HTML

HTML stands for HyperText Markup Language.

- HyperText = links between pages.
- Markup Language = a way to mark or label parts of a page (like headings, paragraphs, images).

Think of HTML as the skeleton of a webpage. It tells the browser what to show — like "this is a heading", "this is a paragraph", "this is an image". Use tags (like <h1>, <p>, <img>) to mark different parts.

Note: HTML = "How To Make Layout". It helps lay out the structure of a webpage.

HTML Document

An HTML document is just a file (usually ending with .html) that contains HTML code. It is the full page that the browser reads to show a website.

It’s like the recipe that tells the browser how to build and display a webpage.

Basic structure of an HTML document

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>Hello, World!</h1>
    <p>Welcome to my website!</p>
  </body>
</html>
```

- `<!DOCTYPE html>` Tells the browser: "Hey, this is an HTML5 document!". Without this line, the browser might get confused and display the page wrongly. It's like telling the browser what language you're speaking. The <!DOCTYPE> declaration is not case sensitive.
- `<html>` : Start of the webpage. Everything you want to show or set up must be inside <html> and </html>. It's like a container for the whole page.
- `<head>` : Contains meta information about the HTML page. it include page title, link to css, link to icons
- `<title>` : Sets the name that appears on the browser's tab. If you write `<title>`My Website`</title>`, then the browser tab will show "My Website".
- `<body>` : Visible part of the page. Whatever you write inside <body> (headings, text, images, videos, buttons) — people can see it when they visit the site.

HTML Element

An HTML element is defined by a start tag, some content, and an end tag.

Never skip the end tag, some elemets display correctly even if we forget. However, never rely on this Unexpected results and errors may occur if you forget the end tag.

HTML tags are not case sensitive, `<P> means the same as <p>`. Recommends lowercase in HTML.

```
<p>Hello, world!</p>
```

Think of an element like a sandwich. The top bread = start tag (`<p>`), The filling = content ("Hello, world!") and The bottom bread = end tag (`</p>`).

Types of HTML Elelments

There are two types of html elements : Container Elements and Empty Elements.

1. Container Elements : Have both a start tag and an end tag. Ex, `<p>`Paragraph text`</p>`
2. Empty Elements : Only have a start tag (no content, no end tag). Ex, `<img src="image.jpg" alt="Picture">` `<br>` (line break), `<hr>` (horizontal line)

Nested HTML Elements

Nested means one element is placed inside another element. In HTML, when you put one element inside another, it's called nesting.

```
<p>This is a <strong>very important</strong> message.</p>

<p> : parent element (paragraph)
<strong> : child element (makes the text bold)
```

```
<div>
  <h1>My Blog</h1>
  <p>Welcome to my blog. Enjoy <em>reading</em> the posts!</p>
</div>
```

Rules of Nesting :

- Always close inner elements first.
- Keep it clean and properly indented to make it easy to read.

HTML Attributes

Attributes are extra information that you add to an HTML element. They help control or modify how an element behaves or looks.

```
<tagname attributeName="value">Content</tagname>
```

Attributes are always inside the start tag. Attributes are name-value pairs: name="value". You can have multiple attributes in one tag.

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

href attribute

Sets where a link should go (the URL). `<a href="https://google.com">Visit Google</a>`. When you click "Visit Google", it opens Google.

src attribute

Tells which image file to display. `<img src="cat.jpg" alt="Cute Cat">`. Show cat picture.

width and height

Set the size of an image. `<img src="cat.jpg" width="300" height="200" alt="Cute Cat">`. Image will be 300 pixels wide and 200 pixels tall.

alt attribute

Shows alternative text if the image can't load. `<img src="missing.jpg" alt="Image not found">`. If the image is missing, "Image not found" is shown.

style attribute

Adds CSS styles like color, font size, background, etc. `<p style="color:blue; font-size:20px;">Hello World!</p>`.The text "Hello World!" will be blue and bigger.

lang attribute

Declares the language of the page (for browsers and search engines). `<html lang="en">`. Page is written in english.

title attribute

Shows extra information when you hover the mouse over the element.`<p title="This is a paragraph.">Hover over me!</p>`. When you hover your mouse, a tooltip "This is a paragraph." appears.

HTML Headings

Headings are used to give titles to sections of a web page.

Types of Headings There are 6 levels of headings in HTML:

```
<h1> → Biggest and most important heading
<h2> → Second most important
<h3> → Third most important
<h4> → Fourth
<h5> → Fifth
<h6> → Smallest and least important heading
```

Imagine a Newspaper :

- Main Title =

  # (e.g., "Today's Top News")

- Section Title =

  ("Sports", "Weather", "Politics")

- Smaller Story Title =

  ,

  , etc.

HTML Paragraphs

A paragraph in HTML is used to write normal text — like sentences, descriptions, articles, etc. Use the

tag to create a paragraph.

A paragraph always starts on a new line, and browsers automatically add some white space (a margin) before and after a paragraph.

```
<-- single paragraph -->

<p>This is my first paragraph in HTML!</p>
```

```
<!-- multiple paragraph -->

<p>HTML is very easy to learn.</p>
<p>It is used to create web pages.</p>
<p>Each paragraph is a separate block of text!</p>
```

`<hr>` : Horizontal Line

`<hr>` stands for Horizontal Rule. It draws a horizontal line across the web page. It is used to separate sections of content. It does not need a closing tag. (

---

alone is enough).

