**Website**

A website is a collection of web pages and related content that is identified by a common domain name and published on at least one web server. It is typically accessible via the Internet or a private local area network (LAN) using a web browser. Websites can serve various purposes, such as providing information, entertainment, commerce, communication, or interaction with users. They may contain text, images, videos, hyperlinks, forms, and other multimedia elements to convey information or facilitate user interaction. Websites can range from simple static pages to complex dynamic web applications with interactive features and databases.

**Web application**

A web application is a software program that users interact with through a web browser over the internet. It's different from a static website because it can perform various functions like processing data, storing information, or allowing users to interact with it in different ways, such as filling out forms, uploading files, or conducting transactions. Examples of web applications include email services like Gmail, social media platforms like Facebook, and online shopping websites like Amazon.

**Library**

Library is a collection of pre-written code modules, functions, or classes that developers can use to perform specific tasks or solve common problems. These libraries are designed to be reusable and often focus on providing functionality for specific areas of development.

**Framework**

In web development a framework is a pre-established collection of libraries, tools, and conventions that provide a structured foundation for building web applications.

**HTML - HyperText Markup Language**

1. **HTML is a Markup Language**
   * **Hypertext**: "Hypertext" refers to text displayed on a computer or other electronic device with references (hyperlinks) to other text that the reader can immediately access. These hyperlinks can lead to different sections of the same document, other documents, or entirely different websites. The concept of hypertext is central to HTML, as it allows for the creation of interconnected documents and is fundamental to the structure of the web..
   * **Markup Language**: A set of rules that defines how the layout and content should be displayed in the browser. It focuses on the presentation and structure of content.
2. **Uses of HTML**
   * **Creating Webpages**: HTML is primarily used to create and design webpages.
   * **Website Skeleton**: It acts as the skeleton of a website, providing the basic structure.
   * **Elements and Tags**: HTML is made up of elements represented by tags, which tell the browser what to display and how to display it.

**Key Concepts**

1. **Tags and Elements**
   * Tags are enclosed in angular brackets, e.g., <tag>.
   * Elements consist of **opening tags, content, and closing tags,**

e.g., <p>Content</p>.

1. **HTML tags are classified :**
2. **Paired Tags or Container tags:** Paired tags have both opening and closing tags, and they enclose content between them. For example, **<p>** and **</p>** in the above example.
3. **Self-Closing Tags or Empty tags or Void tags:** Self-closing tags do not have a separate closing tag, and they are used to insert elements that do not contain any content or do not require closing. For example, **<img>** for an image element, or **<br>** for a line break.

**HTML History**

1. **Initial Version (1991)**
   * Created by Tim Berners-Lee.
   * Consisted of 18 elements.
2. **HTML 2.0 (1995)**
   * The first standardized version of HTML.
3. **HTML 3.0 (1997)**
   * Introduced more advanced features for formatting web pages.
4. **HTML 4.0 (1997)**
   * Included improvements and new features for better webpage design.
5. **HTML 5.0 (2014)**
   * The latest major version.
   * Introduced new elements, attributes, and behaviors.
   * Improved support for multimedia and graphical content.

**Summary**

* **HTML** is the foundational language for creating web pages.
* It uses **tags** and **elements** to define the structure and content.
* The language has evolved significantly since its inception, with HTML5 being the current standard, offering enhanced capabilities for modern web development.

**Basic Structure of HTML**

HTML documents have a fundamental structure that consists of several essential elements:

**Document Declaration Type**

* **<!DOCTYPE html>**: This declaration is placed at the very top of an HTML document. It informs the browser about the type and version of the document, ensuring proper rendering. Omitting this declaration can cause the document to be rendered in "quirks mode," which attempts to mimic older browser behaviors.

**Root Element**

* **<html>**: This is the root element of an HTML document. It encompasses all the content and other HTML elements.

**Head Section**

* **<head>**: This section contains meta-information about the document, such as its title, character set, and links to external resources like stylesheets and scripts. Typical elements within the <head> include:
  + **<title>**: Sets the title of the document, which appears in the browser tab.
  + **<meta>**: Provides metadata such as the character set, author, and viewport settings.
  + **<link>**: Links to external resources like CSS files.

**Body Section**

* **<body>**: This section contains the actual content of the website that is displayed by the browser. All visible elements like paragraphs, headings, images, and other content are placed here.

<!DOCTYPE html>

<html>

<head>

    <title>Page Title</title>

</head>

<body>

    <!-- Content goes here -->

</body>

</html>

**Common HTML Tags**

**Heading tags in html**  
Heading tags in HTML are used to define headings or titles for sections of content on a webpage. They range from <h1> to <h6>, with <h1> being the largest and most important heading, and <h6> being the smallest and least important.

* **<h1>**: Largest heading
* **<h2>**: Second largest heading
* **<h3>**: Third largest heading
* **<h4>**: Fouth largest heading
* **<h5>**: fifth largest or second smallest heading
* **<h6>**: Smallest heading

    <h1>Heading 1</h1>

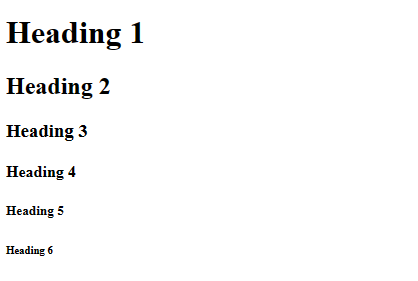
    <h2>Heading 2</h2>

    <h3>Heading 3</h3>

    <h4>Heading 4</h4>

    <h5>Heading 5</h5>

    <h6>Heading 6</h6>



**Preformatted Tag**

* **<pre>**: This tag is used for preformatted text. It preserves spaces, line breaks, and formatting as they are in the HTML code.

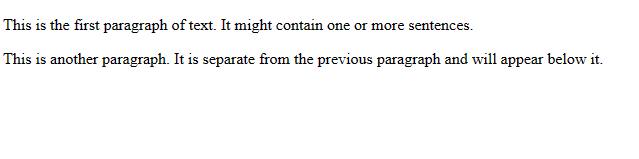
**Paragraph tags**

* **<p>**: This tag is used to define a paragraph. It is a block-level element and automatically adds some space before and after the content.

paragraph tag in HTML, denoted by **<p>**, is used to define paragraphs of text within an HTML document. Paragraphs are blocks of text that are separated from other content by vertical spacing. When the browser renders HTML containing paragraph tags, it automatically adds space above and below the text within the **<p>** tags, visually separating it from surrounding content.

    <p>This is the first paragraph of text. It might contain one or more sentences.</p>

    <p>This is another paragraph. It is separate from the previous paragraph and will appear below it.</p>



**Basic structure of an HTML document:**

<!DOCTYPE html>

<html>

<head>

    <title>Page Title</title>

</head>

<body>

    <!-- Content goes here -->

</body>

</html>

* **<!DOCTYPE html>**: Declares the document type and version of HTML. In this case, it specifies that the document is HTML5.
* **<html >**: This is the root element of the HTML document.
* **<head>**: This section contains meta-information about the HTML document, such as character encoding, viewport settings, and the title of the page.
  + **<title>Page Title</title>**: This sets the title of the HTML document, which appears in the browser's title bar or tab.
* **<body>**: This is where the main content of the HTML document goes. It contains all the visible elements that users see and interact with when they visit the webpage.

**Heading tags in html**  
Heading tags in HTML are used to define headings or titles for sections of content on a webpage. They range from <h1> to <h6>, with <h1> being the largest and most important heading, and <h6> being the smallest and least important.

    <h1>Heading 1</h1>

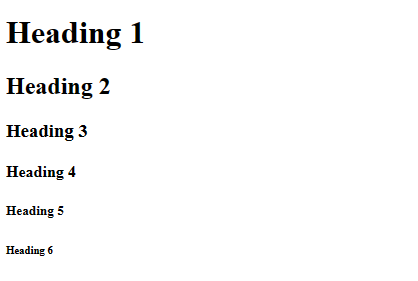
    <h2>Heading 2</h2>

    <h3>Heading 3</h3>

    <h4>Heading 4</h4>

    <h5>Heading 5</h5>

    <h6>Heading 6</h6>



**HTML tags are classified :**

1. **Paired Tags or Container tags:** Paired tags have both opening and closing tags, and they enclose content between them. For example, **<p>** and **</p>** in the above example.
2. **Self-Closing Tags or Empty tags or Void tags:** Self-closing tags do not have a separate closing tag, and they are used to insert elements that do not contain any content or do not require closing. For example, **<img>** for an image element, or **<br>** for a line break.

