```