```
<!-- line will appear between the two paragraphs -->

<h1>Welcome to My Website</h1>
<p>This is the first section.</p>

<hr>

<p>This is the second section, separated by a line.</p>
```

`<br>` : Line Break

`<br>` stands for Break (line break). It moves the next text to a new line without starting a new paragraph. It also does not need a closing tag. (
alone!)

```
<p>Hello <br>How are you Joshi? <br> Have a nice day! </p>
```

`<pre>` : Preformatted Text

It is used to display text exactly as you type it. This means:Spaces and line breaks are preserved.The text is shown in a monospace font (like code). It's typically used for displaying things like code snippets or ASCII art.

```
<pre>
This is some preformatted text.
  It keeps all spaces and line breaks.
  Useful for displaying code or ASCII art.
</pre>
```

HTML Styles

HTML Styles allow you to change the look and feel of your web page, things like color, font size, background, margins, and more.

Styles are added in three main ways:

- Inline styles
- Internal styles
- External styles

1. Inline Styles

You can add styles directly to an element using the style attribute. It's called inline because the style is written inside the HTML tag itself.

```
<!-- syntax -->
<tag style="property:value;">

<!-- example -->
<p style="color: red; font-size: 20px;">This is a red paragraph with big text!</p>
```

2. Internal Styles

You can add styles in the head section of your HTML document using the `<style>` tag. These styles will apply to the whole page.

```
<!-- syntax -->
<head>
  <style>
    /* CSS rules go here */
  </style>
</head>


<!-- example -->
<head>
  <style>
    p {
      color: green;
      font-size: 18px;
    }
  </style>
</head>

<body>
  <p>This paragraph is green with 18px font size.</p>
</body>
```

3. External Styles

You can use a separate CSS file to define your styles. This way, you keep HTML and CSS separate and more organized.

```
<!-- syntax -->
<head>
  <link rel="stylesheet" href="styles.css">
</head>
```

```
<!-- styles.css -->
p{
  color: blue;
  font-size: 16px;
}
```

Property	What It Does	Example
`color`	Changes the text color	`color: red;`
`font-size`	Changes the font size	`font-size: 18px;`
`background-color`	Changes the background color of an element	`background-color: yellow;`
`font-family`	Changes the font style	`font-family: Arial, sans-serif;`
`margin`	Adds space outside the element	`margin: 20px;`
`padding`	Adds space inside the element	`padding: 10px;`
`border`	Adds a border around an element	`border: 1px solid black;`

HTML Text Formatting

In HTML, text formatting refers to how text appears on the web page. You can modify its style, structure, and layout using various HTML tags and attributes.

1. `<b>` : Bold Text

Makes the text bold. `<b>This text is bold.</b>`

2. `<strong>` : Strong Emphasis (Bold)

Indicates strong emphasis, typically shown as bold in browsers. `<strong>` is semantically meaningful and indicates importance, while `<b>` just makes text bold without any meaning. `<strong>This is strongly emphasized text (bold).</strong>`

3. `<i>` : Italic Text

Makes the text italic. `<i>This text is italic.</i>`

4. `<em>` : Emphasized Text (Italic)

Indicates emphasis, usually displayed in italic in browsers.`<em>` conveys meaning (emphasis), while `<i>` just styles the text. `<em>This text is emphasized (italic).</em>`

5. `<u>` : Underlined Text

Underlines the text. `<u>This text is underlined.</u>`

6. `<mark>` : Highlighted Text

Highlights the text by changing the background color. `<mark>This text is highlighted.</mark>`.

7. `<small>` : Smaller Text

Makes the text smaller than the surrounding text. `<small>This text is smaller.</small>`

8. `<sub>` : Subscript Text

Displays text as subscript (below the baseline). `H<sub>2</sub>O`. This will display H_2O where the "2" is subscripted.

9. `<sup>` : Superscript Text

Displays text as superscript (above the baseline). `E = mc<sup>2</sup>`. This will display $E = mc^2$, with the "2" as a superscript.

10. `<del>` : Deleted Text

Indicates deleted text (typically shown with a strikethrough). `<del>This text is deleted.</del>`.

11. `<ins>` — Inserted Text

Indicates inserted text (typically shown with an underline). `<ins>This text is inserted.</ins>`.

Tag	What It Does	Example
`<b>`	Makes the text bold	`<b>Bold text</b>`
`<strong>`	Strong emphasis (typically bold)	`<strong>Important text</strong>`
`<i>`	Italicizes the text	`<i>Italic text</i>`
`<em>`	Emphasizes text (typically italic)	`<em>Emphasized text</em>`
`<u>`	Underlines the text	`<u>Underlined text</u>`
`<s>`	Adds a strikethrough to the text	`<s>Strikethrough text</s>`
`<mark>`	Highlights text	`<mark>Highlighted text</mark>`
`<small>`	Makes text smaller	`<small>Small text</small>`
`<sub>`	Makes text subscript	`H<sub>2</sub>O`
`<sup>`	Makes text superscript	`E = mc<sup>2</sup>`
`<del>`	Indicates deleted text	`<del>Deleted text</del>`
`<ins>`	Indicates inserted text	`<ins>Inserted text</ins>`
`<code>`	Represents inline code	`<code>let x = 5;</code>`
`<pre>`	Displays preformatted text	`<pre>Preformatted text</pre>`
`<blockquote>`	Defines a block quote	`<blockquote>Quoted text</blockquote>`
`<q>`	Defines a short inline quote	`<q>Short quote</q>`

Tag	What It Does	Example
`<span>`	Generic inline container for styling	`<span>Styled text</span>`
` `	Adds a line break	` `

HTML Comments

HTML comments are used to add explanatory notes or comments within the code, which are not displayed on the web page. They are helpful for documenting the code, explaining sections of HTML, or leaving reminders for future development.

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

Example for understanding comment

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Comments Example</title>
</head>
<body>

    <!-- This is the main heading -->
    <h1>Welcome to My Website</h1>

    <!-- This section contains the navigation menu -->
    <nav>
        <ul>
            <li><a href="#">Home</a></li>
            <li><a href="#">About</a></li>
            <li><a href="#">Services</a></li>
            <li><a href="#">Contact</a></li>
        </ul>
    </nav>

    <!-- This paragraph contains some basic text -->
    <p>Here is a simple paragraph of text.</p>

    <!-- Remember to update the footer section later -->
    <footer>
        <p>Footer information goes here.</p>
    </footer>

</body>
</html>
```

