



HTML Cheat Sheet

(Cluster 1)

(YouTube Link: [Click Me](#))

➤ HTML: An Overview

Introduction:

HTML (HyperText Markup Language) is the standard language used to create web pages. It describes the structure of a web page by using markup. HTML elements are represented by tags, which define the content and layout of a web page.

Founder:

HTML was created by **Tim Berners-Lee** in **1991** while working at CERN (European Organization for Nuclear Research). He is also credited with inventing the World Wide Web.

Year of Introduction:

HTML was introduced in 1991, initially as a simple markup language with basic tags for text formatting.

Versions of HTML:

1. HTML 1.0 (1991): The first version of HTML, which was quite basic and limited.
2. HTML 2.0 (1995): Included forms and tables.
3. HTML 3.2 (1997): Supported complex tables, scripting languages, and applets.
4. HTML 4.01 (1999): Introduced CSS for styling, improved accessibility, and web standards.

5. XHTML (2000): A more strict version of HTML, following XML syntax rules.
6. HTML5 (2014): The current version, providing new features like semantic tags, multimedia support (audio, video), and local storage.

Why HTML is Used:

- 1. Structure Web Pages:** HTML is the backbone of web pages, organizing and formatting content.
- 2. Cross-Platform:** HTML works across all platforms, including desktops, mobiles, and tablets.
- 3. Easy to Learn and Use:** Its tag-based structure makes it beginner-friendly.
- 4. Support for Multimedia:** With HTML5, you can embed videos, audio, and other interactive content.
- 5. SEO-Friendly:** HTML helps search engines understand the content of a webpage, which is essential for Search Engine Optimization (SEO).

Basic Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>My First HTML Page</title>
</head>
<body>
  <h1>Hello, World!</h1>
  <p>This is a simple HTML document.</p>
</body>
</html>
```

Output on browser:



Hello, World!

This is a simple HTML document.

Explanation:

- **<!DOCTYPE html>** : Declares that the document is an HTML5 document.
- **<html>** : Root element of an HTML page.
- **<head>** : Contains metadata and the title of the document.
- **<body>** : Contains the content that is displayed in the browser, such as headings and paragraphs.

➤ HTML Documents

Definition:

An HTML document is a file that contains the code that defines the structure and content of a webpage. These documents are saved with a .html or .htm extension and can be opened in any web browser.

Structure of an HTML Document:

An HTML document typically follows a specific structure composed of different elements, which include:

1. Document Type Declaration (<!DOCTYPE html>):

- This tag tells the browser that the document follows HTML5 standards.
- It ensures proper rendering across different browsers.

2. HTML Element (<html>):

- This is the root element that contains all other HTML elements.
- It usually contains two main sections: <head> and <body>.

3. Head Section (<head>):

- Contains metadata about the document, like the title, character encoding, styles, and links to external files.
- Does not display content directly on the web page.

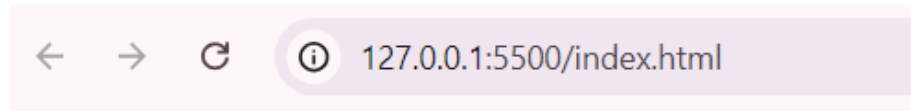
4. Body Section (<body>):

- Contains the visible content of the web page, such as text, images, links, forms, etc.

Basic Structure of an HTML Document:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>HTML Document Structure</title>
</head>
<body>
  <h1>Welcome to My Webpage</h1>
  <p>This is a basic HTML document structure.</p>
</body>
</html>
```

Output On Browser:



Welcome to My Webpage

This is a basic HTML document structure.

Explanation of Key Elements:

1. **<!DOCTYPE html>** : Informs the browser that this is an HTML5 document.
2. **<html lang="en">** : Root element, the `lang` attribute specifies the language of the document (in this case, English).
3. **<head>** : Contains metadata (data about the data), like `<title>` , `<meta>` , and links to stylesheets.
4. **<meta charset="UTF-8">** : Ensures the page uses UTF-8 encoding, which supports special characters.
5. **<meta name="viewport" content="width=device-width, initial-scale=1.0">** : Makes the page responsive on mobile devices.
6. **<title>** : Specifies the title of the webpage (appears in the browser tab).
7. **<body>** : Contains the actual content (headings, paragraphs, etc.) displayed on the webpage.

Key Points to Remember:

- The `head` section contains information that helps browsers and search engines understand the page.
- The `body` section contains the visual elements seen by users when the webpage is loaded.

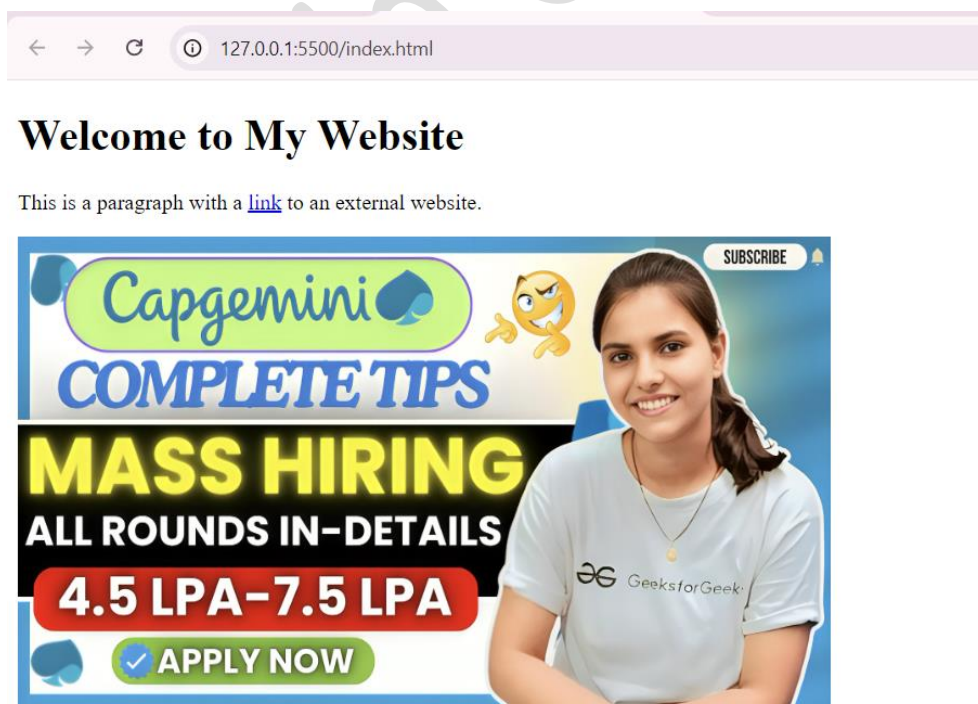
- Every HTML document begins with `<!DOCTYPE html>` and is wrapped inside `<html>` tags.

Example of an HTML Document with Links and Images:

html

```
<> index.html X
<> index.html > html > body
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Sample Web Page</title>
7  </head>
8  <body>
9      <h1>Welcome to My Website</h1>
10     <p>This is a paragraph with a
11     <a href="https://www.example.com">link</a>
12     to an external website.</p>
13     
14 </body>
15 </html>
16
```

Output:



Explanation:

- The `<a>` tag creates a hyperlink.
- The `` tag is used to display an image with the `src` attribute specifying the image file and `alt` providing alternative text.

➤ HTML Elements

Definition:

An HTML element is a component of an HTML document, consisting of a start tag, content (optional), and an end tag. Elements define the structure and content on a webpage, such as headings, paragraphs, images, and links.