**Display behaviors**

**Block-Level Elements:**

* Block-level elements in HTML are elements that, by default, start on a new line and occupy the full width available to them. They create "blocks" of content on a webpage
* Block-level elements such as <div>, <p>, <h1> to <h6>, <ul>, <ol>, and <li> always starts on new line.

**Inline Elements:**

* Inline elements in HTML are elements that do not start on a new line and only occupy as much width as necessary. They flow within the text or other inline elements.
* Inline elements, such as <span>, <a>, <strong>, <em>, <img>, and <input>, do not start on a new line

**Attributes**

Attributes provide additional information about an element and are specified within the opening tag of the element. Attributes consist of a name and a value, separated by an equals sign (=), and are enclosed within the element's opening tag in quotation marks (single or double).

<tagName **attributeName="attributeValue"**>Content</tagName>

**Comments**

comments are used to insert notes or annotations within the code that are ignored by the web browser when rendering the webpage. They are helpful for documenting the purpose or functionality of certain sections of HTML code, making it easier for developers to understand and maintain the codebase.

It is also used for preventing the code execution

<!-- This is a comment -->

**Break tag**

Break tag in HTML is used to insert a single line break within the text content of an HTML document.

<p>This is the first line of text.**<br>**This is the second line of text.</p>

**Hr tag**  
The **<hr>** tag in HTML is used to create a horizontal rule, also known as a horizontal line or divider, within the content of an HTML document. It is a self-closing tag. When the browser renders the HTML, it displays a horizontal line across the width of its containing element, typically to visually separate sections of content.

**Pre tag**

The pre tag in HTML stands for "preformatted text". It is used to display text exactly as it appears in the HTML code, preserving both spaces and line breaks. This tag is particularly useful for displaying code snippets, ASCII art, or any text where maintaining the original formatting is important.

**<pre>** Text or code snippet here**</pre>**

**Heading Tags and Their Sizes**

HTML provides six levels of heading tags (<h1> to <h6>) with different default font sizes.

**Default Sizes:**

* <h1>: 32px
* <h2>: 1.5em (approximately 24px)
* <h3>: 1.17em (approximately 18.72px)
* <h4>: 1em (16px)
* <h5>: 0.83em (approximately 13.28px)
* <h6>: 0.67em (approximately 10.72px)

**Html list tags**

1. Unordered list
2. Ordered list
3. Description list

**Unordered List <ul>**:

* Creates a list of items without any specific order.
* Attributes:
  + **type**: Specifies the bullet style. Values can be "**disc**" (default), "**circle**", or "**square**".

    <ul type="circle">

      <li>Item 1</li>

      <li>Item 2</li>

    </ul>

**Ordered List <ol>**:

* Used to create a list of items In specific order
* Attributes:
  + **type**: Specifies the numbering style. Values can be "**1**" (default, decimal), "**A**" (uppercase letters), "**a**" (lowercase letters), "**I**" (uppercase Roman numerals), "**i**" (lowercase Roman numerals).
  + **start**: Specifies the starting value of the first item.
  + **reversed**: Reverses the numbering order.

    <ol type="I" start="10" reversed>

      <li>Item 1</li>

      <li>Item 2</li>

    </ol>

**Description List <dl>**:

* Creates a list of term-description pairs.
* Attributes:
  + None.

  <dl>

      <dt>Term 1</dt>

      <dd>Description 1</dd>

      <dt>Term 2</dt>

      <dd>Description 2</dd>

    </dl>

**Tables**

HTML tables are used to display data in a structured format with rows and columns. They are defined using the <table> tag.

**Basic Structure**

A basic HTML table consists of:

* <**table**>: The container for the table.
* <**tr**> (table row): Defines a row in the table.
* <**th**> (table header): Defines a header cell in the table.
* <**td**> (table data): Defines a standard cell in the table.

Ex:-

        <table>

<!--first row-->

            <tr>

              <th>Header 1</th>

              <th>Header 2</th>

            </tr>

<!—second row>

            <tr>

              <td>Data 1</td>

              <td>Data 2</td>

            </tr>

          </table>

**Attributes**

**border**

Defines the border width of the table and its cells.

<table border="1">

**cellpadding**

Defines the space between the cell content and its border.

<table cellpadding="10">

**cellspacing**

Defines the space between cells.

<table cellspacing="10">

**Width and height**

Defines the width and height of the table.

<table width="500" height="300">

**bgcolor**

Defines the background color of the table, rows, or cells.

<tr bgcolor="yellow">

**Table Sections**

**<thead>**

Groups the header content in a table.

**<tbody>**

Groups the body content in a table.

**<tfoot>**

Groups the footer content in a table.

    <table border="1">

        <thead>

          <tr>

            <th>Header 1</th>

            <th>Header 2</th>

          </tr>

        </thead>

        <tbody>

          <tr>

            <td>Data 1</td>

            <td>Data 2</td>

          </tr>

        </tbody>

        <tfoot>

          <tr>

            <td>Footer 1</td>

            <td>Footer 2</td>

          </tr>

        </tfoot>

      </table>

**Caption**

A <caption> tag is used to provide a title for the table.

<table>

    <caption>Employees Table</caption>

    ...

 </table>

**Summary**

* HTML tables are used to display data in a structured format.
* Basic structure includes <table>, <tr>, <th>, and <td>.
* Attributes like border, cellpadding, cellspacing, width, height, and bgcolor customize the table.
* <thead>, <tbody>, and <tfoot> help group different parts of the table.
* <caption> provides a title for the table.
* CSS can be used to style tables for better presentation.

**Table tags**

HTML tables are a fundamental component for displaying data on web pages.

1. **Basic Structure**: HTML tables are defined using the **<table>** element. Within the **<table>** element, you typically have rows defined by **<tr>** (table row) elements, and within each row, you have cells defined by **<td>** (table data) elements. Additionally, you can use **<th>** (table header) elements to define header cells in the first row or column.
2. **Attributes**: Common attributes for the **<table>** element include **border**, **width**, **cellspacing**, and **cellpadding**, among others. However, it's generally recommended to use CSS for styling rather than HTML attributes for presentation.
3. **Accessibility**: It's important to make tables accessible for users who rely on screen readers. This involves using appropriate markup, such as **<th>** for headers, and providing alternative text for complex tables using the **summary** attribute or **<caption>** element.
4. **Table Headers**: Use **<th>** elements for header cells rather than **<td>** elements. This helps screen readers and other assistive technologies identify header cells.
5. **Spanning Cells**: You can use the **rowspan** and **colspan** attributes to make a cell span multiple rows or columns, respectively. This can be useful for creating more complex layouts or merging cells for visual purposes.
6. **Styling**: While basic styling can be done using HTML attributes like **border**, **bgcolor**, etc., it's more common and flexible to use CSS for styling tables. This allows for greater control over appearance, such as borders, background colors, spacing, and responsiveness.
7. **Responsive Design**: When designing tables for responsive websites, consider using CSS techniques like media queries to adjust the table layout for different screen sizes. You can also use **overflow-x: auto** to enable horizontal scrolling on smaller screens.