Hide Content

HTML comments can be used to temporarily hide content. While it is not a method for dynamically hiding content during runtime like CSS or JavaScript, it can be useful when you want to disable certain parts of your code for testing or debugging without removing them entirely.

```
<!-- This content is hidden and will not be displayed in the browser -->
<p>This paragraph will be hidden because it is inside a comment.</p>
```

Hide Inline Content

HTML comments can be used to hide inline content anywhere within your HTML code, including in the middle of elements or between text. This is useful when you want to temporarily disable part of your content, such as a specific paragraph, image, or text, without deleting it.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Hide Inline Content Example</title>
</head>
<body>

    <h1>Welcome to My Website</h1>

    <p>This is a visible paragraph of text.</p>

    <!-- <p>This paragraph is hidden inline and won't be displayed.</p> -->

    <p>This is another visible paragraph.</p>

    <p><strong>This is bold text.</strong> <!-- This part of text is hidden -->
More text here.</p>

</body>
</html>
```

HTML Links

HTML links are created using the (anchor) element. The `<a>` element is used to define hyperlinks, which allow users to click and navigate to another webpage, section of the same page, or external resource.

By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

```
<!-- syntax -->
<a href="URL">Link Text</a>

<!-- example : In this example, the user will be directed to
"https://www.example.com" when they click on the text "Visit Example Website." -->

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

Types of Links

1. External Links

Links that point to a different website or domain.

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

When clicked, this will take the user to the Google homepage.

2.Internal Links

Links that navigate within the same website or domain.

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

Clicking this link will navigate the user to the about.html page within the same website.

3. Anchor Links (Internal Navigation)

Links that point to a specific part of the same page, often used for navigation within long pages.

```
<a href="#section2">Go to Section 2</a>
```

This would navigate the user to an element on the page with the id="section2", like this:

```
<div id="section2">This is Section 2</div>
```

4. Email Links

Links that allow the user to open their default email client and send an email.

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

Clicking this link will open the user's email client with the recipient's email pre-filled.

5. Phone Links

Links that initiate a phone call on mobile devices.

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

On a mobile device, this would trigger the phone dialer with the given phone number.

Example of Multiple Links

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>HTML Links Example</title>
</head>
<body>
    <h1>Welcome to My Website</h1>
    <p>Click the links below to navigate:</p>

    <a href="https://www.google.com">Go to Google</a><br>
    <a href="/about.html">About Us</a><br>
    <a href="#contact">Go to Contact Section</a><br>
    <a href="mailto:someone@example.com">Send an Email</a><br>
    <a href="tel:+1234567890">Call Us</a>

    <h2 id="contact">Contact Section</h2>
    <p>Here is the contact information.</p>
</body>
</html>
```

HTML Images

In HTML, images are used to display pictures on a webpage. This is done using the `<img>` tag.

The `<img>` tag is self-closing (it doesn't have a closing `</img>`). It must include the following key attributes: src, alt, width and height.