Structure of an HTML Element:

- **Start Tag** : Indicates the beginning of an element (e.g., `<p>` for a paragraph).
- **Content** : The information or other elements enclosed within the tags (e.g., text or images).
- **End Tag** : Indicates the closing of an element (e.g., `</p>`).

Example:

```
html
```

```
<p>This is a paragraph element.</p>
```

- `<p>` : Start tag.
- This is a paragraph element : Content.
- `</p>` : End tag.

Self-Closing Elements:

Some elements do not require a closing tag because they do not contain content. These are known as self-closing elements (or void elements).

Example of Self-Closing Elements:

```
html


<br />
```

- **** : Used to embed an image. It is self-closing because it does not have any internal content.
- **
** : Inserts a line break.

➤ Types of HTML Elements:

1. Block-Level Elements:

- These elements take up the full width available and start on a new line.
- They typically define sections of a webpage.

Examples:

- **<h1>** to **<h6>** (Headings)
- **<p>** (Paragraph)
- **<div>** (Division/Container)
- **<header>**, **<footer>**, **<article>**, **<section>**, etc.

Example:

```
html

<h1>This is a heading</h1>
<p>This is a paragraph.</p>
```


2. Inline Elements:

- Inline elements only take up as much width as necessary and do not start on a new line.
- Typically used for styling small parts of the document.

Examples:

- `<a>` (Anchor/Link)
- `` (Text container for styling)
- `` (Image)
- `` , `` , `<code>` , etc.

Example:

```
html
```

```
<p>This is a <strong>bold</strong> word in a paragraph.</p>
```

3. Semantic Elements:

- These elements provide meaning to the content, making it more understandable for both browsers and developers. They help with SEO and accessibility.

Examples:

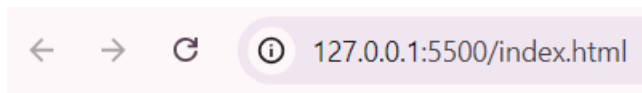
- `<header>` (Defines the header of a document or section)
- `<footer>` (Defines the footer)
- `<article>` (Defines an independent, self-contained piece of content)
- `<nav>` (Defines navigation links)
- `<aside>` (Defines content related to the surrounding content)

Example:

```
html

<header>
  <h1>My Website</h1>
  <nav>
    <a href="#home">Home</a>
    <a href="#about">About</a>
    <a href="#contact">Contact</a>
  </nav>
</header>
```

Output:



My Website

[Home](#) [About](#) [Contact](#)

➤ Nesting HTML Elements:

HTML elements can be nested, meaning one element can contain other elements inside it.

Example of Nested Elements:

```
html Copy code

<div>
  <h2>My Blog</h2>
  <p>This is a blog post with a <a href="https://example.com">link</a> inside it.</p>
</div>
```

- The `<div>` element contains an `<h2>` heading and a `<p>` paragraph.
- Inside the paragraph, there's an `<a>` element for a link.

➤ Commonly Used HTML Elements:

1. Headings (<h1> to <h6>):

- Defines different levels of headings.

```
html

<h1>Main Heading</h1>
<h2>Subheading</h2>
```

2. Paragraph (<p>):

- Defines a block of text as a paragraph.

```
html

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

3. Anchor (<a>):

- Defines a hyperlink to another webpage or resource.

```
html

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

4. Image ():

- Embeds an image into the webpage.

```
html


```

5. List (, ,):

- : Unordered list (bulleted).
- : Ordered list (numbered).
- : List item.

Example:

```
html

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

<ol>
  <li>Step 1</li>
  <li>Step 2</li>
</ol>
```

➤ Empty Tags in HTML

Definition:

Empty tags, also known as void elements, are HTML elements that do not contain any content between their opening and closing tags. They are self-contained and do not require a closing tag.

Characteristics of Empty Tags:

- They are self-closing.
- They do not have any content or child elements.
- They perform specific tasks like inserting line breaks, images, or meta-information.

Syntax:

In HTML5, it is not mandatory to close empty tags with a slash (/>), but some older versions of HTML or XHTML require the closing slash. Modern browsers interpret both formats correctly.

Common Empty Tags:

1.
 (Line Break):

- Inserts a line break in the text.
- Does not require an end tag.

Example:

Html Code

```
<p>This is the first line.<br>This is the second line.</p>
```

2. (Image):

- Embeds an image into the webpage.
- Requires the src attribute for the image source and alt attribute for alternative text.

Example:

Html Code

```

```

3. <hr> (Horizontal Rule):

- Inserts a horizontal line to visually separate content.
- Commonly used to divide sections.

Example:

Html Code

```
html
```

```
<p>This is some text.</p>  
<hr>  
<p>This is after the horizontal line.</p>
```

4. <input> (Input Field):

- Defines an input field for forms.
- Used with various attributes like type , name , and value .

Example:

Html Code

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

5. <meta> (Metadata):

- Provides metadata about the HTML document (such as charset, viewport settings, keywords for SEO).
- Used inside the <head> section.

Example:

Html Code

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

```
<meta name="description" content="A brief description of the webpage">
```

6. <link> (Link to External Resources):

- Links external resources such as stylesheets or icons to the HTML document.
- Always used in the <head> section.

Example:

Html Code

```
<link rel="stylesheet" href="styles.css">
```

7. <source> (Media Source):

- Specifies multiple media resources for elements like <video> or <audio> .
- Does not contain any content.

Example:

Html Code

```
html

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

8. <area> (Image Map Area):

- Defines an area within an image map, linking different regions of an image to URLs.

Example:

Html Code

```

<map name="planetmap">
  <area shape="rect" coords="34,44,270,350" alt="Mercury" href="mercury.html">
</map>
```

Key Points to Remember:

- Empty tags are self-closing and perform tasks without requiring content.
- Some commonly used empty tags include
 , , and <meta> .
- In HTML5, closing slashes (/>) are optional, but it's a good habit to use them in self-closing tags, especially when ensuring compatibility with XHTML.

Case Sensitivity in HTML:

HTML is not case-sensitive. This means that HTML tags and attributes can be written in uppercase, lowercase, or a combination of both, and they will still be interpreted the same way by browsers.

However, the standard convention (best practice) in HTML5 is to write tags and attributes in lowercase for consistency and readability.

```
<!DOCTYPE html>
<html>
<head>
  <title>Case Sensitivity Example</title>
</head>
<body>
  <H1>This is a heading</H1>
  <P>This is a paragraph.</P>
</body>
</html>
```

Output:



This is a heading

This is a paragraph.

In the above example, `<H1>` and `<h1>` are treated the same by the browser, but it's recommended to use lowercase (`<h1>`).

Note:

- CSS and JavaScript are case-sensitive, so when working with IDs, classes, or script, the exact case should be used.

➤ HTML Attributes

HTML attributes provide additional information about HTML elements. They are placed within the start tag of an element and usually come in name/value pairs. Attributes modify the behavior or appearance of elements.

Common HTML Attributes:

1. href (Hypertext Reference):

- Specifies the URL for a link.
- Used within the <a> tag.

Example:

Html Code

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

2. src (Source):

- Specifies the URL of an image or other media files.
- Used within the , <script>, and <iframe> tags.

Example:

html

```

```

3. width :

- Specifies the width of an element, such as an image or a table.
- Used with the , <table>, and other elements.

Example:

Html Code

```

```

4. height :

- Specifies the height of an element.
- Used with the , <table> , and other elements.

Example:

Html Code

```

```

5. style:

- Applies inline CSS styles to an element.
- Used with any HTML tag to define styling directly within the element.

Example:

Html Code