**table features**

**table caption** - it is used to provide the title for a table

**cellpadding** - it is used to create the space around the content in a cell

**cellspacing** - it is used to create the space between the cells

**bgcolor**- it is used to apply background color to the table

**width and height** - it is used to define a height and width for a table

**grouping** - we can group the cells by thead , tbody and tfoot

**spanning-** table cells can span into multiple rows and columns - also called as merging

<table>

  <thead>

    <tr>

      <th>Item</th>

      <th>Quantity</th>

      <th>Price</th>

    </tr>

  </thead>

  <tbody>

    <tr>

      <td>Apples</td>

      <td>10</td>

      <td>$20</td>

    </tr>

    <tr>

      <td>Oranges</td>

      <td>8</td>

      <td>$15</td>

    </tr>

    <tr>

      <td>Bananas</td>

      <td>15</td>

      <td>$10</td>

    </tr>

  </tbody>

  <tfoot>

    <tr>

      <td>Total</td>

      <td>33</td>

      <td>$45</td>

    </tr>

  </tfoot>

</table>

**Styling Tables**

CSS can be used to style tables for better presentation.

table {

    width: 50%;

    border-collapse: collapse;

  }

  th, td {

    border: 1px solid black;

    padding: 8px;

    text-align: left;

  }

  th {

    background-color: #f2f2f2;

    font-weight: bold;

    background-color: lightcoral;

  }

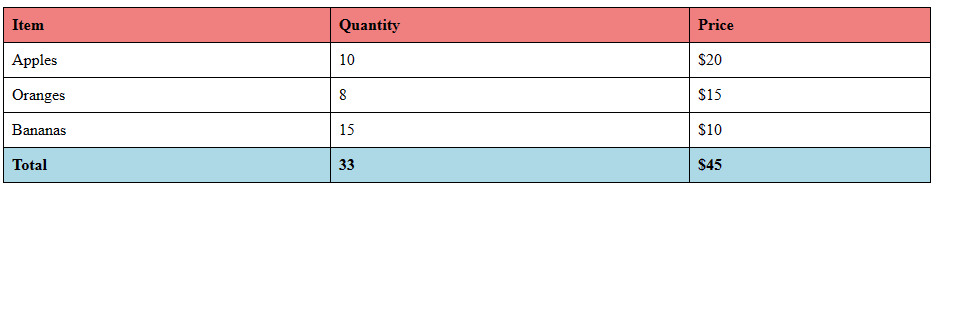
  tfoot {

    font-weight: bold;

    background-color: lightblue;

  }

**//output**

****

**Table tags with rowspan and collspan with cellpadding and cell spacing**

<table cellpadding="10" cellspacing="5">

  <tr>

    <td colspan="2">occupies two columns</td>

    <td rowspan="3">occupies three rows</td>

  </tr>

  <tr>

    <td rowspan="2">occupies two rows</td>

    <td>Data</td>

  </tr>

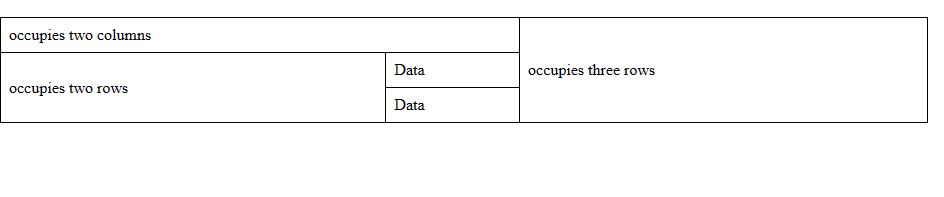
  <tr>

    <td>Data</td>

  </tr>

</table>

//output



1. The first row (**<tr>**) has three cells (**<td>**):
   * The first cell spans two columns (**colspan="2"**) and contains the text "occupies two columns".
   * The second cell contains the text "occupies three rows" and spans three rows (**rowspan="3"**).
   * The third cell is empty in this row.
2. The second row (**<tr>**) has two cells:
   * The first cell spans two rows (**rowspan="2"**) and contains the text "occupies two rows".
   * The second cell contains the text "Data".
3. The third row (**<tr>**) has one cell:
   * The cell contains the text "Data".

Each cell's content and spanning behavior are described using the **colspan** and **rowspan** attributes. Additionally, the table has **cellpadding** set to 10 pixels and **cellspacing** set to 5 pixels

**Marquee tag**

**<marquee>** tag in HTML is used to create a scrolling or moving text or image effect within a web page. It was commonly used in the past to create attention-grabbing animations, such as scrolling text banners or images.

    <marquee behavior="alternate" direction="left">Text or image content here</marquee>

* **behavior**: Specifies the behavior of the scrolling content. Possible values include:
  + **scroll**: Content continuously scrolls across the screen.
  + **slide**: Content slides across the screen once and then stops.
  + **alternate**: Content alternates between scrolling back and forth.
* **direction**: Specifies the direction of the scrolling content. Possible values include:
  + **left**: Content scrolls from right to left.
  + **right**: Content scrolls from left to right.
  + **up**: Content scrolls from bottom to top.
  + **down**: Content scrolls from top to bottom.

It has some attributes

**scrollamount**

* Defines the speed of the scrolling, in pixels. Higher values result in faster scrolling

**scrolldelay**

* Sets the delay between each scroll movement, in milliseconds. Lower values result in smoother scrolling.

**loop**

* Specifies the number of times the marquee should loop. The value can be a number or infinite.

**Anchor tags**

Anchortags, are used in HTML to create hyperlinks

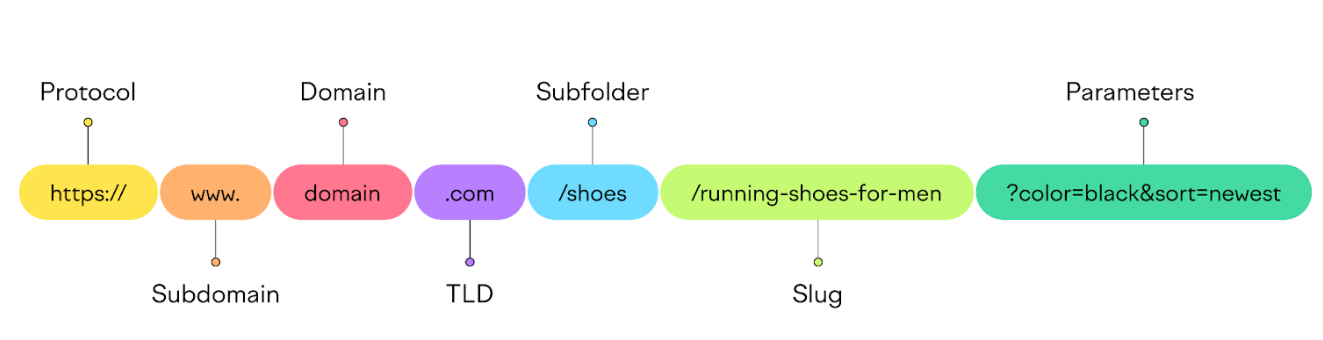
1. **Basic Structure**: An anchor tag is written as **<a></a>**. It has an opening tag **<a>** and a closing tag **</a>**. The content that goes between these tags becomes the clickable part of the link.
2. **Href Attribute**: The most important attribute of an anchor tag is **href**, which stands for "hypertext reference". It specifies the URL of the page the link goes to.

    <a href="https://www.example.com">Linked text</a>

1. **Link Text**: The text between the opening and closing **<a>** tags is what the user sees and clicks on. This is called the link text or anchor text.
2. **Target Attribute**: The **target** attribute specifies where to open the linked document. It can take values like **\_blank** (opens the linked document in a new window or tab) or **\_self** (opens the linked document in the same frame or tab).

    <a href="https://www.example.com**" target="\_blank"**>Click here</a>

**Structure of url**

****

**Protocol**

protocol is a set of rules and standards that define how data is transmitted and received over a network. Protocols ensure that devices on a network can communicate effectively, reliably, and securely. They cover various aspects, including data format, signaling, authentication, and error detection.

**HTTP and HTTPS**

**HTTP (Hypertext Transfer Protocol)**

* **Purpose**: The foundation of data communication on the web.
* **Function**: Defines how messages are formatted and transmitted, and how web servers and browsers should respond to various commands.
* **Port**: Typically uses port 80.
* **Stateless Protocol**: Each request from a client to a server is independent.

**HTTPS (Hypertext Transfer Protocol Secure)**

* **Purpose**: Secure version of HTTP.
* **Function**: Uses encryption to secure data between the client and server, providing confidentiality and integrity.
* **Port**: Typically uses port 443.
* **Encryption**: Uses SSL/TLS protocols for encryption.

Consider the domain name: www.example.com

* **www**: Subdomain (optional)
* **example**: Second-Level Domain (SLD)
* **.com**: Top-Level Domain (TLD)

**Subdomain**

A subdomain is an additional part of your main domain name. It helps organize and navigate to different sections of your website. Subdomains can be used to create unique URLs for different content areas, such as a blog, a store, or support pages, without needing to purchase additional domain names.

blog.example.com

shop.example.com

forum.example.com

**Second-Level Domain (SLD)**:

* The part directly to the left of the TLD.
* Chosen by the domain owner.
* Example: example in www.example.com

**Domain**

A domain name is a unique address used to identify a website on the internet.

**Domain Name System (DNS)**

The Domain Name System (DNS) is a hierarchical system that translates human-readable domain names into IP addresses. It allows users to access websites using easy-to-remember domain names instead of numerical IP addresses.