```
<img src="logo.png" alt="Company Logo" width="200" height="100">
```

Attribute	Description
`src`	(source) The URL or path to the image file

Attribute	Description
`alt`	(alternative text) Describes the image if it cannot be displayed or for screen readers
`width`	Sets the width of the image (in pixels or %)
`height`	Sets the height of the image (in pixels or %)

The width, height, and style attributes are all valid in HTML. However, we suggest using the style attribute. It prevents styles sheets from changing the size of images.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      img {
        width: 100%;
      }
    </style>
  </head>
  <body>

  <img src="html5.gif" alt="HTML5 Icon" width="128" height="128">

  <img src="html5.gif" alt="HTML5 Icon" style="width:128px;height:128px;">

  </body>
</html>
```

Image in a Subfolder

To display images located in another folder in HTML, you just need to provide the correct relative or absolute path in the src attribute of the tag.

```
1. Assume your folder structure
project/
├── index.html
└── images/
    └── logo.png
<img src="images/logo.png" alt="Company Logo" width="200">

2. If the image is in a folder above your HTML file
project/
├── images/
│   └── logo.png
└── pages/
    └── about.html
<img src="../images/logo.png" alt="Company Logo">

3. Image in a Different Folder
```

```
project/
├── assets/
│   └── imgs/
│       └── profile.jpg
└── public/
    └── index.html
<img src="../assets/imgs/profile.jpg" alt="Profile Picture">
```

HTML Favicon

A favicon (short for "favorite icon") is a small icon associated with a particular website, typically displayed in: Browser tabs, Bookmarks, Address bar (in some browsers) and Shortcut links on mobile home screens.

Purpose of a Favicon

- Brand recognition: Acts like a website's logo.
- Usability: Helps users visually identify a site among many open tabs or bookmarks.
- Professionalism: Makes your site look complete and polished.

You define a favicon using the `<link>` tag inside the `<head>` section of your HTML document.

```
<!-- syntax -->
<link rel="icon" href="/path-to-your/favicon.ico" type="image/x-icon">
```

Modern websites often include multiple formats and sizes for broad compatibility:

```
<!-- Standard favicon -->
<link rel="icon" href="/favicon.ico" type="image/x-icon">

<!-- PNG format -->
<link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">
<link rel="icon" type="image/png" sizes="16x16" href="/favicon-16x16.png">

<!-- Apple Touch Icon (iOS bookmark) -->
<link rel="apple-touch-icon" sizes="180x180" href="/apple-touch-icon.png">

<!-- Web Manifest (for PWA support) -->
<link rel="manifest" href="/site.webmanifest">
```

Example: Full HTML with Favicon

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
```

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

  <!-- Favicons -->
  <link rel="icon" href="/favicon.ico" type="image/x-icon">
  <link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">
  <link rel="apple-touch-icon" sizes="180x180" href="/apple-touch-icon.png">
</head>
<body>
  <h1>Welcome to My Website</h1>
</body>
</html>
```

HTML Page Title

The HTML Page Title is the text that appears in the browser tab, search engine results, and when bookmarking a page. It's defined using the `<title>` tag inside the `<head>` section of an HTML document.

```
<!-- syntax -->
<title>Your Page Title</title>
```

Example : Show My Portfolio in browser tab

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>My Portfolio</title>
</head>
<body>
  <h1>Welcome to My Portfolio</h1>
</body>
</html>
```

HTML table

An HTML table is used to display data in rows and columns, similar to a spreadsheet.

```
<table>
  <caption>Student Grades</caption>
  <tr>
    <th>Name</th>
    <th>Subject</th>
    <th>Grade</th>
```

```
    </tr>
    <tr>
      <td>Joshi</td>
      <td>CS</td>
      <td>A</td>
    </tr>
    <tr>
      <td>Sourabha</td>
      <td>Science</td>
      <td>B</td>
    </tr>
  </table>
```

dHTML Table Tags and Their Purpose

Tag	Purpose
`<table>`	Defines the table
`<tr>`	Table row
`<th>`	Table header cell (bold and centered)
`<td>`	Table data cell (regular cell)
`<caption>`	(Optional) Adds a title above the table

HTML Table Borders

In HTML, borders help visually separate cells and improve the readability of tables. You can add borders to tables using either HTML attributes (deprecated) or CSS (recommended).

Method 1: Using HTML border Attribute. It is old method

```
<table border="1">
  <tr>
    <th>Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Alice</td>
    <td>25</td>
  </tr>
</table>
```

This adds a basic black border around table cells. border="1" sets the border width to 1 pixel. You can use values like border="2" for thicker borders.

Method 2: Using CSS (Recommended)

```
<style>
  table {
    border-collapse: collapse;
    width: 50%;
  }

  th, td {
    border: 1px solid #333;
    padding: 8px;
  }
</style>

<table>
  <tr>
    <th>Product</th>
    <th>Price</th>
  </tr>
  <tr>
    <td>Pen</td>
    <td>$1.00</td>
  </tr>
</table>
```

1. How to Add a Border

You can add borders to a table and its cells using the border property in CSS.

```
table, th, td {
  border: 1px solid black;
}

<!--
  table applies a border around the entire table.
  th, td apply borders around each header and data cell.
  1px solid black: sets the border thickness, style, and color.
 -->
```

2. Collapsed Table Borders

By default, browsers show double borders where cells meet. To fix that, use border-collapse.

```
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}

<!-- border-collapse: collapse; removes space between borders, so adjacent cells
share a single border. -->
```

3. Style Table Borders (Invisible-Like Look)

You can make borders blend into the background for a softer look.

```
table, th, td {
  border: 1px solid white;        /* white border (same as background) */
  border-collapse: collapse;
}

th, td {
  background-color: #96D4D4;      /* pastel background */
}

<!-- This gives the illusion of cells floating without visible borders -->
```

4. Round Table Borders

We can create rounded corners using the border-radius property.

