HTML

Hyper Text Markup Language

- ➔ HTML's primary function is to define the structure of a document using a tagging system; it allows web browsers to present web pages using a standardised language.
- ➔ Originally used to create content and apply style information to web pages: defines which parts of a document are headings, paragraphs, tables etc; and can be used to define how each section is presented.
- ➔ The latest iteration is HTML5, which introduced tags for multimedia content to be displayed natively in HTML.

Properties of HTML:

HTML is made up of a series of pre-defined "tags", which are indicated by angular brackets.

- Example: `<p>This text is tagged as a paragraph</p>`

Tags need to be opened and closed.

- An opening tag looks like this: `<p>`
- A closing tag looks like this: `</p>`

Tags must be "nested" – i.e., opened and closed in the correct order.
In practice, this means that when you open a tag which nests inside another tag, it must be closed before the surrounding tag is closed.

- Example: `<p>The nested tag <em>must be closed</em> before the surrounding tag</p>`

List of Few HTML tags and attributes

Tag	Purpose / Description	Example
`<!DOCTYPE html>`	Declares document type / HTML version	`<!DOCTYPE html>`
`<html>` … `</html>`	Root of HTML document	`<html lang="en">…</html>`
`<head>` … `</head>`	Contains metadata, title, linked resources	`<head><title>My Page</title></head>`
`<title>` … `</title>`	Title shown in browser tab	`<title>Home Page</title>`
`<meta>`	Metadata (charset, description, viewport)	`<meta charset="UTF-8">`

`<link>`	Links external resources (CSS, icons)	`<link rel="stylesheet" href="styles.css">`
`<style>` … `</style>`	Internal CSS styles	`<style>body { background: #fafafa; }</style>`
`<script>` … `</script>`	JavaScript code or linking external scripts	`<script src="script.js"></script>`
`<body>` … `</body>`	Visible content of web page	`<body><h1>Welcome</h1><p>Hello!</p></body>`
`<header>` … `</header>`	Introductory content or navigation for a section/page	`<header><nav>…</nav></header>`
`<footer>` … `</footer>`	Footer content (copyright, links)	`<footer>© 2025 My Site</footer>`
`<nav>` … `</nav>`	Navigation links of the site	`<nav><ul><li><a href="/">Home</a></li></ul></nav>`
`<section>` … `</section>`	A thematic grouping of content	`<section><h2>About us</h2><p>…</p></section>`
`<article>` … `</article>`	Independent, self-contained content (blog post, comment)	`<article><h3>Post Title</h3><p>…</p></article>`
`<div>` … `</div>`	Generic container, grouping other elements	`<div class="container"><p>…</p></div>`
`<h1>` … `<h6>`	Headings, h1 biggest, h6 smallest	`<h1>Main Title</h1>`
`<p>` … `</p>`	Paragraph	`<p>This is a paragraph.</p>`
`<a>` … `</a>`	Hyperlinks	`<a href="https://example.com">Visit site</a>`
`<img>`	Embeds an image	`<img src="image.jpg" alt="A scenery">`

`<ul>` … `</ul>`, `<ol>` … `</ol>`, `<li>`	Unordered / ordered lists and list items	`<ul><li>Item A</li><li>Item B</li></ul>`
`<table>` … `</table>`	Table layout	`<table><tr><th>Name</th><th>Age</th></tr><tr><td>Amy</td><td>30</td></tr></table>`
`<tbody>`, `<thead>`, `<tfoot>`, `<tr>`, `<td>`, `<th>`	Parts of a table	see table example above
`<form>` … `</form>`	Form for user input	`<form action="/submit" method="post"><input type="text" name="name"><button type="submit">Send</button></form>`
`<input>`	Input field (text, checkbox, etc)	`<input type="text" name="username" placeholder="Enter username">`
`<textarea>` … `</textarea>`	Multi-line text input	`<textarea name="message"></textarea>`
`<button>` … `</button>`	Button	`<button type="button">Click Me</button>`
`<select>` … `</select>` + `<option>`	Drop-down list	`<select name="options"><option value="1">One</option></select>`
`<label>`	Label for form controls	`<label for="email">Email:</label><input id="email" type="email">`
`<em>` … `</em>`	Emphasized text (usually italics)	`<p>This is <em>very</em> important.</p>`
`<strong>` … `</strong>`	Strong importance (usually bold)	`<p>This is <strong>urgent</strong>!</p>`

Attributes:

- Attributes modify the behavior or appearance of an element.
- They are written in the opening tag.
- Most attributes come in name="value" form, but some are boolean (like checked, disabled).

Attribute	Which tags / purpose	Example
`id`	Many tags; unique identifier for CSS/JS	`<div id="main-content">…</div>`
`class`	Many tags; used for CSS styling / JS targeting	`<p class="intro">Hello!</p>`
`style`	Inline styles, many tags	`<span style="color: blue;">Blue text</span>`
`title`	Many tags; tooltip text on hover	`<a title="Go to homepage" href="/">Home</a>`
`lang`	`<html>` (or any); language of content	`<html lang="en">…</html>`
`href`	`<a>` (link targets)	`<a href="https://example.com">Example</a>`
`src`	`<img>`, `<script>`, `<iframe>`, `<audio>`, `<video>` etc.	`<img src="pic.jpg" alt="Pic">`
`alt`	`<img>` (alternate text)	`<img src="photo.png" alt="Photo of tree">`