**TLD**

TLDs are a fundamental component of the domain name system, playing a crucial role in organizing and identifying websites on the internet. They help in categorizing domains, establishing credibility, and supporting branding efforts. Understanding TLDs is essential for anyone involved in web development, SEO, or digital marketing.

**Popular TLDs and Their Uses**

1. **.com**: Commercial entities, the most popular and widely recognized TLD.
2. **.org**: Non-profit organizations, open to anyone.
3. **.net**: Originally for network services, now open to anyone.
4. **.edu**: Educational institutions, restricted to accredited entities.
5. **.gov**: U.S. government entities, restricted.
6. **.in**: indian websites

**Path**

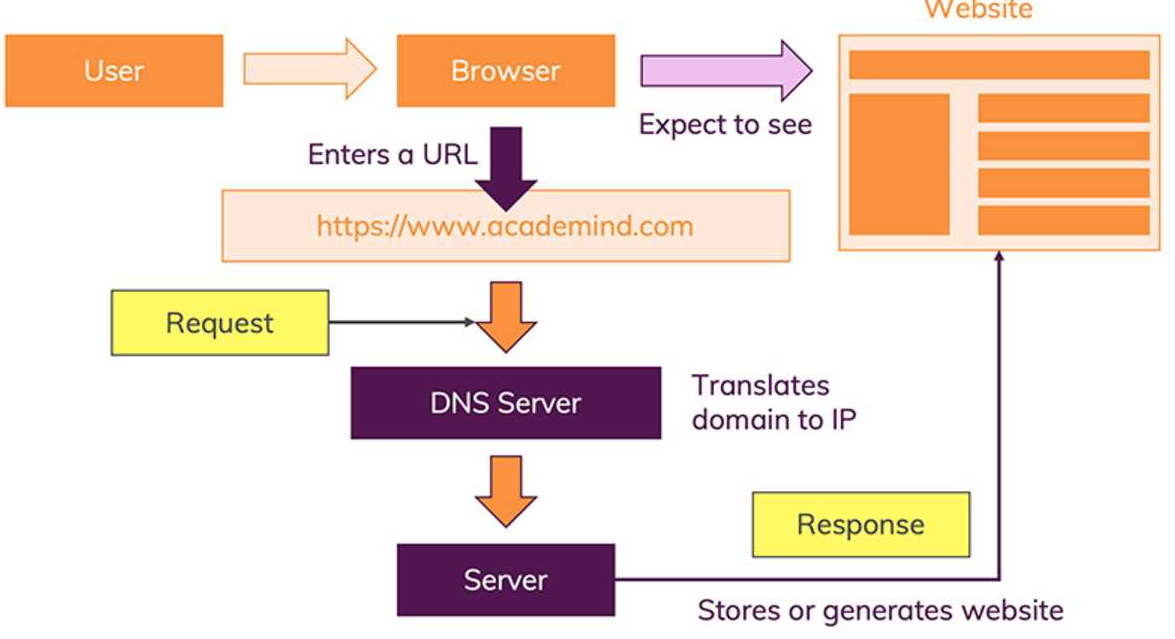
path specifies the location of a resource on the web server. It follows the domain name and precedes any query parameters or fragment identifiers.

<https://www.example.com/folder/subfolder/file.html>

scheme: https

domain: www.example.com

path: /folder/subfolder/file.html



**Absolute Paths**

An absolute path specifies the complete URL to a resource. This means it includes the protocol (such as http:// or https://), the domain name, and the full path to the resource.

**Example:**

<img src="https://www.example.com/images/logo.png" alt="Logo">

In this example, the browser will go to the specified URL and fetch the image, regardless of where the HTML document is located.

**Advantages of Absolute Paths:**

1. **Consistency**: The resource will be fetched from the exact same location, no matter where the HTML document is.
2. **External Resources**: Useful for linking to resources hosted on different servers.

**Disadvantages of Absolute Paths:**

1. **Dependency on Domain**: If the domain changes, all absolute paths need to be updated.
2. **Longer URLs**: Absolute paths are usually longer and can make the code harder to read.

**Relative Paths**

A relative path specifies the location of a resource in relation to the current document. Relative paths are shorter and more flexible than absolute paths.

**Types of Relative Paths:**

1. **Same Directory**:

<img src="logo.png" alt="Logo">

This refers to an image located in the same directory as the HTML file.

1. **Subdirectory**:

<img src="images/logo.png" alt="Logo">

This refers to an image located in the images subdirectory of the current directory.

1. **Parent Directory**:

<img src="../images/logo.png" alt="Logo">

The .. indicates moving up one level to the parent directory.

1. **Root Directory**:

<img src="/images/logo.png" alt="Logo">

The leading / refers to the root directory of the website

### Advantages of Relative Paths:

1. **Flexibility**: Easier to move entire directories or projects without changing paths.
2. **Shorter URLs**: More concise and easier to manage.

### Disadvantages of Relative Paths:

1. **Context Dependent**: Paths can break if the directory structure changes.
2. **Less Clear**: Can be harder to understand where the resource is located, especially in deeply nested directories.

**WAYS TO ADD PATH TO ANCHOR TAGS**

1. **ABSOLUTE PATH:**

An absolute URL contains the complete address of the resource, including the protocol (such as "http://" or "https://") and the domain name.

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

1. **RELATIVE PATH:**

A relative URL specifies the path to the linked resource relative to the current directory. Relative URLs do not include the domain name or protocol.

    <a href="about.html">About Us</a>

* To go out from a file we use “./”
* To go out from a folder we use ”../”

**SEMANTIC AND NON-SEMANTIC**

**Semantic elements**

Semantic elements are tags that convey meaning about the content to the browser and developer.By using semantic elements appropriately, web developers can create documents that are more accessible, search engine-friendly, and maintainable.

list of semantic HTML tags:

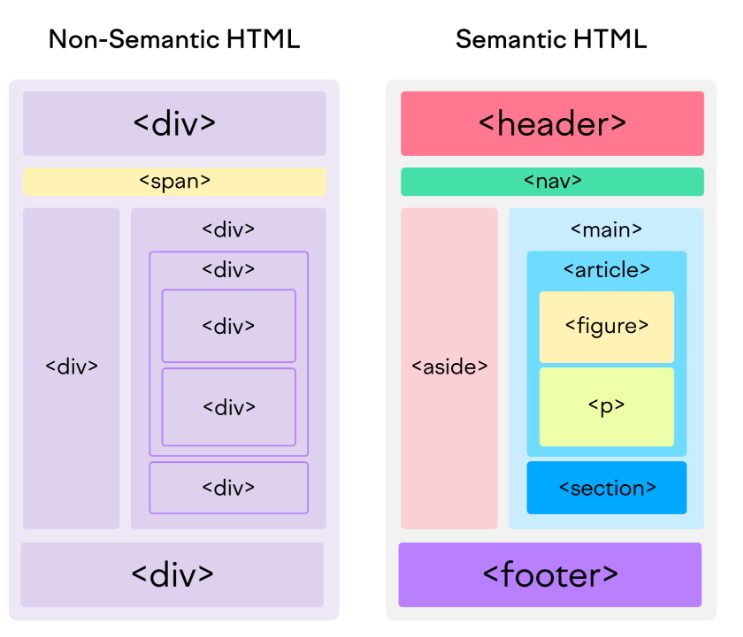
<header>, <nav>, <main>, <article>, <section>, <aside>, <footer>, <figure>, <figcaption>, <time>, <address>, <blockquote>, <cite>, <abbr>, <mark>, <audio>, <video>

**Non semantic elements**

Non-semantic elements in HTML are HTML tags that do not inherently convey any specific meaning about the content they contain.

list of non-semantic HTML tags:

<div>, <span>, <table>, <tr>, <td>, <th>, <ul>, <ol>, <li>, <dl>, <dt>, <dd>, <form>, <input>, <button>, <textarea>, <select>, <option>, <label>, <iframe>, <iframe>, <script>, <style>, <link>



**Html bookmarking**

What are HTML Bookmarks?

HTML bookmarks (also known as anchor links or internal links) allow users to navigate to specific sections within the same web page.

**1) Identify the Target Section:** Use the id attribute to create a target for the bookmark.

<section **id="section1"**>

<h2>Section 1</h2>

<p>This is the content for section 1.</p>

</section>

**2) Create a Link to the Bookmark:** Use the href attribute with the # symbol followed by the id of the target section.

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

**FORMATTING TAGS**

It is used to define the visual presentation of the web content.