```
<p style="color: blue; font-size: 16px;">This is a styled paragraph.</p>
```

6. title :

- Provides additional information about an element, usually displayed as a tooltip when hovering over the element.
- Used with most HTML tags.

Example:

Html Code

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

7. alt (Alternative Text):

- Provides a text alternative for an image if the image cannot be displayed.
- Used with the tag.

Example:

Html Code

```

```

8. id:

- Provides a unique identifier for an element, used for styling with CSS or scripting with JavaScript.
- Must be unique within a document.

Example:

Html Code

```
<div id="header">This is the header</div>
```

9. class :

- Assigns one or more class names to an element, allowing CSS styling and JavaScript manipulation.
- Multiple classes can be separated by spaces.

Example:

Html Code

```
<p class="text-bold highlight">This is a bold and highlighted paragraph.</p>
```

10. name :

- Provides a name for form elements, used to identify form fields when submitting data.
- Used with elements like <input> , <textarea> , and <select> .

Example:

Html code

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

11. value :

- Defines the value of an input element, such as a text field or a button.
- Used with form elements.

Example:

Html Code

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

12. placeholder:

- Provides a hint within an input field about what the user should enter.
- Used with <input> and <textarea> .

Example:

Html Code

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

13. required :

- Specifies that an input field must be filled out before submitting the form.
- Used with form elements.

Example:

Html Code

```
<input type="text" name="email" required>
```

Key Points:

- Attributes are used to provide additional details and modify the behavior or appearance of HTML elements.
- Most attributes come in name/value pairs.
- Some attributes like href , src , style , and title are commonly used across various HTML elements.

➤ lang and quote Attributes in HTML

1. lang Attribute:

Definition:

The lang attribute specifies the language of the content within an HTML element. It helps search engines and browsers understand and properly render the text, and it is also useful for screen readers and other assistive technologies.

Usage:

- Can be applied to the <html> , <body> , or any other HTML element.
- The value is a language code, which can be a primary language (e.g., en for English) or a language code with regional information (e.g., en-US for American English, en-GB for British English).

Example:

Html Code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document in English</title>
</head>
<body>
  <p lang="fr">Bonjour, comment ça va?</p>
  <p lang="es">Hola, ¿cómo estás?</p>
</body>
</html>
```

- In the example, the <html> element specifies that the default language for the page is English.
- The <p> elements specify different languages (French and Spanish) for the text content.

2. quote Attributes:

HTML does not have a `quote` attribute; however, there are specific HTML elements related to quotations:

a. `<blockquote>`:

- Used to indicate a section of text that is a block quotation from another source.
- Typically used for longer quotations and automatically adds indentations.

Example:

Html Code

```
<blockquote cite="https://www.example.com">  
  <p>"The only limit to our realization of tomorrow is our doubts of today."</p>  
  <footer>— Franklin D. Roosevelt</footer>  
</blockquote>
```

- `cite` attribute can be used to provide a URL of the source of the quote.

b. `<q>` :

- Used to indicate a short inline quotation.
- Typically adds quotation marks around the text.

Example:

Html Code

```
<p>He said, <q>HTML is a fundamental web technology.</q></p>
```

Summary:

- The `lang` attribute specifies the language of an element's content.
- HTML uses `<blockquote>` and `<q>` elements for quotations, with `<blockquote>` for longer, block-style quotations and `<q>` for shorter, inline quotations.

➤ HTML Headings

Definition:

HTML headings are used to define the structure and hierarchy of content on a webpage. There are six levels of headings, ranging from <h1> to <h6>, with <h1> being the highest (most important) level and <h6> being the lowest (least important) level.

Importance of Headings:

- They help organize content for better readability.
- They enhance SEO (Search Engine Optimization) by indicating the importance of content sections to search engines.
- They assist screen readers in navigating the webpage more easily.

HTML Heading Elements:

1. <h1> :

- The main heading of the page, usually reserved for the title or primary topic.
- Only one <h1> should typically be used per page for semantic clarity.

Example:

Html Code

```
<h1>Welcome to My Website</h1>
```

2. <h2> :

- Subheading under <h1>, used for main sections of content.
- Can be used multiple times on a page.

Example:

Html Code

```
<h2>About Us</h2>
```

3. <h3> :

- Subheading under <h2> , used for subsections within a section.
- Can also be used multiple times.

Example:

Html code

```
<h3>Our Mission</h3>
```

4. <h4> :

- Subheading under <h3> , used for additional subsections.
- Provides further breakdown of content.

Example:

Html code

```
<h4>Short-term Goals</h4>
```

5. <h5> :

- Subheading under <h4> , used for smaller subsections.
- Helps in detailing specific topics within a section.

Html code

```
<h5>Improving Efficiency</h5>
```

6. <h6> :

- The smallest heading, used for the least important subsections.
- Suitable for minor details within content.

Example:

Html code

```
<h6>Team Achievements</h6>
```


Key Points to Remember:

- Use headings to create a clear hierarchy and structure in your content.
- Stick to one <h1> per page to maintain clarity.
- Use <h2> to <h6> to create subsections as needed, allowing for a well-organized document.

Output:



This is a heading 1

This is a heading 2

This is a heading 3

This is a heading 4

This is a heading 5

This is a heading 6

➤ HTML Paragraphs

Definition:

An HTML paragraph is a block of text enclosed within the <p> tag. It is used to define a section of text that is a distinct block and usually separated by vertical space from other blocks of text.

Characteristics:

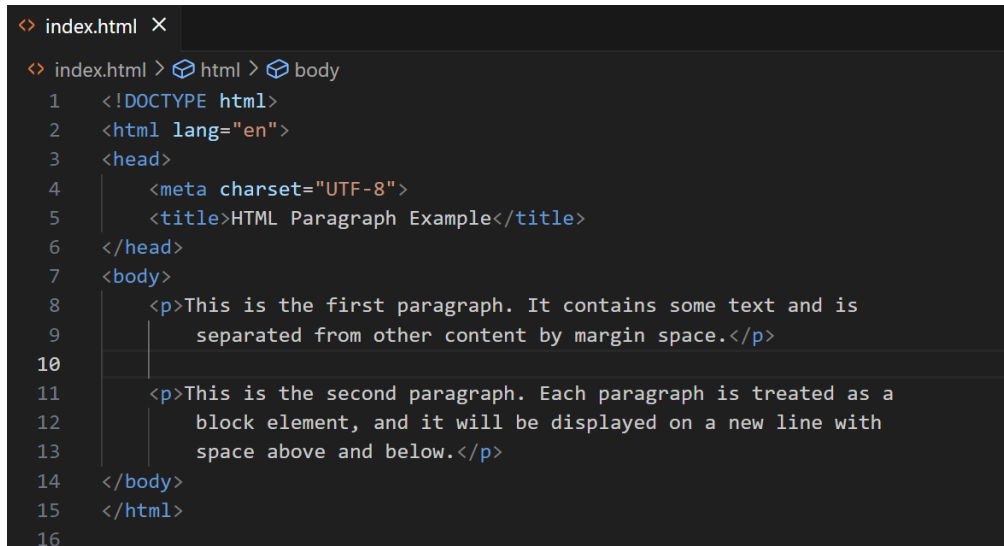
- Paragraphs automatically add space above and below themselves, creating separation from other elements.
- They help in organizing and structuring text content on a webpage.