```
table, th, td {
  border: 1px solid black;
  border-radius: 10px;
}
<!-- border-radius adds curvature to corners. -->

th, td {
  border: 1px solid black;
  border-radius: 10px;
}
<!-- For individual cells only (not the outer table border), omit table from the
selector -->
```

5. Dotted and Styled Borders

he border-style property lets you define the appearance of the border : dotted dashed solid double groove ridge inset outset none hidden

```
th, td {
  border: 2px dotted black;
}
```

6. Border Color

Customize the color of the borders using border-color.

```
th, td {
  border: 1px solid;
  border-color: #96D4D4;
}
```

HTML Table Sizes

In HTML, you can control the size of a table, including its width, height, and the dimensions of its rows and columns, using either HTML attributes (deprecated) or CSS (recommended).

You can set the table size using the width and height CSS properties.

```
<style>
  table {
    width: 80%;         /* Table takes 80% of the page width */
    height: 200px;      /* Fixed table height */
  }

  th, td {
    height: 50px;       /* Each cell has a height of 50px */
    width: 100px;       /* Each cell has a width of 100px */
  }
</style>

<table border="1">
  <tr>
    <th>Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Alice</td>
    <td>25</td>
  </tr>
</table>
```

HTML Table Headers

In HTML, a table header is used to label the rows or columns in a table. It helps people understand what each part of the table means.

`<th>` : Table Header Cell

The `<th>` tag is used to define header cells in an HTML table. These are usually bold and center-aligned by default.

```
<table border="1">
  <tr>
    <th>Name</th>
    <th>Age</th>
```

```
    </tr>
    <tr>
      <td>Alice</td>
      <td>25</td>
    </tr>
  </table>
```

HTML Table Colspan & Rowspan

colspan (short for "column span") is an HTML attribute used in a `<td>` or `<th>` tag to make a cell stretch across multiple columns. Joins multiple columns into one cell (side by side).

Example : colspan – Merge Columns

```
<table border="1">
  <tr>
    <th colspan="2">User Info</th>
  </tr>
  <tr>
    <td>Name</td>
    <td>Alice</td>
  </tr>
</table>
```

rowspan (short for "row span") is an HTML attribute used to make a cell stretch across multiple rows. Joins multiple rows into one cell (top to bottom).

Example : rowspan – Merge Rows

```
<table border="1">
  <tr>
    <th rowspan="2">Name</th>
    <td>Alice</td>
  </tr>
  <tr>
    <td>Bob</td>
  </tr>
</table>
```

HTML Lists

HTML provides different types of lists to display grouped items. Lists are useful for organizing content clearly and cleanly.

Types of Lists in HTML Ordered List, Unordered List and Description List.

HTML Unordered Lists

An unordered list in HTML is a collection of list items that appear without any specific order. Each item is typically marked with a bullet point (•) by default.

- HTML Tag: `<ul>` (unordered list)
- List items inside: `<li>` (list item)

Choose List Item Marker :

You can customize bullet styles using the CSS list-style-type property: disc, circle, square and none.

You can also use custom images with list-style-image or even emojis via ::before pseudo-element in CSS.

```
<h3>Shopping List</h3>
<ul>
  <li>Milk</li>
  <li>Bread</li>
  <li>Eggs</li>
</ul>
```

HTML Ordered Lists

An ordered list in HTML is used to display a list of items in a specific sequence or order. Each item is numbered automatically by the browser.

- HTML Tag: `<ol>` (ordered list)
- List items inside: `<li>` (list item)

Choose Numbering Style

You can change the numbering style using the type attribute in `<ol>` or using CSS. : A, I, i, a, 1

```
<ol type="A">
  <li>First</li>
  <li>Second</li>
  <li>Third</li>
</ol>
```

HTML Description Lists

An HTML Description List is used to group terms and their associated descriptions. It consists of a series of terms (usually keywords) and their explanations or definitions.

- HTML Tag:

  (description list)

- Each term:
  (description term)
- Each description:

(description definition)

```
<h3>HTML Glossary</h3>
<dl>
  <dt>HTML</dt>
  <dd>HyperText Markup Language, used to structure content on the web.</dd>

  <dt>CSS</dt>
  <dd>Cascading Style Sheets, used to style the appearance of content in HTML.
</dd>