Certainly! Here are some notes about formatting tags in HTML:

1. **Use of <b> and <i> Tags**: **<b>** and **<i>** tags were traditionally used for bold and italic formatting, respectively. However, they are considered presentational and semantically incorrect. Instead, prefer **<strong>** for importance and **<em>** for emphasis. If you need bold or italic text for styling purposes only, use CSS (**font-weight** and **font-style** properties) instead.
2. **<strong> and <em> Tags**: **<strong>** and **<em>** tags are used to indicate semantic importance and emphasis, respectively. They provide meaning to the text rather than just changing its appearance. Screen readers and search engines interpret these tags differently from **<b>** and **<i>**.
3. **<u> Tag**: The **<u>** tag is used to underline text. However, it's generally not recommended because underlined text is commonly associated with hyperlinks. Instead, use CSS (**text-decoration: underline;**) to underline text for styling purposes.
4. **<del></del>** Indicates deleted text, typically rendered with a strikethrough
5. **<mark></mark>** Highlights text, often used to mark text as important
6. **<sup> and <sub> Tags**: **<sup>** and **<sub>** tags are used to display superscript and subscript text, respectively. They are useful for mathematical equations, footnotes, and chemical formulas.
7. **<abbr title="World Health Organization">** tags are used to define abbreviations, respectively. They can improve accessibility by providing the full expansion of the abbreviation when hovered over.
8. **<blockquote> Tag**: Defines a section that is quoted from another source
9. **<q>**  it is used to insert quotations
10. **<cite> -** Defines the title of a work, such as a book, article, or song
11. **<address> -** Provides contact information, usually rendered in italics.
12. **<code> Tag**: The **<code>** tag is used to represent a fragment of computer code. It's typically rendered in a monospaced font and often styled with a background color or border to distinguish it from regular text.
13. **<pre> Tag**: The **<pre>** tag is used to preserve whitespace and line breaks within its content. It's commonly used for displaying code snippets or other preformatted text.
14. **<hr> Tag**: The **<hr>** tag is used to create a horizontal line or thematic break in the content. It's often used to separate sections of a webpage.
15. **<br> Tag**: The **<br>** tag is used to create a line break within a paragraph. It should be used sparingly, and CSS (**margin** and **padding**) should be preferred for controlling spacing between elements.

**VS Code - Integrated Development Environment (IDE)**

**Overview**

VS Code is an open-source code editor developed by Microsoft, primarily used for web development. It provides a range of resources and tools to assist in building applications with a graphical user interface (GUI).

**Guide to vs code**

**Download VS Code:** Visit the official Visual Studio Code website at <https://code.visualstudio.com/> and download the install

**Key Features**

* **Platform Support**: macOS, Linux, and Windows operating systems.
* **Development Capabilities**: Write and debug code for various applications.
* **Open Source**: Free to use and modify.

**Other Examples of IDEs**

* Sublime Text
* Notepad++
* Eclipse

**VS**

* **Themes Code Functionality**: Customize the appearance of the editor.
* **Open Folder**: Open an entire project folder.
* **Open File**: Quickly access and edit individual files.

**Navigate to your project folder**:Open the File Explorer (Windows) the folder containing your project files

1. Right-click on the file select "Open with vs Code". This option should appear if Visual Studio Code is installed and configured correctly. If you don't see this option, you can open VS Code manually and then use the File > Open Folder option to navigate to your project folder.
2. **Using Command Line (Optional)**: Another option is to open Visual Studio Code from the command line (Terminal or Command Prompt). Simply navigate to your project folder using the **cd** command and then type **code .** This command opens VS Code in the current directory (**.**).
3. **Using code editor**

**Productivity and Usability**

VS Code offers a variety of shortcuts and commands to enhance productivity and streamline the development process. Below are some essential shortcuts for basic operations, editing, navigation, and window management.

**Shortcuts**

**Basic Operations**

* **Ctrl + S:** Save the current file.
* **Ctrl + C:** Copy the selected text or item.
* **Ctrl + V:** Paste the copied content.
* **Ctrl + X:** Cut the selected text or item.
* **Ctrl + Z:** Undo the last action.
* **Ctrl + Y:** Redo the previously undone action**.**
* **Ctrl + A:** Select all content in the current file.
* **Ctrl + Plus (+):** Zoom in the editor view.
* **Ctrl + Minus (-):** Zoom out the editor view.

**Editing**

* C**trl + Shift + Alt + Down:** Add cursors to multiple lines for simultaneous editing.
* **Alt + Up Arrow:** Move the selected line or block of code one line up.
* **Alt + Down Arrow:** Move the selected line or block of code one line down.

**Navigation and Window Management**

* **Ctrl + Shift + P**: Open Command Palette
* **Ctrl + Shift + N**: New window
* **Ctrl + Shift + W**: Close window

**Additional Tips**

* **Extensions**: Enhance VS Code’s functionality with a wide range of extensions available in the Visual Studio Code Marketplace.
* **Integrated Terminal**: Access the terminal within VS Code for command-line operations without leaving the editor.
* **Version Control**: Integrated support for Git allows for seamless version control and collaboration.
* **IntelliSense**: Offers smart code completions based on variable types, function definitions, and imported modules.
* **Debugging**: Powerful debugging capabilities for various programming languages, with breakpoints, call stacks, and an interactive console.

**HTML EmmetTop of Form**

**Bottom of Form**

**Overview**

Emmet is a powerful toolkit for web developers that allows for rapid HTML and CSS coding by using shorthand abbreviations. These abbreviations are then expanded into complete code blocks, significantly speeding up the coding process and reducing errors. Emmet is widely supported in various code editors, including VS Code, Sublime Text, and Atom.

**Emmet abbreviations**

1) **!** – Expands to the complete html5 template

2) **tagname** – Expand the complete the tag

3) **>** - Used to create child elements within a parent element

4) **+** - Used to create sibling elements

5) **^** - Moves up one level in the hierarchy, allowing you to add sibling elements to parent

elements.

6) **\*** - Create multiple elements

7) {}- for content

8) []- for attributes

**Extensions needs to be installed**

1. Live server
2. Live preview
3. Prettier
4. Emmet live
5. Material theme icons
6. ESLint

**Image tag in html**

1. **Purpose:** The <img> tag is used to embed images into an HTML document.
2. **Attributes:**
   * src: Specifies the URL or path to the image file. This attribute is required.
   * alt: Provides alternative text for the image, which is displayed if the image cannot be loaded or by screen readers for accessibility and used to describe images to visually impaired users.This attribute is recommended for all <img> tags.
   * width: Specifies the width of the image in pixels or as a percentage of the containing element.
   * height: Specifies the height of the image in pixels or as a percentage of the containing element.
   * title: Adds a tooltip that appears when the user hovers over the image.

  <img src="image.jpg" alt="Description" width="200" height="150" title="image" >

To add an anchor link to an image in HTML, you can wrap the **<img>** tag within an **<a>** tag.

    <a href="link-url">

      <img src="image.jpg" alt="Description" />

    </a>

How to download image using anchor tag

    <a href="path/to/your/image.jpg" download="image.jpg">

      <img src="path/to/your/image.jpg" alt="Description" />

    </a>

How to add a telephone link using anchor tag

    <a href="tel:+1234567890">Call us at +1 (234) 567-890</a>

How to add email link using anchor tag

    <a href="mailto:developer@gmail.com">Send us an email</a>

How to add whatsap using anchor tag

  <a href="https://wa.me/1234567890" target="\_blank">Chat with us on WhatsApp</a>

Including a Pre-filled Message

# https://wa.me/PHONE\_NUMBER?text=YOUR\_MESSAGE

**Multi media tags**   
HTML provides several multimedia elements to embed different types of media content within web pages.

Note:- Media which is attached to the multimedia tags must be in your source

**Video tag**

    <video width="320" height="240" controls autoplay muted>

      <source src="mov\_bbb.mp4" />

    </video>

1. **HTML Video Element**:
   * The **<video>** element is used to embed video content within an HTML document.
2. **Attributes**:
   * **width="320"**: Sets the width of the video player to 320 pixels.
   * **height="240"**: Sets the height of the video player to 240 pixels.
   * **controls**: Enables playback controls such as play, pause, volume, etc.
   * **autoplay**: Specifies that the video should start playing automatically when the page loads.
   * **muted**: Sets the video to be muted by default, meaning no sound will be played.
   * **loop**: video will loop continously
3. **Source File**:
   * **<source src="mov\_bbb.mp4" />**: Specifies the source of the video file. In this case, "mov\_bbb.mp4" is the video file to be played.