Basic Syntax:

Html Code

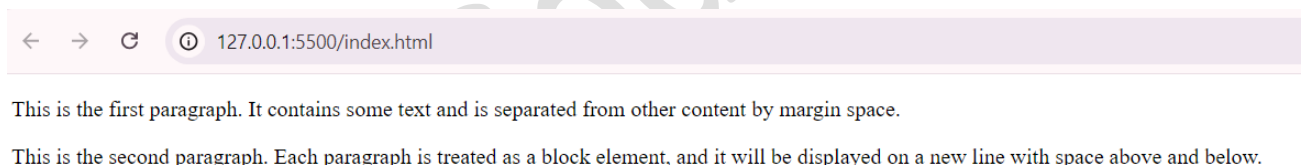
```
<p>This is a paragraph of text.</p>
```

Example:



```
<code>index.html X
index.html > html > body
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <title>HTML Paragraph Example</title>
6  </head>
7  <body>
8    <p>This is the first paragraph. It contains some text and is
9    separated from other content by margin space.</p>
10
11   <p>This is the second paragraph. Each paragraph is treated as a
12   block element, and it will be displayed on a new line with
13   space above and below.</p>
14 </body>
15 </html>
16</code>
```

Output:



```
<code>← → ↻ ⓘ 127.0.0.1:5500/index.html

This is the first paragraph. It contains some text and is separated from other content by margin space.

This is the second paragraph. Each paragraph is treated as a block element, and it will be displayed on a new line with space above and below.</code>
```

Key Points:

1. Automatic Spacing:

- Browsers automatically add margin above and below `<p>` elements, so you don't need to manually add line breaks for spacing.

2. Text Wrapping:

- Text within a `<p>` tag automatically wraps to the next line when it reaches the end of the container.

3. Nesting:

- You cannot nest `<p>` tags within each other. If you need to group paragraphs or text blocks, consider using `<div>` or `<section>` tags.

Incorrect:

Html Code

```
<p>This is a paragraph.  
  <p>This is another paragraph inside the first one.</p>  
</p>
```

Correct:

Html code

```
<p>This is the first paragraph.</p>  
<p>This is a second paragraph, properly separated from the first one.</p>
```

4. Formatting Text:

- For inline text formatting, such as bold or italic, use tags like `` or `` inside the `<p>` tag.

Example:

Html Code

```
<p>This is a paragraph with <strong>bold</strong> and <em>italic</em>  
text.</p>
```

Advanced Usage:

1. Styling Paragraphs:

- You can style paragraphs using CSS to adjust margins, padding, font size, line height, and more.

Example with CSS:

Html Code

```
<style>
  p {
    margin: 20px 0;
    font-size: 16px;
    line-height: 1.5;
  }
</style>
<p>This is a styled paragraph with increased margin and custom font size.</p>
```

2. Handling Long Text:

- For long texts or articles, use paragraphs to break the content into manageable sections for better readability.

➤ HTML Styles

Definition:

In HTML, styles are used to control the appearance and layout of elements. Styles can be applied directly within HTML using the `style` attribute or through external CSS (Cascading Style Sheets).

Inline Styles with the `style` Attribute

Syntax:

The `style` attribute allows you to apply CSS styles directly to an HTML element.

Example:

Html code

```
<> index.html > p
1  <p style="color: blue;
2      font-size: 18px;
3      text-align: center;
4      background-color: lightyellow;
5      font-family: Arial, sans-serif;">
6      This is a styled paragraph with inline CSS.
7  </p>
8
```

Common CSS Properties

1. color :

- Sets the text color of an element.

Example:

Html Code

```
<p style="color: red;">This text is red.</p>
```

2. background-color :

- Sets the background color of an element.

Example:

Html Code

```
<p style="background-color: lightblue;">This paragraph has a light blue background.</p>
```

3. font-family :

- Specifies the font of the text. Multiple fonts can be listed as fallback options.

Example:

Html Code

```
<p style="font-family: Times New Roman , serif;">This text is in Times New Roman font.</p>
```

4. font-size :

- Defines the size of the font. Can use values in pixels (px), ems (em), rems (rem), or percentages.

Example:

Html Code

```
<p style="font-size: 20px;">This text has a font size of 20 pixels.</p>
```

5. text-align:

- Aligns the text within its container. Options include left , right , center , and justify .

Example:

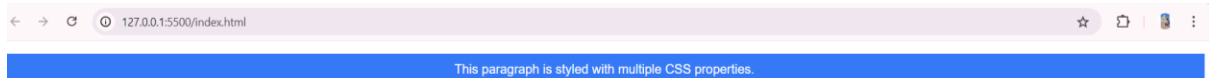
Html Code

```
<p style="text-align: center;">This text is centered.</p>
```

Example Combining Multiple Styles:

Html Code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>HTML Styles Example</title>
</head>
<body>
  <p style="
    color: white;
    background-color: #007BFF;
    font-family: 'Arial', sans-serif;
    font-size: 16px;
    text-align: center;
    padding: 10px;
  ">
    This paragraph is styled with multiple CSS properties.
  </p>
</body>
</html>
```



Key Points:

- Inline styles are applied directly to individual elements using the `style` attribute.
- For complex or site-wide styles, it's better to use external or internal CSS to keep HTML clean and separate content from design.
- The `style` attribute is useful for quick styling or testing, but for maintainability and scalability, CSS files or `<style>` blocks are preferred.

HTML Text Formatting

HTML provides various tags for formatting text to emphasize, highlight, or adjust the appearance of content. Here's an overview of commonly used text formatting tags:

1. Bold (``) :

- **Definition:** Renders text in bold font without implying any additional importance.
- **Usage:** Primarily used for stylistic purposes.

Example:

Html Code

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

2. Italic (`<i>`) :

- **Definition :** Renders text in italic font without implying any additional emphasis.
- **Usage :** Used for stylistic purposes like emphasizing a word or phrase.

Example:

Html Code

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

3. Strong (``) :

- **Definition :** Renders text in bold font and also implies that the text is of strong importance.

- Usage : Used to give emphasis on text, indicating its importance or seriousness.

Example:

Html Code

```
<p>This is <strong>strong</strong> text.</p>
```

4. Emphasized () :

- Definition : Renders text in italic font and implies emphasis or stress.
- Usage : Used to emphasize a part of the text.

Example:

Html code

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

5. Marked (<mark>) :

- Definition : Highlights text with a yellow background, similar to a highlighter pen.
- Usage : Used to mark or highlight parts of the text for reference.

Example:

Html code

```
<p>This is <mark>highlighted</mark> text.</p>
```

6. Deleted () :

- Definition : Renders text with a strikethrough, indicating that it has been deleted or is no longer relevant.
- Usage : Used to show text that has been removed or changed.

Example:

Html code

```
<p>This text is <del>deleted</del> but was here before.</p>
```


7. Inserted (<ins>) :

- Definition : Renders text with an underline, indicating that it has been inserted or added.

- Usage : Used to show newly added text.

Example:

Html Code

```
<p>This is <ins>inserted</ins> text.</p>
```

8. Small (<small>) :

- Definition : Renders text in a smaller font size.

- Usage : Used for fine print or less important information.

Example:

Html Code

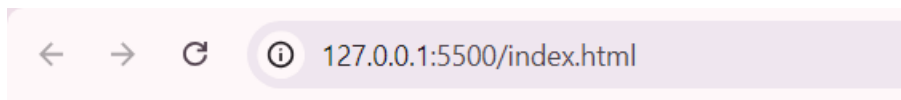
```
<p>This is <small>small</small> text.</p>
```

➤ Example Combining Multiple Formats:

Html Code

