HTML

What is HTML?

HTML (HyperText Markup Language) was created by Tim Berners-Lee in 1991 as a standard for creating web pages. It's a markup language used to structure content on the web, defining elements like headings, paragraphs, links, and images. HTML forms the backbone of web content. In layman's terms, HTML is like the skeleton of a website. It's a set of instructions that tells a web browser how to display text, images, videos, and other elements on a webpage. Think of it as the building blocks that create the structure and look of a website, similar to how bricks and mortar are used to build a house.

- HTML is the language of the web, used to create websites.
- HTML defines the barebone structure or layout of web pages that we see on the Internet.
- HTML consists of a set of tags contained within an HTML document, and the associated files typically have either a **.html** or **.htm** extension.
- There are several versions of HTML, with HTML5 being the most recent version.

Features of HTML

1. It is platform-independent. For example, Chrome displays the same pages identically across different operating systems such as Mac, Linux, and Windows.
2. Images, videos, and audio can be added to a web page (For example - YouTube shows videos on their website).
3. HTML is a markup language and not a programming language.
4. It can be integrated with other languages like CSS, JavaScript, etc. to show interactive (or dynamic) web pages.

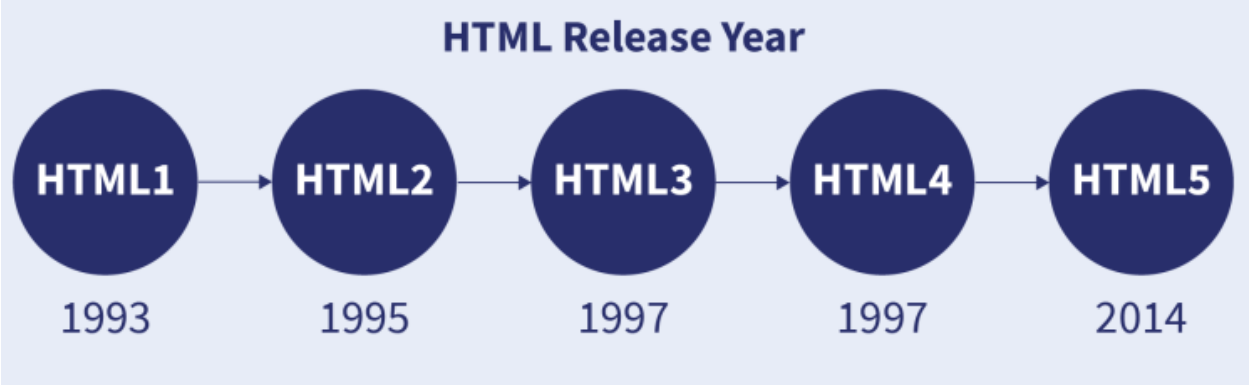
Why the Term HyperText & Markup Language?

The term 'Hypertext Markup Language' is composed of two main words: 'hypertext' and 'markup language.' 'Hypertext' refers to the linking of text with other documents, while 'markup language' denotes a language that utilizes a specific set of tags.

Thus, HTML is the practice of displaying text, graphics, audio, video, etc., in a certain way using special tags.

Note: Tags are meaningful texts enclosed in angle braces, like '<...>'. For example, the " tag. Each tag has a unique meaning and significance in building an HTML page, and it can influence the web page in various ways.

History of HTML



HTML Page Structure

An HTML document is structured using a set of nested tags. Each tag is enclosed within <...> angle brackets and acts as a container for content or other HTML tags.

Basic HTML document structure:

```
<!DOCTYPE html>
<html>
<head>
    <title>Document</title>
</head>
<body>
   <!-- content -->
</body>
</html>
```

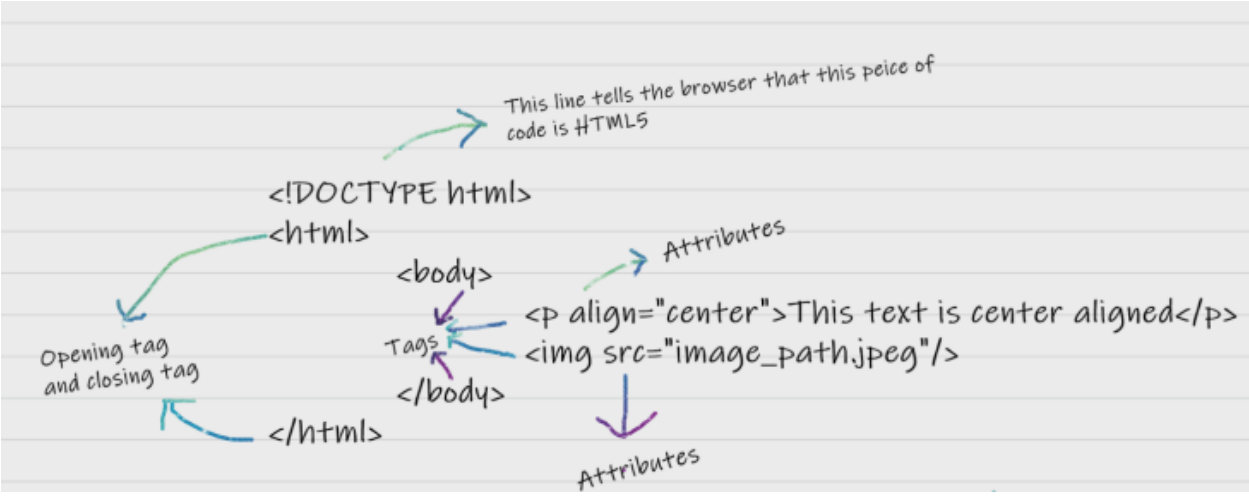
How This Content Appears in a Web Browser:

Consider this HTML code:

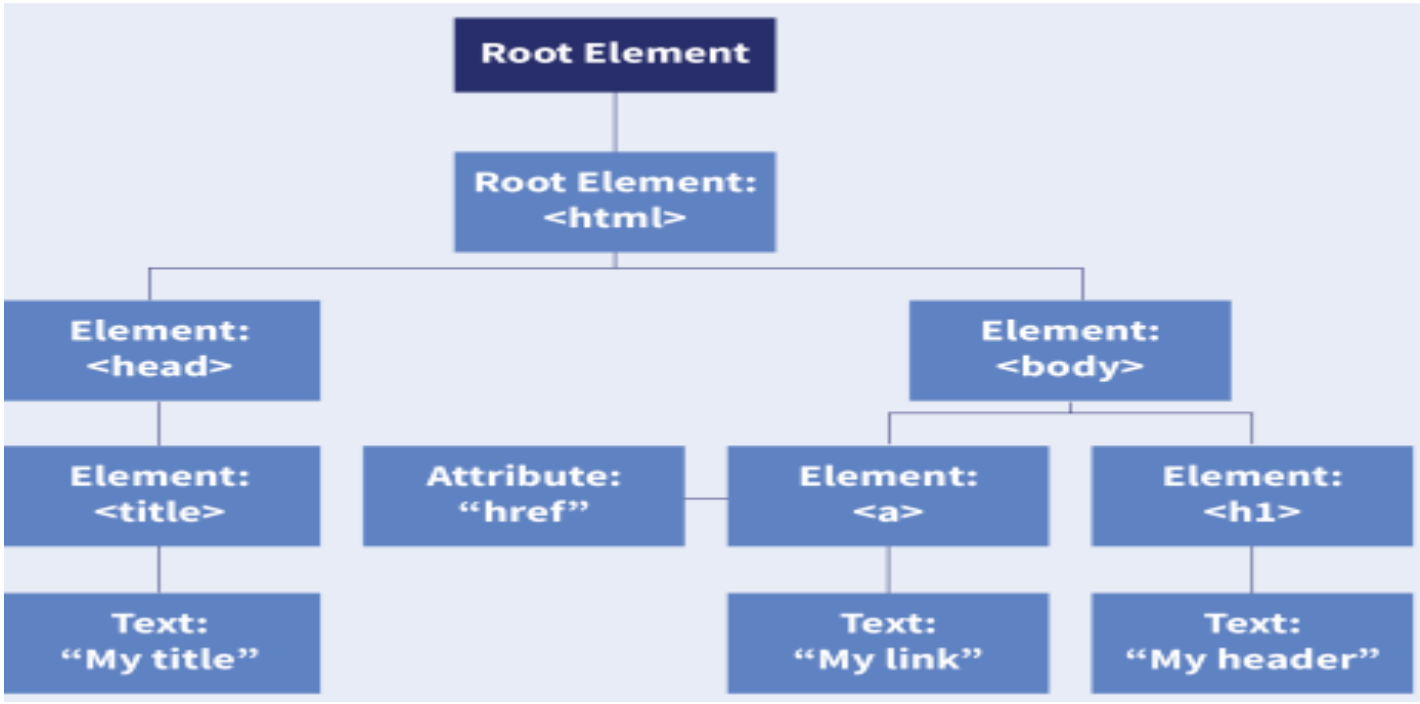
```
<!DOCTYPE html>
<html>
<head>
    <title>Document</title>
</head>
<body>
    <h1> This is a heading</h1>
    <p>This is a paragraph</p>
</body>
</html>
```

Below is an image showing how this HTML document will be rendered in a web browser:

Basics Understanding of HTML Syntax



HTML Tree Structure:



HTML Tag:

Document Structure Tags

1. `<DOCTYPE html>`: Specifies the document type.
2. `<html>`: Encloses the entire HTML document.
3. `<head>`: Contains meta-information and links to scripts and stylesheets.
4. `<body>`: Contains the content of the web page.

Text Formatting Tags

1. `<p>`: Paragraph.
2. `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`: Headings.
3. `<strong>`: Strong emphasis (typically bold).
4. `<em>`: Emphasis (typically italic).
5. `<br>`: Line break.
6. `<hr>`: Horizontal rule.

List Tags

1. `<ul>`: Unordered list.
2. `<ol>`: Ordered list.
3. `<li>`: List item.

Hyperlink and Media Tags

1. `<a>`: Anchor (used for links).
2. `<img>`: Image.
3. `<audio>`: Audio content.
4. `<video>`: Video content.