4. **Fallback Content**:

* + The text or content within the **<video>** tags serves as fallback content. This content is displayed in browsers that do not support the **<video>** element or cannot play the specified video.

**Audio tag**

    <audio controls autoplay muted>

      <source src="audio\_file.mp3">

      Your browser does not support the audio element.

    </audio>

1. **HTML Audio Element**:
   * The **<audio>** element is used to embed audio content within an HTML document.
2. **Attributes**:
   * **controls**: Enables playback controls such as play, pause, volume, etc.
   * **autoplay**: Specifies that the audio should start playing automatically when the page loads.
   * **muted**: Sets the audio to be muted by default, meaning no sound will be played.
   * **loop :** audio will loop continously
3. **Source File**:
   * **<source src="audio\_file.mp3" />**: Specifies the source of the audio file. In this case, "audio\_file.mp3" is the audio file to be played.
4. **Fallback Content**:
   * The text or content within the **<audio>** tags serves as fallback content. This content is displayed in browsers that do not support the **<audio>** element or cannot play the specified audio.

**Details and Summary tag**

Details and summary tags are used together to create interactive collapsible sections of content block(accordion).

* The <**details**> tag is a container for content that can be expanded or collapsed by the user. When collapsed, only the content inside the <**summary**> tag (if provided) is visible. When expanded, all the content inside the tag is shown.
* **<summary>** tag: This tag defines a visible heading <details> tag for the **<details>** element. It is the text or element that the user clicks on to toggle the visibility of the content wrapped by the **<details>** tag.

    <details>

      <summary>Click to expand</summary>

      <p>

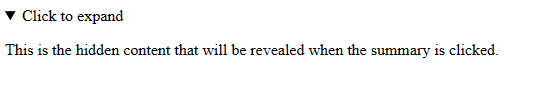
        This is the hidden content that will be revealed when the summary is

        clicked.

      </p>

    </details>

//ouput



**Iframes**  
iframe element is used to embed another HTML document within the current HTML document. It allows you to display content from another source, such as a different website or a different page within the same website, within a designated area of your webpage.

  <iframe src="https://en.wikipedia.org/wiki/HTML" width="600" height="400"></iframe>

How to display the external website or document in a iframe with anchor tag

    <!-- Inside the iframe -->

    <iframe width="300" height="300" name="myiframe"></iframe>

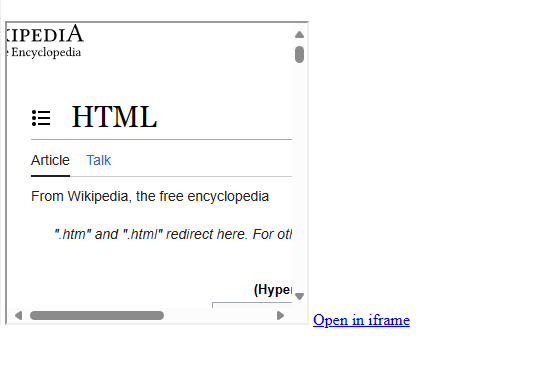
    <!-- Outside the iframe -->

    <a href="https://en.wikipedia.org/wiki/HTML" target="myiframe"

      >Open in iframe</a

    >

Above example sets up an **iframe with the name** **"myiframe"** and a hyperlink outside the iframe and **hyperlink target set to the “myiframe”** which targets the iframe When the hyperlink is clicked, the linked Wikipedia page about HTML will open within the iframe. This setup allows for displaying external content within a specific frame on a webpage.

****

**Bi-Directional Override**

The **<bdo>** tag in HTML stands for "Bi-Directional Override." It is used to override the text directionality of its content, allowing you to specify whether the text should be displayed from left-to-right (LTR) or right-to-left (RTL), regardless of the surrounding text directionality.

<p><bdo dir="rtl">Text will be displayed in reverse direction</bdo></p>

//output



**Meta tags:-**

Meta tags provide metadata about the HTML document. They are placed within the **<head>** element of an HTML document and are not displayed on the web page itself. Meta tags are used to provide information such as character encoding, viewport settings, keywords and description for search engines,

Examples:

1) Below example deals with the viewport responsiveness of browser by taking browser viewport widths

# <meta name="viewport" content="width=device-width, initial-scale=1.0">

2) Provides a brief description of the page's content to the search engine

# <meta name="description" content="A brief description of the page">

# <meta name="keywords" content="courses, placement, keyword3">

the page after a specified time interval.

# <meta http-equiv="refresh" content="5">

**Favicon:-**

HTML Favicon is a small icon that represents a website and appears in the browser’s tab or bookmark bar. It is defined in [HTML](https://www.geeksforgeeks.org/html-introduction/) using the [<link> tag](https://www.geeksforgeeks.org/html-link-tag/) with the rel attribute set to “icon.”

# <link rel="icon" type="image/x-icon" href="url">

**Forms**

HTML forms are used to collect user input. They contain form elements like text fields, checkboxes, radio buttons, and submit buttons, allowing users to input and submit data to a server for processing.

1. **Structure**: Forms are created using the **<form>** element. This element wraps all the form controls (like input fields, buttons, checkboxes, etc.) that allow users to enter data.

  <form action="/submit\_form.php" method="post">

    <label for="username">Username:</label><br>

    <input type="text" id="username" name="username" required>

  </form>

**Form Attributes**

* **action**: Specifies the URL where the form data will be sent for processing.
* **method**: Specifies the HTTP method to be used when submitting the form. Common values are GET and POST.

1. **Input Types**: The <input> element is incredibly versatile and can be used for various input types like text, password, checkbox, radio button, date, email, etc. The type attribute specifies the type of input.

  <label for="text">Text:</label>

  <input type="text" id="text" name="text"><br><br>

  <label for="email">Email:</label>

  <input type="email" id="email" name="email" required><br><br>

  <label for="password">Password:</label>

  <input type="password" id="password" name="password"><br><br>

  <label for="datetime">Date time:</label>

  <input type="datetime-local" id="datetime" name="datetime"><br><br>

  <label for="birthdate">Birthdate:</label>

  <input type="date" id="birthdate" name="birthdate"><br><br>

  <label for="quantity">Quantity:</label>

  <input type="number" id="quantity" name="quantity" min="1" max="10"><br><br>

  <label for="file">Upload File:</label>

  <input type="file" id="file" name="file"><br><br>

  <input type="checkbox" id="agree" name="agree" required>

  <label for="agree">I agree to the terms and conditions</label><br><br>

  <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>

  <label for="color">color</label>

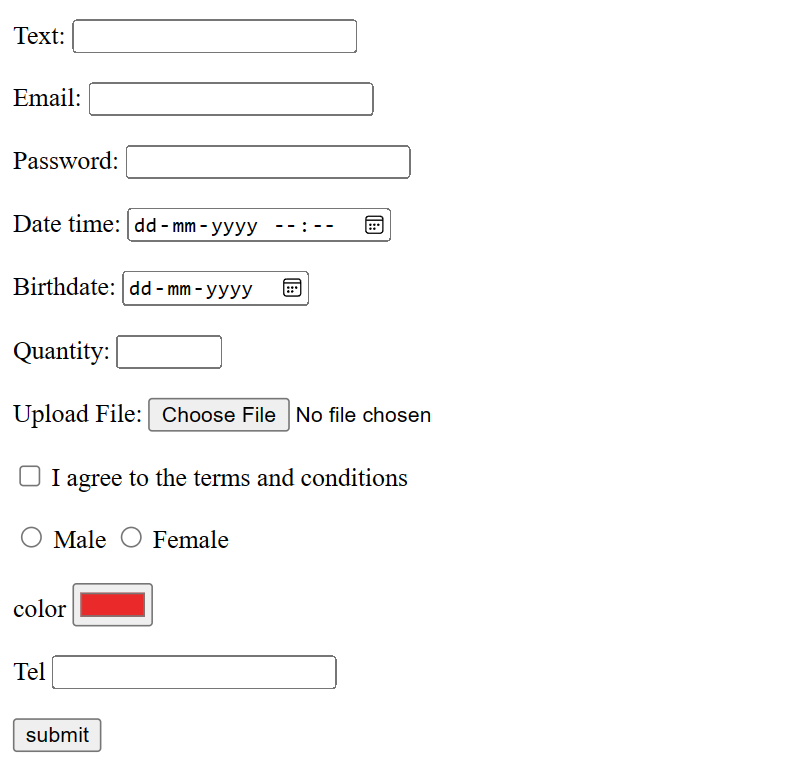
  <input type="color" id="color" name="color"><br><br>

  <label for="tel">Tel</label>

  <input type="tel" id="tel" name="tel"><br><br>

  <input type="button" value="submit">

1. **Form Controls**: Form controls are elements that allow users to input data. Common form controls include <input>, <textarea>, <select>, <button>, etc.