```
<? index.html X  Extension: Live Server  Release Notes: 1.93.0
<? index.html > html > body > p
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <title>HTML Text Formatting Example</title>
6  </head>
7  <body>
8  |   <p><strong>Strong text</strong> and <em>emphasized text</em>
9  |   with <mark>highlighted</mark> and <small>small</small> text.</p>
10 |   <p>Some <del>deleted</del> and <ins>inserted</ins> text examples.</p>
11 </body>
12 </html>
13
```

Output:



Strong text and *emphasized text* with highlighted and small text.

Some ~~deleted~~ and inserted text examples.

Key Points:

- `` and `<i>` are purely stylistic, without semantic meaning.
- `` and `` provide semantic meaning and are used for emphasis and importance.
- `<mark>` highlights text, `` shows deleted content, and `<ins>` indicates inserted content.
- `<small>` reduces the font size of the text.

➤ HTML Comments

Definition:

HTML comments are used to insert notes or explanations within the HTML code. These comments are not displayed in the browser but can be viewed in the page's source code. They are useful for documenting code, leaving notes for other developers, or temporarily disabling code.

Syntax:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>HTML Comments Example</title>
</head>
<body>
  <!-- This is a comment and will not be displayed in the browser -->
  <p>This is a visible paragraph.</p>

  <!-- Comments can also be used to explain code -->
  <p>This paragraph is styled with inline CSS.</p>
  <p style="color: blue;">This text is blue.</p>
</body>
</html>
```

Key Points:

- Comments are not rendered: The content inside `<!--` and `-->` does not appear on the web page.
- Usage: Helpful for code documentation, providing explanations, or disabling code temporarily.
- Structure: Comments must not be nested. Nested comments are not allowed and can break the HTML rendering.

HTML Colors

Definition:

HTML colors are used to define the color of text, backgrounds, borders, and other elements on a web page. Colors can be specified using color names, HEX codes, RGB values, or HSL values.

1. Color Names

HTML supports 140 standard color names that can be used directly in CSS.

Example:

Html Code

```
<p style="color: red;">This text is red.</p>
```

```
<p style="color: blue;">This text is blue.</p>
```

2. HEX Codes

HEX codes are a six-digit code that represents colors in hexadecimal format. The code starts with a # followed by six hexadecimal digits (two for red, two for green, and two for blue).

Example:

Html Code

```
<p style="color: #FF5733;">This text is in a specific shade of orange.</p>
```

```
<p style="background-color: #D3D3D3;">This background is light gray.</p>
```

3. RGB Values

RGB (Red, Green, Blue) values specify color using three integers (0-255) representing the intensity of red, green, and blue light.

Example:

Html Code

```
<p style="color: rgb(0, 255, 0);">This text is green.</p>
```

```
<p style="background-color: rgb(255, 255, 0);">This background is yellow.</p>
```

4. HSL Values

HSL (Hue, Saturation, Lightness) represents colors in terms of hue (degrees on a color wheel), saturation (percentage), and lightness (percentage).

Example:

Html Code

```
<p style="color: hsl(240, 100%, 50%);">This text is a bright blue.</p>
```

```
<p style="background-color: hsl(120, 100%, 50%);">This background is a bright green.</p>
```

Example Combining All Color Methods:

```
<?xml version="1.0" encoding="UTF-8" ?>
<index.html>
  <!DOCTYPE html>
  <html lang="en">
    <head>
      <meta charset="UTF-8">
      <title>HTML Colors Example</title>
    </head>
    <body>
      <h1 style="color: blue;">Welcome to My Colorful Page</h1>
      <p style="color: #FF5733;">This paragraph uses a HEX color.</p>
      <p style="color: rgb(0, 255, 0);">This paragraph uses an RGB color.</p>
      <p style="color: hsl(120, 100%, 50%);">This paragraph uses an HSL color.</p>
      <p style="background-color: #D3D3D3; padding: 10px;">This paragraph has a light gray background.</p>
    </body>
  </html>
</index.html>
```

Key Points:

- Colors can be specified using color names , HEX codes , RGB , or HSL .
- Choosing the right method depends on the use case and preference.
- Using CSS for colors keeps HTML clean and maintains separation of content and style.

➤ HTML Links

Definition:

HTML links, created using the <a> (anchor) tag, are used to connect one page to another, either on the same website or on a different one. Links can also point to specific parts of a page or trigger actions like sending emails.

Basic Syntax

Html Code

```
<a href="URL">Link Text</a>
```

Common Attributes

1. href :

- Specifies the URL of the page the link goes to.
- This attribute is required for links

Example:

Html Code

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

2. target :

- Specifies where to open the linked document.
- Common values:
 - `_self` : Opens the link in the same frame (default).
 - `_blank` : Opens the link in a new tab or window.

- `_parent` : Opens the link in the parent frame.
- `_top` : Opens the link in the full body of the window.

Example:

Html Code

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

3. title :

- Provides additional information about the link, typically displayed as a tooltip when the mouse hovers over it.

Example:

Html Code

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

4. rel :

- Specifies the relationship between the current document and the linked document.

- Common values:

- `noopener` : Prevents the new page from being able to access the `window.opener` property.
- `noreferrer` : Prevents the browser from sending the HTTP referer header when navigating to the link.
- `nofollow` : Tells search engines not to follow the link.

Example:

Html Code

```
<a href="https://www.example.com" target="_blank" rel="noopener noreferrer">Secure Link</a>
```

Example of HTML Links with Attributes:

```
<> index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <title>HTML Links Example</title>
6  </head>
7  <body>
8  |   <h1>Welcome to My Website</h1>
9  |   <p>
10 |       Visit <a href="https://www.example.com" target="_blank" rel="noopener noreferrer"
11 |         title="Go to Example.com">Example.com</a> for more information.
12 |   </p>
13 |   <p>
14 |       Go to the <a href="#section1">First Section</a> of this page.
15 |   </p>
16 |
17 |   <h2 id="section1">First Section</h2>
18 |   <p>This is the content of the first section.</p>
19 | </body>
20 </html>
21
```

Key Points:

- The `<a>` tag is used to create links in HTML.
- The `href` attribute is essential for specifying the link's destination.
- The `target` attribute controls how the link is opened, while `title` and `rel` provide additional information and control link behavior.
- Links can point to external websites, internal pages, or specific sections within the same page.

➤ Absolute URL vs. Relative URL

Absolute URL :

An absolute URL provides the complete address of a resource on the web, including the protocol (such as `http://` or `https://`), the domain name, and the path to the specific file or resource. It can be used to link to any resource on the internet regardless of the location of the current page.

Example of an Absolute URL :

Html code

```
<a href="https://www.example.com/images/photo.jpg">View Photo</a>
```

Relative URL :

A relative URL provides a partial address to a resource, which is relative to the current document's location. It does not include the protocol and domain name. This is useful for linking to pages or resources within the same website.

Example of a Relative URL :

Html code

```
<a href="/images/photo.jpg">View Photo</a>
```

- Here, /images/photo.jpg points to a resource in the "images" directory relative to the root of the current website.

Key Points:

- Absolute URLs are complete links that can point to resources anywhere on the web.
- Relative URLs are shorter and easier to manage, linking to resources within the same site.

➤ HTML Images

Definition:

Images in HTML are embedded using the tag, which allows you to display pictures or graphics on a web page. The tag is a self-closing tag and does not require a closing tag.

Syntax

Html Code

```

```


➤ Common Attributes

1. src (Source):

- Specifies the path to the image file. This can be an absolute URL or a relative URL.

- Example :

Html code

```

```

2. alt (Alternative Text):

- Provides a textual description of the image. This is important for accessibility, as it describes the image to users who cannot see it (e.g., screen readers) and is displayed if the image fails to load.

- Example :

Html code

```

```

3. width:

- Specifies the width of the image in pixels or as a percentage of the containing element.

- Example:

Html code

```

```

4. height :

- Specifies the height of the image in pixels or as a percentage of the containing element.

