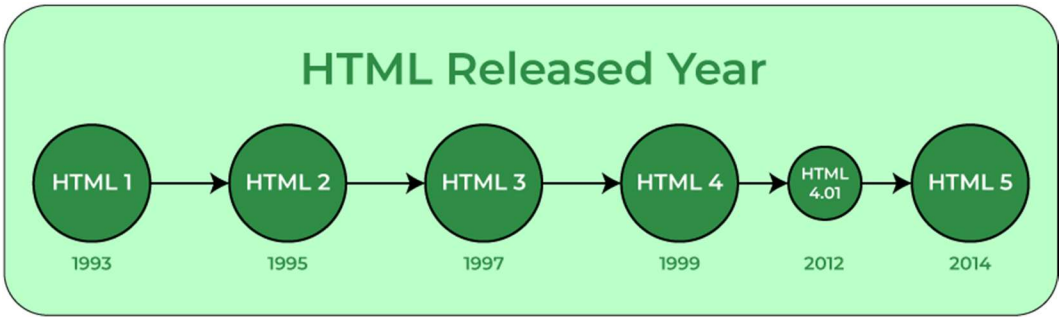
HTML Introduction

HTML, or Hyper Text Markup Language, is the standard markup language for creating web pages and web applications.

The first version of HTML was written by **Tim Berners-Lee** in **1993**.

It is used to define the structure and content of a web page, and is interpreted by a web browser to render the page into a visible or audible form.

There have been many different versions of HTML. The most widely used version throughout the 2000's was HTML 4.01, which became an official standard in December 1999.



The HTML uses a markup language which is called Tags.

Tags are enclosed within '<TagName>'

There are 6 types of tags, please check the below.

1. **Structural tags:** Describes the structure of the web page.
   Ex : <!DOCTYPE html>, <html>, <head>, and <body>
   a. *Html :*
   b. *Head :*
      i. ***Meta :*** Provides metadata about the HTML document, such as character encoding, author, description, etc.
      Example :

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">

ii. ***Base :*** Specifies a base URL/target for all relative URLs in a document.

Example :

```
<base href="Base_link”>
<body>
    <a href = “/”> base link here </a>
</body>
```

iii. ***Script :*** Used to mention the javascript inside it.

iv. ***Noscript :*** Defines alternative content to be displayed when JavaScript is not supported or disabled in the browser.

Example :

```
<noscript>
    <!-- Alternative content for users without JavaScript support -->
    <p>This website requires JavaScript to function properly.</p>
</noscript>
```

2. ***Formatting tags:*** Helps to format the content or helps to add the links to the content.
   Ex : <h1>, <p>, and <a>
3. ***List tags:*** Used to show content in an order.
   Ex : <ul> and
4. ***Image tags:*** Image tags are used to add images to the web page.
   Ex : <img>
5. ***Form tags:*** Form tags are used to build forms to the page.
   Ex : <form>, <input>, and <button>
6. ***Meta tags:*** Used provide information about an HTML document to search engines, browsers, and other web services.
   Ex : <meta>

The basic structure of HTML :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Shows the title of the web page</title>
</head>
<body>
  Shows the content of the web page
</body>
</html>
```

Types of Tags in HTML :

Single tags:

This tags will not have any closing tags they themselves are closing and opening tags

Ex :

<!DOCTYPE html> : Describe the document type
<meta> : Meta tags in HTML provide additional information about a web page to search engines and browsers.

Empty tags:

HTML empty tags are self-closing tags that do not need a closing tag because they do not have any content.

Ex :

: Insert a line break
<hr>: Insert a horizontal rule
<img>: Insert an image
<input>: Insert an input field
<link>: Insert a link to a CSS file or other resource

Paired tags :

Tags that have a start tag and an end tag.

Ex :

<p>, div

Semantic tags :

Tags that describe the meaning of the content they contain.

Ex :

<header>, <article>, <footer>, <table>

Non-semantic tags :

Tags that do not describe the meaning of the content they contain.

Ex :

, ,

Elements :

HTML elements are the building blocks of web pages, defined by start and end tags, and can contain content and attributes.

There are 3 types of elements in HTML.

1. *Inline*
2. *Block*
3. *Inline-Block*

Types of Elements :

1. Inline Elements :

Inline takes up only as much space as its content and does not start on a new line. Can be used to create links, images, and other small elements.

Ex :

<a>: Anchor link
: Span element
<img>: Image element
<input>: Input field
<button>: Button

2. Block- level Elements :

Block-level Starts on a new line and takes up the full width of its container. Can be used to create paragraphs, headings, and other structural elements of a web page.

Ex :

* : Division element*
<p>: Paragraph element
<h1> to <h6>: Heading elements
* and <ol>: List elements*
<table>: Table element

3. Inline- Block Elements :

Inline-block Takes up only as much space as its content and starts on a new line. Can be used to create buttons, images, and other elements that need to be displayed as boxes.

Ex :

<button>: Button
<img>: Image element
<input>: Input field

Note : We can use the download attribute when we click on the anchor tag.