Table Tags

1. `<table>`: Table.
2. `<tr>`: Table row.
3. `<td>`: Table data cell.
4. `<th>`: Table header cell.
5. `<thead>`: Table header group.
6. `<tbody>`: Table body group.
7. `<tfoot>`: Table footer group.

Semantic Tags

1. `<header>`: Header section.
2. `<footer>`: Footer section.
3. `<article>`: Article.
4. `<section>`: Section.
5. `<nav>`: Navigation.
6. `<aside>`: Sidebar content.

Form Tags

1. `<form>`: Form.
2. `<input>`: Input field.
3. `<textarea>`: Text area.
4. `<button>`: Button.
5. `<select>`: Dropdown list.
6. `<option>`: Options within a `<select>` or `<datalist>`.

Self-closing tags:

1. Line Break: `<br/>`
2. Horizontal Rule: `<hr/>`
3. Image: `<img src="image.jpg" alt="An example image"/>`

HTML Attributes

- All HTML elements can have attributes.
- Attributes provide additional information about an element.
- Attributes are always specified in the start tag.
- Attributes usually come in name/value pairs like：name="value".

HTML attributes are used to define the characteristics of an HTML element. They are placed within the element's opening tag and consist of two parts: the name and the value.

- Name: Specifies the property for that element.
- Value: Sets the value of that property for the element.

HTML Id & Classes

HTML offers multiple ways to select and style elements. Two of the most commonly used selectors are IDs and Classes. This blog explores what they are, how to use them, and their differences.

What is an ID?

An ID is an attribute, a unique identifier assigned to only one HTML element within a page. It is often used for unique styling and JavaScript manipulations

```
<div id="myUniqueID">This is a div with an ID.</div>
```

What are Classes?

The `class` attribute lets you give the same name to multiple HTML elements. That way, you can easily change their look or behavior all at once. Classes are not unique and can be assigned to multiple elements. They are generally used for applying the same styles or behaviors to a group of elements.

```
<div class="myClass">This is a div with a class.</div>
<p class="myClass">This is a paragraph with the same class.</p>
```

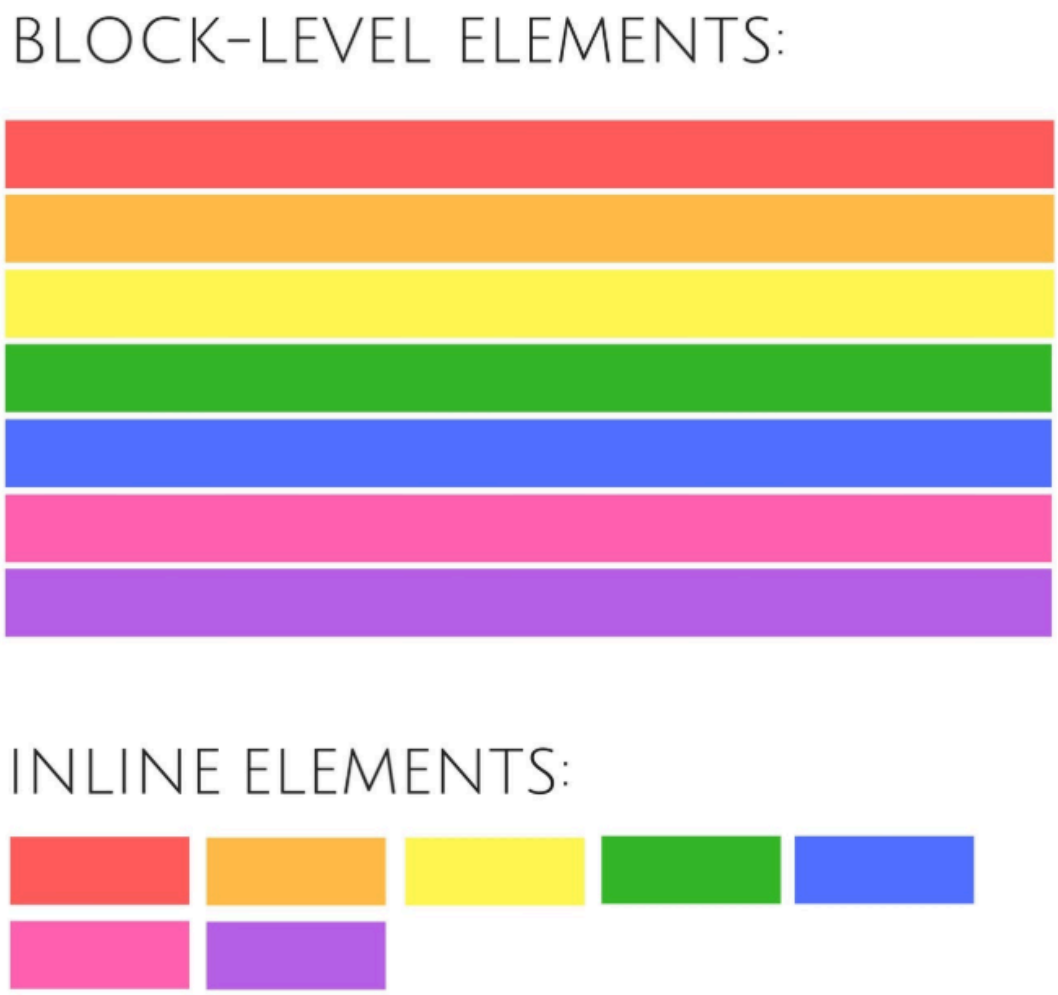
Differences Between IDs and Classes

- Uniqueness: IDs are unique, and classes can be reused.
- JavaScript: IDs are often used for JavaScript operations.
- Styling: Both can be used for styling, but IDs have higher specificity.