- Example:

Html code

```

```

5. title :

- Provides additional information about the image, usually displayed as a tooltip when the mouse hovers over the image.

- Example :

Html code

```

```

Example of Using the Tag:

```
<> index.html > html > body
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <title>HTML Images Example</title>
6  </head>
7  <body>
8    <h1>Image Display</h1>
9    
10
11
12 </body>
13 </html>
14
15
```



Image Display



Key Points:

- The `` tag is used to embed images in HTML documents.
- The `src` attribute is required, while the `alt` attribute is essential for accessibility.
- The `width` and `height` attributes can be used to control the display size of the image.

➤ HTML Favicon

Definition:

A favicon (short for "favorite icon") is a small icon associated with a website, displayed in the browser's address bar, tabs, bookmarks, and history.

Syntax:

To include a favicon in your HTML document, use the `<link>` tag within the `<head>` section of your HTML.

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>HTML Favicon Example</title>
  <link rel="icon" href="favicon.ico" type="image/x-icon" />
</head>
<body>
  <h1>Page with Favicon</h1>
</body>
</html>
```

- rel="icon" specifies that the linked file is an icon.
- href="favicon.ico" points to the location of the favicon file.

➤ HTML Page Title

Definition:

The page title, defined using the <title> tag, is the text displayed in the browser's title bar or tab. It provides a brief description of the page's content and is crucial for SEO and user navigation.

Syntax:

The <title> tag is placed inside the <head> section of the HTML document.

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>HTML Page Title Example</title>
</head>
<body>
  <h1>Welcome to My Website</h1>
</body>
</html>
```

Key Points:

- **Favicon** : Small icon shown in the browser tab or address bar, specified using the `<link>` tag.
- **Page Title** : Text displayed in the browser tab, specified using the `<title>` tag within the `<head>` section.

HTML Lists

HTML supports three types of lists: ordered lists, unordered lists, and description lists. Each type is used for different purposes.

1. Ordered Lists (``)

Definition:

An ordered list is used for items that need to be displayed in a specific sequence. Each item in the list is numbered.

Syntax:

Html code

```
<ol>
```

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

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

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

```
</ol>
```

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Ordered List Example</title>
</head>
<body>
  <h1>My Favorite Fruits</h1>
  <ol>
    <li>Apple</li>
    <li>Banana</li>
    <li>Cherry</li>
  </ol>
</body>
</html>
```

2. Unordered Lists ()

Definition:

An unordered list is used for items that do not need to be in a specific order. Each item is marked with a bullet point.

Syntax:

Html code

```
<ul>
  <li>Item one</li>
  <li>Item two</li>
  <li>Item three</li>
</ul>
```

Example:

Html code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Unordered List Example</title>
</head>
<body>
  <h1>My Hobbies</h1>
  <ul>
    <li>Reading</li>
    <li>Traveling</li>
    <li>Cooking</li>
  </ul>
</body>
</html>
```

3. Description Lists (<dl>)

Definition:

A description list is used for listing terms and their descriptions. Each item in a description list consists of a term (<dt>) and its description (<dd>).

Syntax:

Html Code

```
<dl>

  <dt>Term 1</dt>

  <dd>Description for term 1</dd>

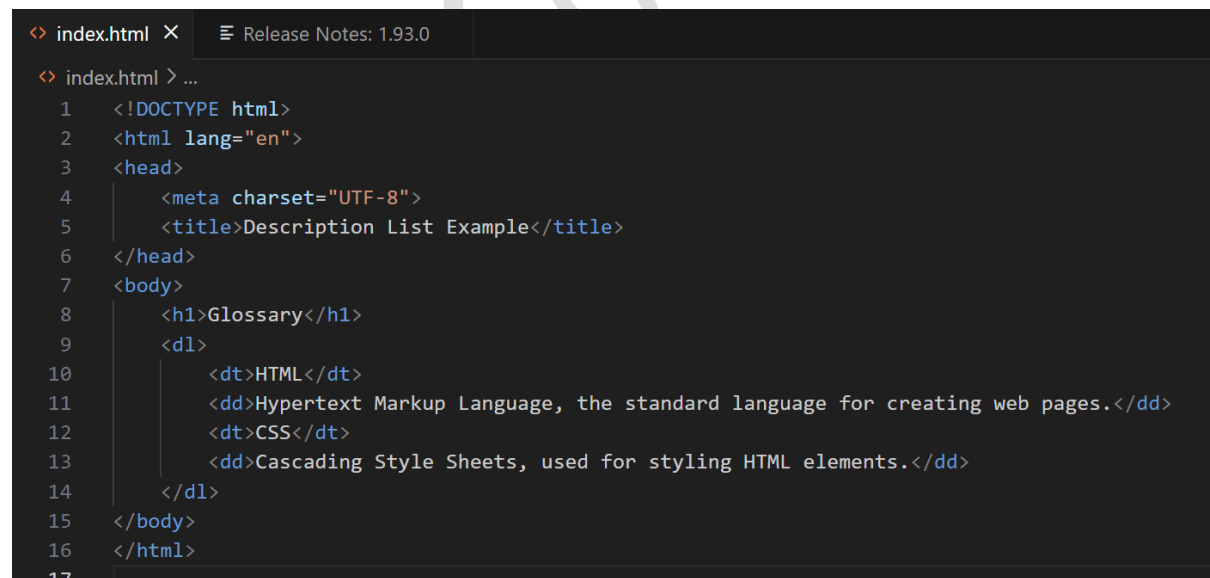
  <dt>Term 2</dt>

  <dd>Description for term 2</dd>

</dl>
```

Example:

Html code



```
< index.html X  Release Notes: 1.93.0

< index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <title>Description List Example</title>
6  </head>
7  <body>
8    <h1>Glossary</h1>
9    <dl>
10     <dt>HTML</dt>
11     <dd>Hypertext Markup Language, the standard language for creating web pages.</dd>
12     <dt>CSS</dt>
13     <dd>Cascading Style Sheets, used for styling HTML elements.</dd>
14   </dl>
15 </body>
16 </html>
17
```

Example Code of Ordered And Unordered List:

```
<> index.html > html > body > ul
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Ordered and Unordered Lists</title>
7  </head>
8  <body>
9      <!-- Ordered List Example -->
10     <h2>Ordered List</h2>
11     <ol>
12         <li>Item 1</li>
13         <li>Item 2</li>
14         <li>Item 3</li>
15     </ol>
16     <!-- Unordered List Example -->
17     <h2>Unordered List</h2>
18     <ul>
19         <li>Item A</li>
20         <li>Item B</li>
21         <li>Item C</li>
22     </ul>
23 </body>
24 </html>
```

Output:

← → ↻ ⓘ 127.0.0.1:5500/index.html

Ordered List

1. Item 1
2. Item 2
3. Item 3

Unordered List

- Item A
- Item B
- Item C

Key Points:

- Ordered Lists (``) : Items are numbered and used when the order matters.
- Unordered Lists (``) : Items are marked with bullet points and used when order does not matter.
- Description Lists (`<dl>`) : Used for defining terms and descriptions.

HTML Div, Classes, and IDs

1. HTML `<div>` Element

Definition:

The `<div>` element is a block-level container used to group other HTML elements together. It is commonly used for layout purposes and applying styles to a section of a webpage.