  <dt>JavaScript</dt>
  <dd>A programming language used to create interactive effects within web
browsers.</dd>
</dl>
```

HTML Block and Inline Elements

HTML elements are categorized into two main types based on their behavior and how they are displayed on a page: Block-level elements and Inline elements.

Block-level Elements

Block-level elements take up the entire width available to them, meaning they start on a new line and stack vertically one after the other. They typically form the structure or "blocks" of a page.

Ex : `<div>`, `<p>`, `<ul>`, `<table>`, `<header>`

Characteristics:

- Takes up the full width of its container (unless a specific width is set).
- Always starts on a new line and occupies as much width as possible.
- Can contain other block-level and inline elements.

Inline Elements

Inline elements do not start on a new line. They only take up as much width as necessary and sit inline with other content. They are typically used for styling smaller pieces of content within a block.

Ex : `<a>`, `<img>`, `<span>`, `<strong>`, `<b>`

Characteristics:

- Does not start on a new line; flows alongside other inline elements.
- Only takes up as much width as required by its content.
- Cannot contain block-level elements (but can contain other inline elements).

Block-level Elements vs Inline Elements

Feature	Block-level Elements	Inline Elements

Feature	Block-level Elements	Inline Elements
Display Behavior	Starts on a new line, takes full width of container	Does not start a new line, takes only necessary width
Can Contain	Can contain other block-level and inline elements	Can contain only other inline elements
Examples	`<div>`, `<p>`, `<ul>`, `<table>`, `<header>`	`<a>`, `<img>`, `<span>`, `<strong>`, `<b>`

and in HTML

Both

and are generic containers that are commonly used in HTML to group and style content. They don't provide any inherent meaning or semantic value by themselves, but they help organize content for styling, layout, and scripting purposes.

`<div>` Element

The `<div>` (short for "division") element is a block-level container used to group other block-level or inline elements. It helps in structuring the layout of a webpage and is used to apply styles or JavaScript to a group of elements.

The `<div>` element is often used as a container for other HTML elements. The `<div>` element has no required attributes, but style, class and id are common.

```
<div class="container">
  <h1>Welcome to My Website</h1>
  <p>This is a block-level paragraph inside a div.</p>
</div>
```

Example with style

```
<div style="background-color:black;color:white;padding:20px;">
  <h2>London</h2>
  <p>London is the capital city of England. It is the most populous city in the
United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>
</div>
```

`<span>` Element

The `<span>` element is an inline container used to group or style a small section of text or content within a line. It does not break the flow of the content and only takes up as much space as its contents require.

The `<span>` element is an inline container used to mark up a part of a text, or a part of a document. The `<span>` element has no required attributes, but style, class and id are common.

```
<p>The <span style="color: red;">quick</span> brown fox jumped over the lazy dog.
</p>
```

example with css style

```
<p>My mother has <span style="color:blue;font-weight:bold;">blue</span> eyes and
my father has <span style="color:darkolivegreen;font-weight:bold;">dark
green</span> eyes.</p>
```

`<div>` **vs** `<span>`

Feature	`<div>`	`<span>`
Type	Block-level element	Inline element
Display Behavior	Starts on a new line, takes full width of its container	Does not start a new line, takes only necessary width
Use Case	Grouping larger sections, applying layout styles	Styling small portions of text or content within a line
Can Contain	Can contain block-level and inline elements	Can only contain inline elements

HTML class Attribute

The class attribute in HTML is used to assign one or more class names to an HTML element. These class names are then used to apply CSS styles, group elements for styling purposes, and target elements with JavaScript for DOM manipulation.

The class attribute is a global attribute that can be applied to any HTML element. It allows you to assign a class name to that element, which can then be used to select it in your CSS or JavaScript.

```
<!-- syntax -->
<tag class="class-name">Content</tag>
```

How the class Attribute Works:

- CSS Styling: The primary use of the class attribute is to apply specific styles to an element or group of elements.
- JavaScript Manipulation: You can use the class attribute to select and manipulate elements with JavaScript (e.g., adding/removing classes dynamically).
- Reusability: By assigning a class name to multiple elements, you can apply the same styles to all of them at once, ensuring consistent design.

Example 1: Using class with CSS

You can use the class attribute to group elements and apply CSS styles to them.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Class Example</title>
  <style>
    .highlight {
      color: red;
      font-weight: bold;
    }
  </style>
</head>
<body>
  <p class="highlight">This is a highlighted paragraph.</p>
  <p class="highlight">This is another highlighted paragraph.</p>
</body>
</html>
```

Example 2: Multiple Classes

You can assign multiple class names to an element, allowing you to apply different sets of styles.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Multiple Classes Example</title>
  <style>
    .red-text {
      color: red;
    }

    .bold-text {
      font-weight: bold;
    }
  </style>
</head>
<body>
  <p class="red-text bold-text">This paragraph is red and bold.</p>
</body>
</html>
```

Example 3: Using class with JavaScript

You can manipulate elements with a specific class name using JavaScript. Here's an example of how to change the background color of all elements with a particular class.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Class and JavaScript Example</title>
  <style>
    .change-bg {
      padding: 10px;
      margin: 5px;
      background-color: lightgray;
    }
  </style>
</head>
<body>

  <div class="change-bg">Box 1</div>
  <div class="change-bg">Box 2</div>
  <div class="change-bg">Box 3</div>

  <script>
    // Select all elements with the class 'change-bg' and change their background
color
    const boxes = document.querySelectorAll('.change-bg');
    boxes.forEach(box => {
      box.style.backgroundColor = 'yellow'; // Change background to yellow
    });
  </script>
</body>
</html>
```

HTML id Attribute

The id attribute in HTML is used to assign a unique identifier to an HTML element. This ID can then be used to reference the element in CSS for styling, JavaScript for manipulation, and for creating hyperlinks to specific parts of a page.