HTML Inline Elements

Inline Elements don't start on a new line. It only takes the width required to cover the content.

HTML elements are generally divided into two categories: Block-level and Inline elements.



What are Inline Elements?

Inline elements do not start on a new line and only take up as much width as necessary. They are part of the flow within other elements.

Inline elements can contain other inline elements, but they generally should not contain block-level elements. For example, you could nest a `<strong>` (strong emphasis) element within a `<span>` (a generic inline container) element, like so:

```
<span>This is <strong>important</strong> text.</span>
```

However, placing a block-level element like a `<div>` or `<p>` inside an inline element like `<span>` or `<a>` is typically considered incorrect HTML and could lead to unexpected behavior in terms of layout and styling.

```
<!-- This is generally considered incorrect -->
<span>This is <div>not recommended</div> text.</span>
```

Common Inline Elements

- `<span>`: A generic inline container for text
- `<a>`: Defines a hyperlink
- `<strong>`: Defines important text
- `<em>`: Defines emphasized text
- `<img>`: Embeds an image
- `<input>`: Defines an input control

Examples

Here is an example of using inline elements within a paragraph.

This text contains a link, an important text, and an *emphasized text*.

Styling Inline Elements

You can use CSS to style inline elements. However, some properties like `width` and `height` may not apply.

Here is an exhaustive list of the most used Inline Elements:

- `<a>`
- `<abbr>`
- `<acronym>`
- `<button>`
- `<br>`
- `<big>`
- `<bdo>`
- `<b>`
- `<cite>`
- `<code>`
- `<dfn>`
- `<i>`
- `<em>`
- `<img>`
- `<input>`
- `<kbd>`
- `<label>`
- `<map>`
- `<object>`
- `<output>`
- `<tt>`
- `<time>`
- `<samp>`
- `<script>`
- `<select>`

- `<small>`
- `<span>`
- `<strong>`
- `<sub>`
- `<sup>`
- `<textarea>`

HTML Block Elements

You already know about HTML inline elements. All HTML tags have specific display behavior: they are either block-level elements or inline elements.

What are Block-level Elements?

Block-level elements are those that start on a new line and take up the entire width of their container by default. They essentially claim all the horizontal space for themselves, pushing any content that comes after them to a new line.

Characteristics of Block-level Elements:

- Always start on a new line.
- Take up the full width available.
- Width and height can be controlled via CSS.
- Can contain other block-level as well as inline elements.

Common Block-level Elements:

- `<h1>,<h2>,<h3>,<h4>,<h5>,<h6>` - Headings

- `<p>` - Paragraphs
- `<hr>` - Horizontal rule
- `<address>` - Address information
- `<article>` - Article content
- `<aside>` - Sidebar content
- `<blockquote>` - Block quotations
- `<canvas>` - Drawing area
- `<dd>` - Description in a description list
- `<div>` - Generic container
- `<dl>` - Description list
- `<dt>` - Term in a description list
- `<fieldset>` - Group of related form elements
- `<figcaption>` - Caption for a figure
- `<figure>` - Image or media with a caption
- `<footer>` - Footer of a section or page
- `<form>` - HTML form
- `<header>` - Header of a section or page
- `<li>` - List item
- `<main>` - Main content of a document
- `<nav>` - Navigation links
- `<noscript>` - Alternate content when JavaScript is not enabled
- `<ol>` - Ordered list
- `<ul>` - Unordered list
- `<pre>` - Preformatted text
- `<section>` - Standalone section in a document
- `<table>` - Table
- `<video>` - Video content

HTML Lists

Our day-to-day lives often involve the use of lists. For example, when we go shopping, the bill we receive includes a list of all the items we've purchased. In a similar manner, web developers use lists to neatly display data on websites.

Types of HTML Lists

HTML provides different types of lists to display data in various forms. Each list contains one or more list items.

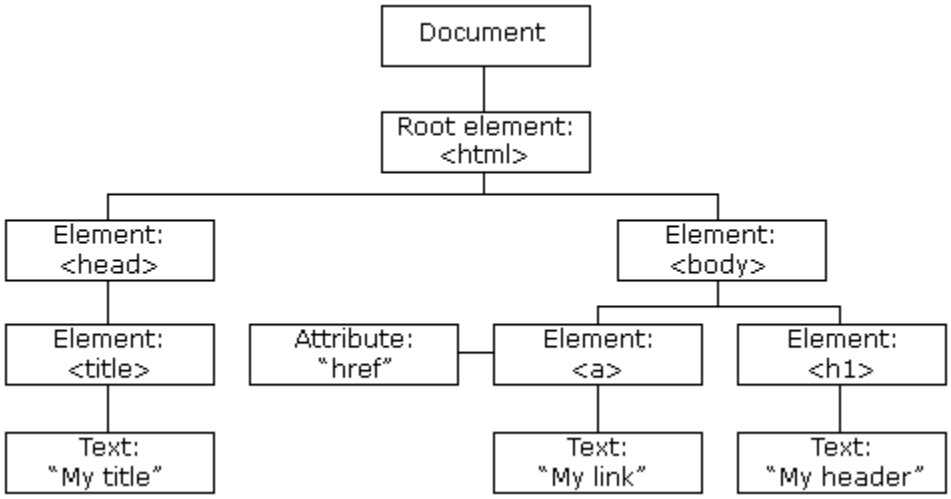
- Unordered List: Displays items using bullets.
- Ordered List: Displays items in a numerical sequence, and supports various numbering styles like Arabic numerals, Roman numerals, and so on.
- Definition List: Organizes items in a format similar to a dictionary, with terms and their corresponding definitions.