Syntax:

Html code

```
<div>  
  
<!-- Other HTML elements go here -->  
  
</div>
```

Example:

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <title>Div Example</title>  
  <style>  
    .box {  
      border: 1px solid black;  
      padding: 20px;  
      margin: 10px;  
      background-color: lightgrey;  
    }  
  </style>  
</head>  
<body>  
  <div class="box">  
    <h2>Welcome</h2>  
    <p>This is a simple div element.</p>  
  </div>  
</body>  
</html>
```

2. HTML Classes

Definition:

Classes are used to apply the same styles or behavior to multiple elements. You can assign one or more classes to an HTML element using the `class` attribute.

Syntax:

Html code

```
<div class="class-name">
```

```
<!-- Content -->
```

```
</div>
```

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Class Example</title>
  <style>
    .red {
      color: red;
    }
    .blue {
      color: blue;
    }
  </style>
</head>
<body>
  <div class="red">
    <p>This text is red.</p>
  </div>
  <div class="blue">
    <p>This text is blue.</p>
  </div>
</body>
</html>
```



3. HTML IDs

Definition:

IDs are unique identifiers assigned to HTML elements. Each ID must be unique within a page, meaning no two elements can have the same ID. You can use the `id` attribute to apply styles or manipulate elements with JavaScript.

Syntax:

Html Code

```
<div id="unique-id">
```

```
<!-- Content -->
```

```
</div>
```

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>ID Example</title>
  <style>
    #highlight {
      background-color: yellow;
      font-weight: bold;
    }
  </style>
</head>
<body>
  <div id="highlight">
    <p>This text is highlighted.</p>
  </div>
  <div id="normal">
    <p>This text is normal.</p>
  </div>
</body>
</html>
```

Key Points:

- <div>: A block-level container for grouping elements.
- Classes (class attribute) : Used for applying styles to multiple elements; can have multiple classes.
- IDs (id attribute) : Unique identifiers for individual elements; should be used once per page.

HTML Block and Inline Elements

1. Block-Level Elements

Definition:

Block-level elements occupy the full width available, creating a new line before and after the element. They are used to structure the layout of a webpage, and they often contain other block-level or inline elements.

Characteristics:

- Start on a new line.
- Extend the full width of their parent container.
- Can contain other block-level and inline elements.

Examples:

- <div>
- <p>
- <h1> , <h2> , <h3> , etc.
- , ,
- <header> , <footer> , <section> , <article>

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>ID Example</title>
  <style>
    #highlight {
      background-color: yellow;
      font-weight: bold;
    }
  </style>
</head>
<body>
  <div id="highlight">
    <p>This text is highlighted.</p>
  </div>
  <div id="normal">
    <p>This text is normal.</p>
  </div>
</body>
</html>
```

2. Inline Elements

Definition:

Inline elements occupy only the space necessary for their content and do not start on a new line. They are used within block-level elements to style small pieces of content or add functionality.

Characteristics:

- Do not create a new line before or after the element.
- Occupy only the width of their content.
- Cannot contain block-level elements.

Examples:

- <a>
-
- ,
-
-
 , <input>

Example:

Html Code

```
<> index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Block-Level Elements</title>
6  </head>
7  <body>
8      <h1>Header</h1>
9      <p>This is a paragraph. Block-level elements create
10     | a new line before and after.</p>
11     <div>
12         <p>Another paragraph inside a div.</p>
13     </div>
14 </body>
15 </html>
16
```

Key Differences:

- Block-Level Elements : Start on a new line, take up the full width, and can contain other block-level and inline elements.
- Inline Elements : Do not start on a new line, take up only the width needed for their content, and cannot contain block-level elements.

HTML Tables

Definition:

HTML tables are used to display tabular data, organizing it into rows and columns. They are created using a combination of the `<table>`, `<tr>`, `<td>`, and `<th>` elements.

Basic Syntax

1. `<table>` : Defines the table.
2. `<tr>` : Defines a table row.
3. `<td>` : Defines a table cell (data).
4. `<th>` : Defines a table header cell.

Example

Basic Table:

Html code

```
<> index.html X  Release Notes: 1.93.0
<> index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>HTML Table Example</title>
6      <style>
7          table {
8              width: 100%;
9              border-collapse: collapse;
10         }
11         th, td {
12             border: 1px solid black;
13             padding: 8px;
14             text-align: left;
15         }
16         th {
17             background-color: #f2f2f2;
18         }
19     </style>
20 </head>
```

```
<body>
  <h1>Sample Table</h1>
  <table>
    <thead>
      <tr>
        <th>Name</th>
        <th>Age</th>
        <th>City</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>John Doe</td>
        <td>30</td>
        <td>New York</td>
      </tr>
      <tr>
        <td>Jane Smith</td>
        <td>25</td>
        <td>Los Angeles</td>
      </tr>
    </tbody>
  </table>
</body>
```

```
      <td>Sam Johnson</td>
      <td>35</td>
      <td>Chicago</td>
    </tr>
  </tbody>
</table>
</body>
</html>
```

Output:

← → ↻ 127.0.0.1:5500/index.html ☆ 📄 📱 ⋮

Sample Table

Name	Age	City
John Doe	30	New York
Jane Smith	25	Los Angeles
Sam Johnson	35	Chicago

Attributes

1. border:

- Specifies the width of the table border.
- Note : This is not commonly used in modern HTML but was previously used with the <table> element.

- Example :

html

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

2. colspan :

- Specifies the number of columns a cell should span.
- Used in <td> or <th> .
- Example :

html

```
<td colspan="2">Spans two columns</td>
```

3. rowspan :

- Specifies the number of rows a cell should span.
- Used in <td> or <th> .
- Example :

html

```
<td rowspan="2">Spans two rows</td>
```

4. thead , tbody , tfoot :

- <thead> : Contains header rows.
- <tbody> : Contains body rows.
- <tfoot> : Contains footer rows.

Key Points:

- `<table>` : Defines the table.
- `<tr>` : Defines a row in the table.
- `<th>` : Defines a header cell in the table.
- `<td>` : Defines a standard cell in the table.
- `<thead>` , `<tbody>` , `<tfoot>` : Organize header, body, and footer sections respectively.

HTML Layout

HTML layout techniques are used to structure content on a webpage. Various elements and CSS properties help create responsive, organized, and visually appealing layouts. Here are some fundamental layout techniques:

1. Basic Layout with `<div>`

Definition:

Using `<div>` elements allows you to create various sections of a webpage. Combined with CSS, you can style and position these sections.

Example:

```
index.html X  Release Notes: 1.93.0

index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Basic Layout Example</title>
6      <style>
7          .container {
8              width: 80%;
9              margin: 0 auto;
10             border: 1px solid #ccc;
11         }
12         .header, .footer {
13             background-color: #f2f2f2;
14             padding: 10px;
15             text-align: center;
16         }
17         .main {
18             padding: 20px;
19         }
20     </style>
21 </head>
22 <body>
```

```
20         .sidebar {
21             width: 25%;
22             float: left;
23             background-color: #f9f9f9;
24             padding: 10px;
25         }
26         .content {
27             width: 75%;
28             float: left;
29             padding: 10px;
30         }
31     </style>
32 </head>
33 <body>
```

```
<body>
  <div class="container">
    <div class="header">Header</div>
    <div class="main">
      <div class="sidebar">Sidebar</div>
      <div class="content">Main Content</div>
    </div>
    <div class="footer">Footer</div>
  </div>
</body>
</html>
```

2. Flexbox Layout

Definition:

Flexbox is a CSS layout module that makes it easier to design flexible and responsive layout structures without using floats or positioning.