The id attribute provides a unique identifier for an HTML element. Unlike the class attribute, which can be applied to multiple elements, the id attribute should only be used on one element per page, ensuring that each ID is unique.

```
<!-- syntax -->
<tag id="unique-id">Content</tag>
```

Key Points About the id Attribute:

- Uniqueness: An id must be unique within a page. No two elements should share the same id.
- CSS Styling: The id attribute is often used to target specific elements for styling.
- JavaScript Manipulation: JavaScript can be used to select, modify, or manipulate elements based on their id.
- Anchor Links: The id attribute can be used to create "anchor links" that allow navigation to specific sections of a page.

Example 1: Using id with CSS

You can use the id attribute to style a specific element uniquely.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>id Attribute Example</title>
  <style>
    #highlight {
      color: red;
      font-weight: bold;
    }
  </style>
</head>
<body>
  <p id="highlight">This is a paragraph with a unique ID.</p>
</body>
</html>
```

Example 2: Using id with JavaScript

You can use the id attribute to manipulate an element in JavaScript.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>id Attribute with JavaScript</title>
  <style>
    #colorBox {
      width: 100px;
      height: 100px;
      background-color: lightgray;
    }
  </style>
```

```
</head>
<body>
  <div id="colorBox"></div>
  <button onclick="changeColor()">Change Color</button>

  <script>
    function changeColor() {
      const box = document.getElementById('colorBox');
      box.style.backgroundColor = 'blue'; // Change background color
    }
  </script>
</body>
</html>
```

Example 3: Using id for Anchor Links

You can use the id attribute to create anchor links that allow users to navigate directly to a specific section of the page.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>id Attribute for Anchor Links</title>
</head>
<body>

  <h1>Welcome to My Website</h1>
  <p>Click the link to go to the section.</p>

  <a href="#section1">Go to Section 1</a>

  <div id="section1">
    <h2>This is Section 1</h2>
    <p>This is some content in Section 1.</p>
  </div>

</body>
</html>
```

Differences Between `id` and `class` Attributes

The `id` and `class` attributes are both used to assign identifiers to HTML elements, but they serve different purposes. Below is a comparison table outlining the differences between the two:

Feature	`id` Attribute	`class` Attribute
Uniqueness	Must be unique within a page	Can be used on multiple elements

Feature	`id` Attribute	`class` Attribute
CSS	Targets one specific element	Can target multiple elements at once
JavaScript	Selects an element by `id` using `getElementById()`	Selects elements by `class` using `getElementsByClassName()` or `querySelectorAll()`
Use Case	Identifying a specific element for unique styling or interaction	Grouping elements for shared styling or behavior

---

HTML Iframes

The `<iframe>` element in HTML is used to embed another HTML page (or other web content like videos, maps, etc.) within the current page. It stands for inline frame, and it's commonly used to display external content like YouTube videos, Google Maps, or other websites inside a webpage.

An `<iframe>` creates an embedded window (or frame) within the current HTML document. This window can display another webpage or media from an external source.

```
<iframe src="URL" width="width" height="height"></iframe>
```

Example 1: Embedding a YouTube Video

The YouTube video is embedded directly within the page using the `<iframe>` element.

The src attribute points to the embedded video URL, and allowfullscreen enables fullscreen mode for the video.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Embedding YouTube Video</title>
</head>
<body>
  <h2>Embedded YouTube Video</h2>
  <iframe width="560" height="315" src="https://www.youtube.com/embed/dQw4w9WgXcQ"
frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope;
picture-in-picture" allowfullscreen></iframe>
</body>
</html>
```

Example 2: Embedding an External Website

An external website (https://www.example.com) is embedded within the iframe.

The iframe has a defined width and height.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Embedding External Website</title>
</head>
<body>
  <h2>Embedding External Website</h2>
  <iframe src="https://www.example.com" width="600" height="400"></iframe>
</body>
</html>
```

Example 3: Using srcdoc to Embed HTML

The srcdoc attribute is used to directly embed HTML content inside the iframe without needing an external file.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Embedding HTML using srcdoc</title>
</head>
<body>
  <h2>Embedding HTML with srcdoc</h2>
  <iframe srcdoc="<p>This is embedded HTML content inside the iframe!</p>"
width="500" height="300"></iframe>
</body>
</html>
```

Pros and Cons of Using `<iframe>`:

Pros:

- Easy way to embed external content like videos, maps, and other websites.
- Prevents content on your page from directly interacting with the content within the iframe.

Cons:

- Can impact page load times, especially if embedding large or multiple external resources.
- Can present security risks if not used carefully, especially when embedding untrusted content.

HTML Layout Elements and Techniques

HTML provides several layout elements and techniques to help organize the structure and design of a webpage. These elements allow for the creation of flexible and responsive layouts that work across different screen sizes and devices. In modern web development, HTML layout elements and CSS techniques play an essential role in creating user-friendly and aesthetically pleasing websites.

HTML Layout Elements

1. `<div>` (Division Element)

The `<div>` element is a block-level element used to group content for styling purposes.

It is commonly used in combination with CSS to create layout structures, such as sections, columns, and containers.

```
<div class="container">
  <div class="row">
    <div class="col">Content 1</div>
    <div class="col">Content 2</div>
  </div>
</div>
```

2.

Represents the introductory content of a document, section, or page.