HTML DOM

What is HTML DOM ?

- The DOM is a W3C (World Wide Web Consortium) standard.
- The W3C DOM standard is separated into 3 different parts
    - Core DOM - standard model for all document types
    - XML DOM - standard model for XML documents
    - HTML DOM - standard model for HTML documents
      We will dive into HTML DOM in following content
- When a web page is loaded, the browser creates a Document Object Model (*DOM*) of the page.

The HTML DOM model is constructed as a tree of Objects:



The HTML DOM is a standard object model and programming interface for HTML. It defines:

- The HTML elements as objects
- The properties of all HTML elements
- The methods to access all HTML elements
- The events for all HTML elements

HTML Tables

HTML tables allow you to arrange data like text, images, and links in rows and columns. You use the `<table>` tag to start and end a table.

Syntax of HTML Table

```
<table>  // table content</table>
```

Key Elements of HTML Table

- `<table>`: Defines the table itself.
- `<tr>`: Used for table rows.
- `<th>`: Used for table headings.
- `<td>`: Used for table cells (data).

Basic Table Structure

```
<table>
  <tr>
    <th>Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Harry</td>
    <td>100</td>
  </tr>
</table>
```

Rowspan and Colspan Attributes

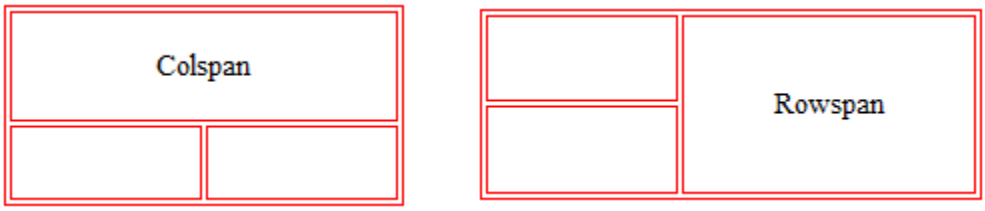
Rowspan: If you want a table cell to span multiple rows, you can use the `rowspan` attribute.

```
<td rowspan="value">
```

Colspan: If you want a table cell to span multiple columns, you can use the `colspan` attribute.

```
<td colspan="value">
```

Visual Representation of Rowspan and Colspan



Examples

Here are simple examples to demonstrate the use of `rowspan` and `colspan` in HTML tables.

Example for Colspan:

```
<table border="1">
  <tr>
    <td colspan="2">Merged Columns</td>
  </tr>
  <tr>
    <td>Column 1</td>
    <td>Column 2</td>
  </tr>
</table>
```

Example for Rowspan:

```
<table border="1">
  <tr>
    <td>Row 1, Column 1</td>
    <td rowspan="2">Merged Rows</td>
  </tr>
  <tr>
    <td>Row 2, Column 1</td>
  </tr>
</table>
```

Adding a Caption

To add a title to your table, you can use the `<caption>` element. This element helps both in terms of SEO and accessibility.

```
<table>
  <caption>Student Details</caption>
  <!-- Rest of the table here -->
</table>
```

Introduction to HTML Forms

HTML forms are essential for collecting user input on web pages. Whether it's a search bar, a login screen, or a multi-field registration form, HTML forms play a key role in web interactions. They enable users to submit data, which can be processed, stored, or returned by a server.

Why Do We Use Forms?

Forms serve as the gateway between the user and the server, allowing for dynamic, interactive web experiences. They are crucial for tasks such as user authentication, data submission, feedback collection, and more. Simply put, forms make websites more engaging and functional.

HTML Forms Structure:

The fundamental structure of an HTML form is encapsulated within the `<form>` tags. Inside these tags, you'll place various form controls like text fields, checkboxes, radio buttons, and buttons for submitting the form.

```
<form action="/submit" method="post">
    <!-- Text input for username -->
    <label for="username">Username:</label>
    <input type="text" id="username" name="username" required>
    <br><br>

    <!-- Password input -->
    <label for="password">Password:</label>
    <input type="password" id="password" name="password" required>
    <br><br>

    <!-- Radio buttons for gender -->
    <label>Gender:</label>
    <input type="radio" id="male" name="gender" value="male">
    <label for="male">Male</label>
    <input type="radio" id="female" name="gender" value="female">
    <label for="female">Female</label>
    <br><br>

    <!-- Submit button -->
    <input type="submit" value="Submit">
</form>
```

How to Use Form Controls?