1. **Labels**: Labels (**<**label**>**) are used to describe the purpose of form controls. They improve accessibility and usability by providing a text label for each form control.

<label for="uname ">Username </label>

1. **Attributes :** Attributes are built in methods which controls behavior of tags.
2. **Accessibility**: It's important to make forms accessible to all users, including those using screen readers. Proper use of labels, semantic HTML, and attributes can improve accessibility.
3. **Styling**: Forms can be styled using CSS to match the design of the website or application. CSS frameworks like Bootstrap provide pre-styled form components that can be easily customized.

**Attributes**

* **Required:**

The required attribute is used in HTML form input elements to specify that a particular field must be filled out before the form can be submitted.

# <input type="text" id="username" name="username" **required**>

* **Disabled:**

The disabled attribute is used in HTML form elements to specify that the input field, button, or other interactive element should be disabled. When an element is disabled, it cannot be clicked, focused, or modified by the user. This attribute is particularly useful for indicating that a form control is inactive or unavailable under certain conditions.

# <input type="password" id="password" name="password" **disabled**>

* **Placeholder:**

The placeholder attribute is used in HTML form input elements to provide a hint or example text to users about what type of information is expected in the field. It's a non-essential attribute, providing guidance or context within the input field itself, usually disappearing when the user begins typing.

  <!-- Text Input Field with Placeholder -->

# <input type="text" id="username" name="username" **placeholder="Enter your username"**>

* **Readonly**  
  The **readonly** attribute in HTML is used to make an input field or textarea read-only, meaning the user can see the content but cannot modify it. This attribute is particularly useful when you want to display data that shouldn't be altered by the user, such as pre-filled information or computed values.

# <input type="text" name="readonlyInput" value="Pre-filled Data" **readonly**>

**maxlength**

* **Description**: Limits the number of characters allowed in an input field.

**min and max**

* **Description**: Set the minimum and maximum values for input fields like number, range, and date.

**step**

* **Description**: Defines the interval between valid input values for fields such as number and range.

**autofocus**

* **Description**: Automatically focuses on the specified input field when the page loads.

**multiple**

* **Description**: Allows multiple values to be selected in input fields of type file or email.

**Pattern:**  
The pattern attribute in HTML is used to specify a regular expression that an input field's value must match for the form to be submitted successfully. It's commonly used with text input fields to enforce specific formats for data input, such as phone numbers, ZIP codes, email addresses, etc

  <form>

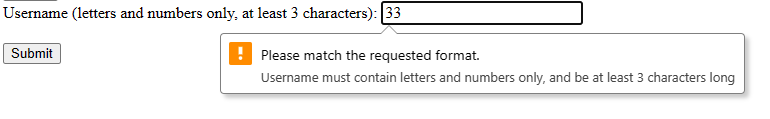
    <label for="username">Username (letters and numbers only, at least 3 characters):</label>

    <input type="text" id="username" name="username" **pattern="[A-Za-z0-9]{3,}"** title="Username must contain letters and numbers only, and be at least 3 characters long" required><br><br>

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

</form>

It uses the **pattern** attribute with a regular expression **[A-Za-z0-9]{3,}**. This pattern specifies that the username must contain letters and numbers only and be at least 3 characters long.



1. **Single Lowercase Alphabet**: [a-z]
   * Accepts only one lowercase alphabet.
2. **Single Uppercase Alphabet**: [A-Z]
   * Accepts only one uppercase alphabet.
3. **Multiple Occurrences**: +
   * Indicates that the preceding element can occur one or more times.
4. **Single Digit**: [0-9]
   * Accepts only one digit.
5. **Multiple Digits**: [0-9]+
   * Accepts multiple digits.
6. **Special Characters**: [@#$%&\*]+
   * Accepts one or more special characters from the set (@, #, $, %, &, \*).
7. **Alphanumeric Characters**: [a-zA-Z0-9]
   * Accepts any single character that is a lowercase letter, uppercase letter, or digit.
8. **Specific Length of Lowercase Alphabets**: [a-z]{3}
   * Accepts exactly three lowercase alphabets.
9. **Three Digits**: \d{3}
   * Accepts exactly three digits.

**Specific Patterns**

**Email Pattern**

* **Pattern**: [a-z0-9.\_]+@[a-z]+\.[a-z]{2,4}
* **Explanation**:
  + [a-z0-9.\_]+: One or more lowercase alphabets, digits, periods, or underscores.
  + @: The "@" symbol.
  + [a-z]+: One or more lowercase alphabets.
  + \.: A period.
  + [a-z]{2,4}: Two to four lowercase alphabets.

**Date Pattern**

* **Pattern**: [0-9]{4}-[0-9]{2}-[0-9]{2}
* **Explanation**:
  + [0-9]{4}: Four digits (year).
  + -: A hyphen.
  + [0-9]{2}: Two digits (month).
  + -: A hyphen.
  + [0-9]{2}: Two digits (day).

**Password Pattern**

* **Pattern**: (?=. \*[A-Z])(?=.\*[a-z])(?=.\*[!@#$%^&\*])(?=.\*[0-9])[A-Za-z!@#$%^&\*0-9]{8,}
* **Explanation**:
  + (?=.\*[A-Z]): At least one uppercase letter.
  + (?=.\*[a-z]): At least one lowercase letter.
  + (?=.\*[!@#$%^&\*]): At least one special character.
  + (?=.\*[0-9]): At least one digit.
  + [A-Za-z!@#$%^&\*0-9]{8,}: At least 8 characters long, consisting of uppercase, lowercase, special characters, and digits.

**Fieldset and legend**

The <fieldset> and <legend> elements are used together to group related form controls and provide a title or description for the group.

* **Fieldset**

The <fieldset> element is used to group related form controls together within a form. It's typically used to create sections or categories within a form, making it easier for users to understand the relationship between different input fields.

* **Legend**

The <legend> element is used to provide a title or caption for the <fieldset>. It typically appears as a heading above or beside the group of form controls enclosed by the <fieldset>.

<fieldset>

  <legend>Select Your Favorite Color</legend>

  <input type="radio" id="red" name="color" value="red">

  <label for="red">Red</label><br>

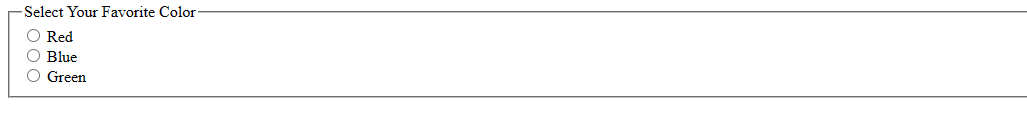
  <input type="radio" id="blue" name="color" value="blue">

  <label for="blue">Blue</label><br>

  <input type="radio" id="green" name="color" value="green">

  <label for="green">Green</label><br>

</fieldset>



**Text area**

The <textarea> element is used to create a multi-line text input field, allowing users to input multiple lines of text.

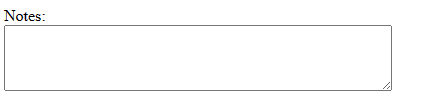
 <form>

    <label for="notes">Notes:</label><br>

    <textarea id="notes" name="notes" rows="4" cols="50"></textarea>

 </form>

* **<textarea>** is the tag used to create a text area.
* **id="notes"** and **name="notes"** are attributes used for identification and form submission purposes.
* **rows** and **cols** attributes determine the size of the text area in terms of rows and columns, respectively.



**Select**

The <select> element is used to create a drop-down list of options, allowing users to select one or multiple items from a list.