Often contains the navigation menu, logo, or introductory text.

```
<header>
  <h1>Website Title</h1>
  <nav>
    <ul>
      <li><a href="#">Home</a></li>
      <li><a href="#">About</a></li>
      <li><a href="#">Services</a></li>
    </ul>
  </nav>
</header>
```

3.

Represents the footer content of a document or section.

It usually includes copyright information, contact details, and links to privacy policies or terms of service.

```
<footer>
  <p>&copy; 2025 My Website. All rights reserved.</p>
  <a href="#">Privacy Policy</a> | <a href="#">Terms of Service</a>
</footer>
```

4.

Represents a distinct section of content, typically within a webpage, that can be identified by a heading.

Useful for organizing content into thematic groups.

```
<section>
  <h2>About Us</h2>
  <p>We are a company dedicated to providing high-quality services.</p>
</section>
```

5.

Represents an independent piece of content that can be distributed or reused, such as a blog post or news article.

```
<article>
  <h2>Latest Blog Post</h2>
  <p>This is an example blog post.</p>
</article>
```

6.

Represents content that is tangentially related to the main content, like a sidebar or callout box.

```
<aside>
  <h3>Related Links</h3>
  <ul>
    <li><a href="#">Link 1</a></li>
    <li><a href="#">Link 2</a></li>
  </ul>
</aside>
```

7.

Represents a navigation section, typically containing links to other pages or sections within the site.

```
<nav>
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">Contact</a></li>
  </ul>
</nav>
```

HTML Layout Techniques

1. CSS Grid Layout

The CSS Grid Layout provides a powerful and flexible way to create grid-based layouts.

It allows you to create complex grid structures with rows and columns.

```
<style>
  .grid-container {
    display: grid;
    grid-template-columns: auto auto auto;
    gap: 10px;
  }
  .grid-item {
    background-color: lightblue;
    padding: 20px;
    text-align: center;
  }
</style>
<div class="grid-container">
  <div class="grid-item">Item 1</div>
  <div class="grid-item">Item 2</div>
  <div class="grid-item">Item 3</div>
</div>
```

2. CSS Flexbox Layout

Flexbox is a one-dimensional layout model for arranging items in rows or columns.

It is highly flexible and responsive, allowing items to adjust automatically.

```
<style>
  .flex-container {
    display: flex;
    justify-content: space-between;
    align-items: center;
  }
  .flex-item {
    background-color: lightgreen;
    padding: 20px;
    margin: 10px;
  }
</style>
<div class="flex-container">
  <div class="flex-item">Item 1</div>
  <div class="flex-item">Item 2</div>
  <div class="flex-item">Item 3</div>
</div>
```

3. CSS Float Layout (Legacy)

Float was historically used to create multi-column layouts, but it is now considered less efficient than CSS Grid or Flexbox.

```
<style>
  .container {
    width: 100%;
  }
  .left-column {
    float: left;
    width: 70%;
    background-color: lightgray;
  }
  .right-column {
    float: right;
    width: 30%;
    background-color: lightblue;
  }
</style>
<div class="container">
  <div class="left-column">Main content here...</div>
  <div class="right-column">Sidebar content here...</div>
</div>
```

4. CSS Multi-Column Layout

Multi-column layout is used for creating layouts where content is split into multiple columns, like a newspaper layout.

```
<style>
  .multi-column {
    column-count: 3;
    column-gap: 20px;
  }
</style>
<div class="multi-column">
  <p>This content will be displayed in three columns.</p>
  <p>More content...</p>
</div>
```

5. CSS Positioning

The positioning property allows you to place elements exactly where you want them on the page (e.g., absolute, relative, fixed, sticky).

```
<style>
  .positioned-element {
```

```
      position: absolute;
      top: 50px;
      left: 100px;
      width: 200px;
      height: 100px;
      background-color: lightcoral;
    }
  </style>
  <div class="positioned-element">
    I am positioned absolutely!
  </div>
```

Responsive Layouts

1. Responsive Web Design (RWD)

Responsive Web Design ensures that a webpage looks good on all devices (desktops, tablets, mobile phones) by using flexible grid layouts, images, and media queries.

Media queries adjust the styles based on the viewport size.

```
<style>
  @media screen and (max-width: 600px) {
    .container {
      display: block;
    }
  }
</style>
```

HTML Semantic Elements

A semantic element clearly describes its meaning to both the browser and the developer.

These elements give structure to a web page, making it easier for developers, browsers, and assistive technologies (like screen readers) to understand the content.

For example, `<header>`, `<footer>`, `<article>`, and `<section>` indicate the type of content they contain.

Semantic Tags

This document provides a description of various semantic HTML5 tags and their purposes.

Tag	Description
`<article>`	Defines independent, self-contained content
`<aside>`	Defines content aside from the page content
`<details>`	Defines additional details that the user can view or hide

Tag	Description
`<figcaption>`	Defines a caption for a `<figure>` element
`<figure>`	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
`<footer>`	Defines a footer for a document or section
`<header>`	Specifies a header for a document or section
`<main>`	Specifies the main content of a document
`<mark>`	Defines marked/highlighted text
`<nav>`	Defines navigation links
`<section>`	Defines a section in a document
`<summary>`	Defines a visible heading for a `<details>` element
`<time>`	Defines a date/time