The `<input>` tag is commonly used to create form controls. The attributes of this tag define the control's behavior.

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

The "type" attribute specifies the type of input control (e.g., text, password, checkbox).

The "name" attribute is used for identifying the control, especially when the data is sent to the server.

The "value" attribute sets a default value for the control, which the user can overwrite.

HTML Input Types

Input types in HTML forms are the backbone of interactive web applications. They allow users to send information to web servers for various purposes like searching, logging in, or providing feedback. In this blog, we'll explore common HTML input types: text, password, radio, and checkbox.

Text Input

The text input type is the most basic form of input and is widely used for collecting simple text data.

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

In the above example, the `placeholder` attribute provides a hint to the user about what to enter.

Password Input

The password input type is similar to the text type but hides the characters entered by the user for security reasons.

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

Radio Buttons

Radio buttons are used when you want the user to select only one option from a set of choices.

```
<input type="radio" id="male" name="gender" value="male">
<label for="male">Male</label>
<input type="radio" id="female" name="gender" value="female">
<label for="female">Female</label>
```

Checkbox

Checkboxes allow the user to select multiple options from a set.

```
<input type="checkbox" id="subscribe" name="subscribe" value="yes">
<label for="subscribe">Subscribe to newsletter</label>
```

Input Types

Input Type	Description
text	Allows the user to type a single line of text.
password	Allows the user to type a password.
submit	Represents a button that, when pressed, submits the form.
reset	Represents a button that, when pressed, resets all the form controls to their initial values.
radio	Represents an option in a set of options that are mutually exclusive with each other.
checkbox	Represents an option in a set that may be selected independently of other options.
button	Represents a clickable button.

color	Allows the user to select a color.
date	Allows the user to select a date.
datetime-local	Allows the user to select a date and time with no time zone.
email	Allows the user to enter an email address.
file	Allows the user to select one or more files from their device storage.
hidden	Represents a value that is not displayed but is submitted to the server.
image	Defines an image that acts as a submit button.
month	Allows the user to select a month and year.

number	Allows the user to enter a number.
range	Allows the user to select a number from a range.
search	Allows the user to enter a search query string.
tel	Allows the user to enter a telephone number.
time	Allows the user to select a time.
url	Allows the user to enter a URL.
week	Allows the user to select a week.

Textarea & Select

In addition to the basic input types, HTML forms offer other controls like `textarea` and `select` for richer user interaction. These elements allow for more complex data collection and provide a better user experience. In this blog, we will dive into these form controls and provide examples.

The Textarea Element

The `textarea` element is used when you need multiline text input from the user. This is particularly useful for comments, reviews, or any other type of input where the length is unpredictable.

```
<textarea name="comment" rows="4" cols="50">
      Enter your comment here...
</textarea>
```

The `rows` and `cols` attributes define the visible dimensions of the textarea.

The Select Element

The `select` element creates a dropdown menu for the user. It is useful when you have a predefined list of options for the user to choose from.

```
<select name="fruits">
      <option value="apple">Apple</option>
      <option value="banana">Banana</option>
      <option value="cherry">Cherry</option>
</select>
```

Each `option` inside the `select` tag represents an item in the dropdown list.

HTML Meta Tags

The `<meta>` tags in HTML provide metadata about the HTML document. Metadata is data (information) about data. `<meta>` tags always go inside the document's `<head>` tag and are typically used to specify the character set, page description, keywords, author, and other metadata.

Below is an example HTML code snippet that includes various types of `<meta>` tags commonly used:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8"> <!-- Character encoding -->
    <meta name="viewport" content="width=device-width, initial-scale=1.0"> <!-- Responsive design -->
    <meta http-equiv="X-UA-Compatible" content="ie=edge"> <!-- Internet Explorer compatibility -->
    <meta name="description" content="This is a description of the web page"> <!-- Description for search engines -->
    <meta name="keywords" content="HTML, CSS, JavaScript"> <!-- Keywords for search engines -->
    <meta name="author" content="Your Name"> <!-- Author name -->
    <title>Document</title>
</head>
<body>
    <!-- Your content here -->
</body>
</html>
```

Explanation of each meta tag:

1. Character Encoding (`charset`): `<meta charset="UTF-8">` sets the character encoding for the webpage. UTF-8 is the most common and recommended.

2. Viewport: `<meta name="viewport" content="width=device-width, initial-scale=1.0">` sets the viewport to scale the page to the screen width, useful for responsive design.
3. IE Compatibility: `<meta http-equiv="X-UA-Compatible" content="ie=edge">` specifies that the page should be rendered using the latest rendering engine available on Internet Explorer.
4. Description: `<meta name="description" content="This is a description of the web page">` provides a brief description of the webpage, which search engines may use in search results.
5. Keywords: `<meta name="keywords" content="HTML, CSS, JavaScript">` specifies keywords for the webpage, which were historically used by search engines but are less relevant today.
6. Author: `<meta name="author" content="Your Name">` indicates the name of the author of the webpage.

Video & Audio Tags

This tutorial aims to provide a comprehensive guide on using `<video>` and `<audio>` tags in HTML to embed media files.

The `<video>` Tag

The `<video>` tag is used to embed video files in an HTML document. It supports multiple attributes to control the video playback.

Example usage:

```
<video src="video.mp4" controls></video>
```

Attributes for `<video>` Tag

- src: Specifies the path to the video file.
- controls: Adds video controls, like play, pause, and volume.
- autoplay: Automatically starts playing the video when the page loads.
- loop: Repeats the video once it ends.
- muted: Mutes the video by default.
- poster: Specifies an image to be displayed before the video starts playing.
- width and height: Specifies the dimensions of the video.

The `<audio>` Tag

The `<audio>` tag is used to embed audio files in an HTML document. It also supports multiple attributes for control.

Example usage:

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

Attributes for `<audio>` Tag

- src: Specifies the path to the audio file.
- controls: Adds audio controls, like play, pause, and volume.
- autoplay: Automatically starts playing the audio when the page loads.
- loop: Repeats the audio once it ends.
- muted: Mutes the audio by default.
- preload: Specifies if and how the audio should be loaded when the page loads ('auto', 'metadata', 'none').

The "preload" attribute can have the following values:

1. none: This is the default value. It indicates that the browser should not preload the audio file at all. The audio file will only start downloading when the user initiates playback.
2. metadata: This value tells the browser to preload only the metadata of the audio file, such as its duration and basic information about the audio. This can be useful if you want to display the audio duration to the user without fully loading the audio data.
3. auto: This value instructs the browser to preload the entire audio file as much as possible without delaying the loading of other important page content. The browser will try to load the audio file in the background so that it's ready to play when the user decides to start it.

iFrames in HTML

iFrames, or Inline Frames, are an integral part of modern web development. They allow you to embed another HTML page within your current page. In this blog, we'll delve into the utility of iFrames, their attributes, and some use-cases.

What is an iFrame?

An iFrame is an HTML element that enables an inline frame for the embedding of external content. Essentially, you can load another web page within a designated area of your current webpage.

Why Use iFrames?

iFrames offer a variety of use-cases:

- Content Isolation: iFrames allow you to isolate third-party content, which can improve security.
- Modularity: Easily embed external plugins, widgets, or content.
- Resource Separation: Content within an iFrame can load separately from the rest of the page.

Basic Syntax

The basic syntax of an iFrame is quite straightforward:

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

Attributes of iFrame

Several attributes can enhance the functionality of an iFrame:

- src: Specifies the URL of the page to embed.
- height and width: Define the dimensions.
- frameborder: Indicates whether to display a border.
- scrolling: Controls the scrollbars.
- name: For targeting the iFrame in JavaScript.

HTML Semantic Tags

HTML5 introduced a range of semantic tags that provide meaning to the structure of web content. This blog will guide you through the importance and usage of these tags.

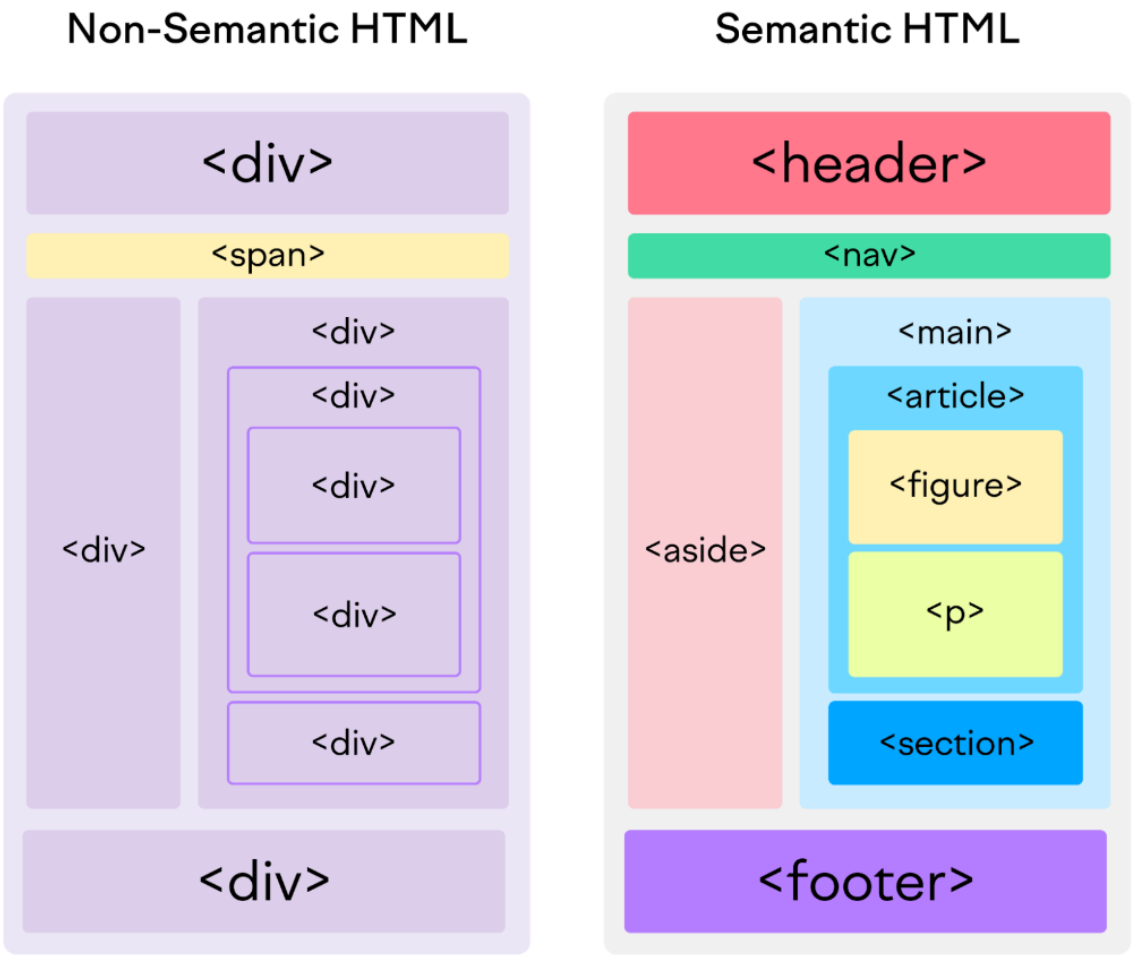
What are Semantic Tags?

Semantic tags add meaning to your HTML. They tell both the browser and the developer what kind of content is being presented.

Here are some of the key semantic tags you must know about:

1. `<header>`: Used to represent the top section of a web page, often containing headings, logos, and navigation.
2. `<nav>`: Signifies a navigation menu on a web page.
3. `<article>`: Indicates a self-contained piece of content, such as a blog post or news article.
4. `<section>`: Represents a thematic grouping of content on a web page.
5. `<aside>`: Typically used for sidebars or content that is tangentially related to the main content.