    <form>

      <label for="cars">Choose a car:</label><br />

      <select id="cars" name="cars">

        <option value="volvo">Volvo</option>

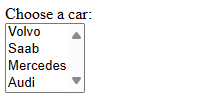
        <option value="saab">Saab</option>

        <option value="mercedes">Mercedes</option>

        <option value="audi">Audi</option>

      </select>

    </form>



* **<select>** is the tag used to create the drop-down list.
* **<option>** tags are used to define the options within the drop-down list.
* The **value** attribute of each **<option>** tag defines the value that will be submitted to the server when the form is submitted.
* The text between the opening and closing **<option>** tags represents the visible option text displayed to the user.

1. **Select and option attributes**.

* **multiple**: Allows users to select multiple options from the dropdown list.

## <select id="colors" name="colors" **multiple**>

* **disabled**: Disables the dropdown list, making it non-interactive.

## <select id="countries" name="countries" **disabled**>

* **selected**: Specifies that the option is pre-selected when the page loads.Example:

## <option value="volvo" **selected**>Volvo</option>

1. **Grouping in select dropdown**

you can group related **<option>** elements within a **<select>** dropdown list using the **<optgroup>** element. This allows you to organize and categorize options for better user experience.

    <select name="cars">

      <optgroup label="European Cars">

        <option value="volvo">Volvo</option>

        <option value="audi">Audi</option>

        <option value="bmw">BMW</option>

      </optgroup>

      <optgroup label="Japanese Cars">

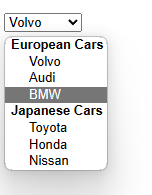
        <option value="toyota">Toyota</option>

        <option value="honda">Honda</option>

        <option value="nissan">Nissan</option>

      </optgroup>

    </select>



**Datalist**

the **<datalist>** element is used in conjunction with **<input>** elements to provide a list of predefined options for users to select from.

    <form>

      <label for="list">Choose a option:</label><br />

      <input type="text" list="options" id="list" name="optionvalue" />

      <datalist id="options" >

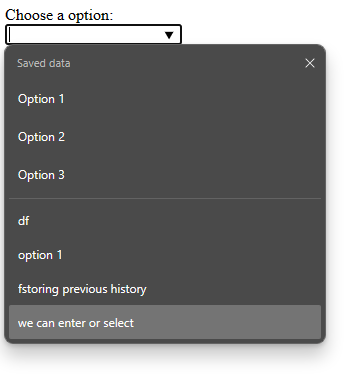
        <option value="Option 1"></option>

        <option value="Option 2"></option>

        <option value="Option 3"></option>

      </datalist>

    </form>



the **<datalist>** element contains a list of **<option>** elements, each specifying a value that the user can choose from. The **<input>** element with the **list** attribute set to the **id** of the **<datalist>** element provides a text field where the user can either type a value or choose one from the dropdown list provided by the **<datalist>.**

**Autofocus**

The **autofocus** attribute is used to specify that an input field, button, or other interactive element should automatically have focus when the page loads. This means that the cursor will be placed in that field or it will be ready for user input without the need for the user to click on it.

    <input type="text" autofocus>

**Method attribute in form**

The method attribute of the <form> element specifies the HTTP method used when submitting the form data to the server.

* **GET Method:**
* The **GET** method is used to request data from a specified resource.
* When a form is submitted using **GET**, the form data is appended to the URL in the form of a query string.
* GET requests can be bookmarked and cached by browsers, making them suitable for retrieving data that doesn't modify server state.

    <form action="/search" method="GET">

      <input type="text" name="query">

      <button type="submit">Search</button>

    </form>

//below value are appended to the url



* **POST Method:**
* The **POST** method is used to submit data to be processed to a specified resource.
* When a form is submitted using **POST**, the form data is sent in the body of the HTTP request.
* POST requests are not cached or bookmarked by browsers and are more secure for sensitive data as they don't expose parameters in the URL.
* Use **POST** when the form submission modifies server state, such as creating, updating, or deleting data on the server.

    <form action="/submit" method="POST">

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

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

      <button type="submit">Submit</button>

    </form>

//below values are not appended to the url



**Action attribute in form tag**

The **action** attribute in the **<form>** tag specifies the URL or endpoint to which the form data should be submitted when the form is submitted by the user. It essentially defines the destination where the form data will be sent for processing.

    <form action="/submit-form" method="POST">

      <!-- Form fields go here -->

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

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

      <button type="submit">Submit</button>

    </form>

* The **action** attribute is set to **"/submit-form"**, indicating that when the form is submitted, the data will be sent to the **/submit-form** endpoint on the server.
* The **method** attribute is set to **POST**, which means that the form data will be sent to the server using the HTTP POST method.

The **action** attribute can be an absolute or relative URL. If it's a relative URL, it's interpreted relative to the current page's URL. If the **action** attribute is omitted, the form data will be submitted to the current page's URL by default.

**Formaction attribute**

The formaction attribute is used to override the action attribute of a <form> element for a specific <button> or <input> element. It allows you to specify a different URL or endpoint where the form data should be sent when that particular button or input is clicked, without modifying the action attribute of the form itself.

    <form action="/default-action" method="POST">

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

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

      <!-- Submit button with formaction attribute -->

      <button type="submit" formaction="/custom-action">Submit</button>

    </form>

* The **<form>** element has an **action** attribute set to **"/default-action"**, specifying the default URL where the form data will be sent.
* The **<button>** element within the form has a **formaction** attribute set to **"/custom-action"**. When this button is clicked, the form data will be sent to **"/custom-action"** instead of the default action specified in the **action** attribute of the form.

    <form action="/default-action" method="post">

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

        <button type="submit" formaction="/special-action">Special

Submit</button>

        <button type="submit">Default Submit</button>

      </form>

clicking the "Special Submit" button will send the form data to /special-action, while clicking the "Default Submit" button will send the form data to /default-action.

**Key Differences**

1. **Scope**:
   * action is for the entire form.
   * formaction is for specific buttons within the form.
2. **Priority**:
   * formaction overrides the action attribute for the button it is applied to.
3. **Use Case**:
   * Use action when you want a single submission URL for the entire form.
   * Use formaction when you need different submission URLs for different buttons within the same form.

By using both attributes effectively, you can create more flexible and dynamic forms that can submit data to different endpoints based on user interaction.

**Html Entities**  
HTML entities are special codes used to represent characters that have special meanings in HTML, or characters that are not easily typed or displayed with a keyboard. They are particularly useful for displaying symbols, foreign characters, mathematical symbols, and reserved characters.

some common HTML entities along with their corresponding characters:

1. **&amp;** - represents the ampersand (&)
2. **&lt;** - represents the less-than symbol (<)
3. **&gt;** - represents the greater-than symbol (>)
4. **&quot;** - represents the double quotation mark (")
5. **&apos;** - represents the single quotation mark (')
6. **&nbsp;** - represents a non-breaking space
7. **&copy;** - represents the copyright symbol (©)
8. **&reg;** - represents the registered trademark symbol (®)
9. **&trade;** - represents the trademark symbol (™)

10.**&euro;** - represents the euro currency symbol (€)

**How to insert Font Awesome icons to the html**

**1.Include Font Awesome in document**:

Need to include the Font Awesome file in your HTML document. You can do this by adding the following line inside the <head> section of your HTML document:

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css">

**2. Choose an Icon**:

Visit the Font Awesome website to browse through the available icons. Once you find the icon you want to use, click on it to view its details.

**3.Copy the HTML Code**: Font Awesome provides HTML code snippets for each icon. Copy the HTML code snippet

<i class="fas fa-check"></i>

**4.Paste into HTML Document**:

<div><i class="fas fa-check" style="font-size: 24px; color: green;"></i><div/>

**How to insert Ion Icons to the html**

1. **Include Ionicons in your HTML document**: Ionicons can be included in your HTML document by linking to the Ionicons CSS file hosted on a CDN (Content Delivery Network). Add the following line within the **<head>** section of your HTML document:

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/ionicons/5.1.2/css/ionicons.min.css">

1. **Choose an icon and paste into the html document** Ionicons are implemented as a font icon set, so you can use them by applying the appropriate CSS classes to HTML elements. For example, to display the "heart" icon, you can use the following code:

<i class="icon ion-heart"></i>

**Figure and figcaption tags**

the **<figure>** and **<figcaption>** tags in HTML. These tags are used together to semantically represent self-contained content, such as images, diagrams, illustrations, code snippets, and more, along with a caption describing the content.

    <figure>

      <img src="image.png" alt="Example Image" width="100" height="100">

      <figcaption>This is a caption describing the image.</figcaption>

    </figure>