`width`, `height`	`<img>`, `<canvas>`, `<video>` etc.	`<img src="pic.jpg" width="200" height="150">`
`type`	`<input>`, `<button>`, `<script>` etc.	`<input type="email" name="email">`
`placeholder`	`<input>`, `<textarea>`	`<input type="text" placeholder="Enter name">`
`name`	Form controls (input, select, etc)	`<input name="username">`
`method`	`<form>` tag (how to send data)	`<form method="post" action="/send">…</form>`

`rel`	`<link>`, `<a>` etc (relationship)	`<link rel="stylesheet" href="styles.css">`
`media`	`<link>` etc. media queries	`<link rel="stylesheet" href="print.css" media="print">`
`controls`	`<video>`, `<audio>` (show play/pause etc)	`<audio controls src="song.mp3"></audio>`

`disabled`	Form controls, buttons etc.	`<button disabled>Can't click me</button>`
`readonly`	Inputs etc.	`<input type="text" readonly>`
`required`	Form fields	`<input type="text" required>`
`maxlength`	`<input>`, `<textarea>`	`<input type="text" maxlength="20">`
`pattern`	`<input>` (for regex validation)	`<input type="text" pattern="[A-Za-z]{3,}">`

`target`	`<a>` etc. (where linked document opens)	`<a href="page.html" target="_blank">Open new tab</a>`

CSS Properties

What is CSS?

- CSS = Cascading Style Sheets.
- Defines how HTML elements look (color, fonts, layout, spacing, etc.).
- Separates content (HTML) from design (CSS).

Why do we need CSS?

- Separation of content & design → cleaner code.
- Consistency → one CSS file can style multiple pages.
- Better design → colors, fonts, layouts, animations.
- Responsive design → adapts to mobile, tablet, desktop.
- Accessibility & user experience → improves readability.
- Efficiency → faster loading & easier maintenance.

Different Types of CSS:

1. Inline CSS - by using the style attribute inside HTML elements
2. Internal CSS - by using a <style> element in the <head> section
3. External CSS - by using a <link> element to link to an external CSS file

CSS Selectors:

1. Element Selector: The element selector selects HTML elements based on the element(tag) name.
2. Id Selector: The id selector uses the id attribute of an HTML element to select a specific element. The id of an element is unique within a page, so the id selector is used to select one unique element! To select an element with a specific id, write a hash (#) character, followed by the id of the element.
3. Class Selector: The class selector selects HTML elements with a specific class attribute. To select elements with a specific class, write a period (.) character, followed by the class name.
4. Universal Selector: The universal selector (*) selects all HTML elements on the page.
5. Grouping selector: The grouping selector selects all the HTML elements with the same style definitions.

```
h1, h2, p {
  text-align: center;
  color: red;
}
```

1. Text and Font

Property	Use	Example	Why
`color`	Sets text color	`p { color: red; }`	Makes text distinct
`font-family`	Defines font	`body { font-family: Arial, sans-serif; }`	Controls typography
`font-size`	Sets text size	`h1 { font-size: 32px; }`	Defines hierarchy
`font-weight`	Boldness	`p { font-weight: bold; }`	Emphasis
`font-style`	Italic/normal	`em { font-style: italic; }`	Adds emphasis

`text-align`	Aligns text	`h1 { text-align: center; }`	Layout control
`text-decoration`	Underline/none	`a { text-decoration: none; }`	Controls links
`line-height`	Line spacing	`p { line-height: 1.5; }`	Improves readability

2. Box Model (Spacing and Sizing)

`width` / `height`	Element size	`div { width: 300px; }`	Layout structure
`margin`	Space outside	`p { margin: 20px; }`	Separation
`padding`	Space inside	`button { padding: 10px; }`	Readability, click area
`border`	Outline style	`div { border: 1px solid black; }`	Visual separation
`box-sizing`	Size calc method	`* { box-sizing: border-box; }`	Easier layouts

3. Background

`background-color`	Background color	`body { background-color: lightblue; }`	Page aesthetics
`background-image`	Background image	`div { background-image: url("bg.jpg"); }`	Branding
`background-repeat`	Repeat control	`div { background-repeat: no-repeat; }`	Avoids tiling
`background-size`	Scale image	`div { background-size: cover; }`	Fits layout
`background-position`	Image position	`div { background-position: center; }`	Alignment

4. Positioning and Display

`display`	Display mode	`span { display: inline; }`	Controls flow
`position`	Positioning	`div { position: absolute; top: 50px; }`	Flexible placement
`top/bottom/left/right`	Position offsets	`div { top: 0; left: 0; }`	Exact placement
`z-index`	Stacking order	`img { z-index: 10; }`	Overlapping control
`float`	Float element	`img { float: left; }`	Text wrapping
`clear`	Clear float	`p { clear: both; }`	Prevent overlap

5. Flexbox and Grid

`display: flex`	Flexbox layout	`div { display: flex; }`	Responsive layouts
`justify-content`	Align horizontally	`div { justify-content: space-between; }`	Spacing
`align-items`	Align vertically	`div { align-items: center; }`	Vertical centering
`flex-direction`	Row/column	`div { flex-direction: column; }`	Orientation
`display: grid`	Grid layout	`div { display: grid; grid-template-columns: 1fr 1fr; }`	2D layouts
`grid-gap`	Gap between items	`div { grid-gap: 10px; }`	Spacing

6. Color and Effects

`opacity`	Transparency	`img { opacity: 0.5; }`	Visual effect
`box-shadow`	Element shadow	`div { box-shadow: 2px 2px 5px gray; }`	Depth
`text-shadow`	Text shadow	`h1 { text-shadow: 1px 1px 2px black; }`	Highlight
`cursor`	Mouse cursor style	`button { cursor: pointer; }`	Interactivity
`overflow`	Overflow control	`div { overflow: auto; }`	Scroll/hidden