6. `<footer>`: Represents the footer of a web page, usually containing copyright information and contact details.
7. `<figure>` and `<figcaption>`: Used for embedding images, diagrams, or charts, along with a caption.
8. `<main>`: Signifies the main content area of a web page.
9. `<time>`: Used to represent time-related information, like dates and times.

What Is Semantic HTML?



Why Use Semantic Tags?

They enhance SEO, improve accessibility, and make your code easier to read and maintain.

Commonly Used Semantic Tags

Here are some commonly used semantic tags in HTML:

- header: Contains introductory content.
- footer: Holds footer information.
- article: Encapsulates a self-contained composition.
- section: Represents a standalone section.
- aside: Contains content aside from the content it is placed in.
- nav: Holds navigation links.

HTML Entities

HTML entities are a crucial part of HTML markup language. They enable you to display characters that are reserved in HTML or that aren't readily available on the keyboard. In this blog, we'll explore what HTML entities are, their types, and how to use them.

What Are HTML Entities?

HTML entities are used to represent special characters in a format that the browser can understand. They start with an ampersand (`&`) and end with a semicolon (`;`).

Why Use HTML Entities?

Here are some reasons:

- Reserved Characters: Characters like `<`, `>`, and `&` are reserved in HTML.
- Special Symbols: For symbols like `&copy;`, `&reg;`, or mathematical symbols.
- Non-Breaking Spaces: To create white spaces that won't break into a new line.

https://www.w3schools.com/html/html_entities.asp

https://www.w3schools.com/html/html_symbols.asp

https://www.w3schools.com/html/html_emojis.asp

HTML Quotation Tag

The use of quotations is common in textual content. HTML provides specific tags to handle this: `<blockquote>` for block quotations and `<q>` for inline quotations. In this blog, we'll explore these tags, their attributes, and how to style them.

What Are `<blockquote>` and `<q>` Tags?

The `<blockquote>` and `<q>` tags serve to define quotations in HTML. While `<blockquote>` is used for longer, block-level quotes, `<q>` is used for shorter, inline quotes.

Why Use These Tags?

They provide semantic meaning to your quotations, making it easier for search engines to understand the context and relevance of the content.

Basic Syntax

`<blockquote>` Tag

```
<blockquote cite="source-url">  Quotation text here.</blockquote>
```

`<q>` Tag

```
<q cite="source-url">Quotation text here.</q>
```

Attributes

Both `<blockquote>` and `<q>` tags support the cite attribute:

- cite: Specifies the URL of the quote's source.

Question :Using HTML & CSS Design 8*8 Chess Board

Solution:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Chessboard</title>
  <style>
    .chessboard {
      width: 400px;
      height: 400px;
      display: grid;
      grid-template-columns: repeat(8, 1fr);
      grid-template-rows: repeat(8, 1fr);
      border: 2px solid #333;
    }
    .chessboard div {
      width: 100%;
      height: 100%;
      border: 2px solid black;
    }
    /* Color squares alternately */
    .chessboard div:nth-child(odd) {
      background: #f0d9b5; /* light */
    }
    .chessboard div:nth-child(even) {
      background: #b58863; /* dark */
    }
  </style>
</head>
<body>
  <div class="chessboard">
    <!-- 64 squares -->
    <div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div>
    <div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div>
    <div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div>
    <div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div>
    <div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div>
    <div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div>
    <div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div>
    <div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div>
  </div>
</body>
</html>
```