Example:

```
<> index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Flexbox Layout Example</title>
6      <style>
7          .container {
8              display: flex;
9              flex-direction: column;
10             height: 100vh;
11         }
12         .header, .footer {
13             background-color: #f2f2f2;
14             padding: 10px;
15             text-align: center;
16         }
17         .main {
18             display: flex;
19             flex: 1;
20         }
```

```
21         .sidebar {
22             width: 25%;
23             background-color: #f9f9f9;
24             padding: 10px;
25         }
26         .content {
27             width: 75%;
28             padding: 10px;
29         }
30     </style>
31 </head>
32 <body>
33     <div class="container">
34         <div class="header">Header</div>
35         <div class="main">
36             <div class="sidebar">Sidebar</div>
37             <div class="content">Main Content</div>
38         </div>
39         <div class="footer">Footer</div>
40     </div>
41 </body>
42 </html>
```

3. Grid Layout

Definition:

CSS Grid Layout provides a powerful way to create complex web layouts using rows and columns. It s especially useful for creating two-dimensional layouts.

Key Points:

- Basic Layout with <div> : Use <div> elements for structuring content; control layout with CSS properties like float and width .
- Flexbox Layout : Provides a flexible and responsive way to align and distribute space among items in a container.
- Grid Layout : Offers a powerful grid-based layout system for creating complex layouts with rows and columns.

Examples:

```
<> index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Grid Layout Example</title>
6      <style>
7          .container {
8              display: grid;
9              grid-template-areas:
10                 "header header header"
11                 "sidebar content content"
12                 "footer footer footer";
13              grid-template-columns: 1fr 3fr;
14              grid-template-rows: auto 1fr auto;
15              height: 100vh;
16          }
17          .header {
18              grid-area: header;
19              background-color: #f2f2f2;
20              padding: 10px;
21              text-align: center;
22
23          .sidebar {
24              grid-area: sidebar;
25              background-color: #f9f9f9;
26              padding: 10px;
27          }
28          .content {
29              grid-area: content;
30              padding: 10px;
31          }
32          .footer {
33              grid-area: footer;
34              background-color: #f2f2f2;
35              padding: 10px;
36              text-align: center;
37          }
38      </style>
39  </head>
40  <body>
41      <div class="container">
42          <div class="header">Header</div>
43          <div class="sidebar">Sidebar</div>
44          <div class="content">Main Content</div>
45          <div class="footer">Footer</div>
46      </div>
47  </body>
48  </html>
```

HTML Forms

Definition:

HTML forms are used to collect user input. They are a fundamental part of web applications, allowing users to submit data to a server. Forms can include various input types, such as text fields, checkboxes, radio buttons, and more.

Basic Structure of an HTML Form

1. `<form>` : The container for all form elements. It may include attributes such as `action` (the URL to send the form data) and `method` (the HTTP method to use, e.g., GET or POST).
2. Input Elements : Various types of input fields that allow users to enter data.

```
<> index.html > html > body > form > br
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <title>HTML Form Example</title>
6  </head>
7  <body>
8    <h1>Contact Form</h1>
9    <form action="/submit" method="POST">
10     <label for="name">Name:</label>
11     <input type="text" id="name" name="name" required>
12     <br><br>
13
14     <label for="email">Email:</label>
15     <input type="email" id="email" name="email" required>
16     <br><br>
17
18     <label for="message">Message:</label>
19     <textarea id="message" name="message" rows="4" required></textarea>
20     <br><br>
21     <input type="submit" value="Submit">
22   </form>
23 </body>
24 </html>
```

Example

Basic Form Example:

Common Input Types

1. Text Field (`<input type="text">`) : For single-line text input.

Html code

```
<input type="text" name="username">
```

2. Password Field (<input type="password">) : For password input, masking the characters.

Html code

```
<input type="password" name="password">
```

3. Email Field (<input type="email">) : For email input; validates the format.

Html code

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

4. Checkbox (<input type="checkbox">) : Allows selection of one or more options.

Html code

```
<input type="checkbox" name="subscribe" value="yes"> Subscribe to  
newsletter
```

5. Radio Button (<input type="radio">) : Allows selection of one option from a set.

Html code

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

6. Dropdown Menu (<select>) : Allows selection from a list of options.

Html code

```
<label for="country">Country:</label>  
<select id="country" name="country">  
  <option value="us">United States</option>  
  <option value="ca">Canada</option>  
  <option value="uk">United Kingdom</option>
```


</select>

7. Textarea (<textarea>) : For multi-line text input.

Html code

```
<textarea name="message" rows="4" cols="50"></textarea>
```

8. File Upload (<input type="file">) : Allows users to upload files.

Html code

```
<input type="file" name="fileUpload">
```

Form Attributes

- action : The URL to which the form data will be sent.
- method : The HTTP method for sending form data (GET or POST).
- enctype : The encoding type for form data (used primarily for file uploads).

Key Points:

- Forms are essential for gathering user input.
- Various input types allow for diverse data collection methods.
- The action and method attributes are crucial for form submission

Example of All Input Types

Html code

```
<> index.html > form > br
1  <form action="/submit" method="POST">
2    <input type="text" name="username" placeholder="Username">
3    <br>
4    <input type="password" name="password" placeholder="Password">
5    <br>
6    <input type="email" name="email" placeholder="Email">
7    <br>
8    <input type="checkbox" name="subscribe" value="yes"> Subscribe
9    <br>
10   <input type="radio" name="gender" value="male"> Male
11   <input type="radio" name="gender" value="female"> Female
12   <br><br>
13   <select name="country">
14     <option value="us">United States</option>
15     <option value="ca">Canada</option>
16   </select>
17   <br><br>
18   <textarea name="message" rows="4" placeholder="Your message"></textarea>
19   <input type="file" name="fileUpload">
20   <br><br>
21   <input type="submit" value="Submit">
22 </form>
```

← → ↻ ⓘ 127.0.0.1:5500/index.html

Username

Password

Email

☐ Subscribe

☐ Male ☐ Female

United States ▼

Your message

Choose File No file chosen

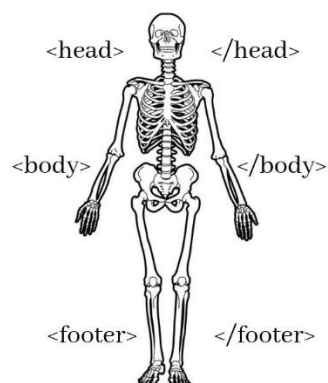
Submit

HTML VS XHTML

HTML	XHTML
HTML stands for Hypertext Markup Language	XHTML stands for Extensible Hypertext Markup Language
It is an SGML application	It is an XML application
Tim Berners-Lee proposed it in 1987	The World Wide Web Consortium recommended it in 2000
HTML is not case sensitive	XHTML is case sensitive
HTML uses a format that is similar to document formats	XHTML uses markup language
HTML can use open tags, such as 	All unclosed tags must be closed in XHTML
HTML is less expressive	XHTML is more expressive as compared to HTML
HTML is not mandatory for a single root element	XHTML documents must contain at least one root element
All content can be included in the body element	All contents must be put in blocks
Attribute values are not significant in HTML	Attribute values are important in XHTML
There is no hard rule on the structure of the elements	The structure of the elements should be followed

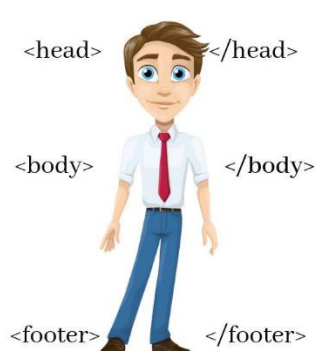
HTML Vs CSS

HTML



Structural Layer

HTML + CSS



Presentational Layer

One Shot HTML Video on YouTube:

1. HTML in One Video (Hindi) : [Click me](#)
2. HTML in One Shot (English): [Click Me](#)
3. HTML Complete Tutorial : [Click Me](#)

Techie CodeBuddy