```
<a href="./form.html" download="form.html">download now</a>
```

Attributes :

Every tag will have its own attributes. Using these attributes we can customize and recognize the tags.

Ex :

Class : used to identify an tag.
Id : It works similarly to class.
Height: can customize height of the tag.
Width : can customize the width of the tag.

There are two types of attributes :

1. Global Attributes
2. Pre defined Attributes

Global Attributes :

These attributes can be used by any tag. There are not limited to certain tags. You can use it in the head section to the body section.

- ***accesskey*** *- Specifies a keyboard shortcut for the element.*
  - *For windows we use alt + key*
  - *For Mac we use ctrl + key*
- ***class*** *- Specifies one or more class names for the element. Classes can be used to style the element with CSS.*
- ***contenteditable*** *- Specifies whether the element's content can be edited by the user.*
- ***dir*** *- Specifies the direction of the element's text. Can be either ltr (left-to-right) or rtl (right-to-left).*
- ***draggable*** *- Specifies whether the element can be dragged by the user.*
  - ***Example :***
    -
```
<p draggable="true">This is draggable content</p>
```

This is draggable content

This is draggable content

- ***dropzone*** - *Specifies whether the element can be used as a drop zone for draggable elements.*
- ***hidden*** - *Specifies whether the element is hidden from the user.*
- ***id*** - *Specifies a unique ID for the element. IDs can be used to style the element with CSS or to reference it with JavaScript.*
- ***lang*** - *Specifies the language of the element's content.*
- ***spellcheck*** - *Specifies whether the element's content should be spellchecked by the browser.*
- ***style*** - *Specifies inline CSS styles for the element.*
- ***tabindex*** - *Specifies the element's place in the tab order.*
- ***title*** - *Specifies a tooltip that should be displayed when the user hovers over the element.*

Text Tags :

In HTML we use the text tags to edit the content in our web page. There are different types of text tags that we use to edit the content on our page.

1. **Headings**:

   used to set headings at different levels in different sizes.

   i. *We have the heading tags in a range of h1 to h6.*
   ii. *The higher we go the smaller text we get.*

   Example :

```
<h1>heading h1</h1>
<h2>heading h2</h2>
<h3>heading h3</h3>
<h4>heading h4</h4>
<h5>heading h5</h5>
<h6>heading h6</h6>
```

Output :

heading h1

heading h2

heading h3

heading h4

heading h5

heading h6

2. **Paragraph**:

used to write the content as a paragraph

Example :

```
<p>This is a paragraph1</p>
<p>This is a paragraph2</p>
```

Output :

This is a paragraph1

This is a paragraph2

3. **Lists**:

used to describe the content in a list

a. Lists are two types :

i. ***Ordered List***

1. We surround the li tags with the ol

Example :

```
<ol>
    <li>list1</li>
    <li>list2</li>
    <li>list3</li>
  </ol>
```

Output :

1. list1
2. list2
3. list3

ii. Unordered List

1. We surround the li tags with the ul for unordered list

Example :

```
<ul>
    <li>list1</li>
    <li>list2</li>
    <li>list3</li>
  </ol>
```

Output :

- list1
- list2
- list3

iii. Definition List

1. In Definition, we have the different tags which are used to specify the content as heading and explanation.

Ex :

dl : used to declare the declaration list
dt : used to declare the definition term

dfn : used to declare the side heading for the definition

dd : used to declare the definition data

```
<dl>
  <dfn>
    definition 1
    <dt>definition term</dt>
    <dd>
      definition description
    </dd>
  </dfn>
</dl>
```

Output :

definition 1
definition term
definition description

4. Table:

Used to create tables in our web page.

a. To create tables we use these tags.

Ex :

Defines a table.

<tr>: Defines a table row.

<th>: Defines a table header cell.

<td>: Defines a table data cell.

1. Rowspan : used to mention how many rows it has to span
2. Colspan : Used to mention how many cols it has to span

<caption>: Defines a table caption.

← These are used to provide structure →

<thead>: Defines a table header section.

<tfoot>: Defines a table footer section.

<tbody>: Defines a table body section.

← These are used to provide structure →

<colgroup>: Defines a group of columns in a table.

<col>: Defines a column in a table.

Example :

```
<table border = "1">
  <caption>This is table heading</caption>
  <thead>
```

```
            <tr>
                <td colspan="3">
                    <center>
                        Student
                    </center>
                </td>
            </tr>
            <tr>
                <th>ID</th>
                <th>Name</th>
                <th>Age</th>
            </tr>
        </thead>
        <tbody>
          <tr>
            <td>
                1
            </td>
            <td>
                Luffy
            </td>
            <td>
                23
            </td>
          </tr>
          <tr>
            <td>
                2
            </td>
            <td>
                Billy
            </td>
            <td>
                24
            </td>
          </tr>
        </tbody>
        <tfoot>
            <tr>
                <td colspan="3">
```

```
                        This is footer
                    </td>
                </tr>
            </tfoot>
        </table>
```

Output :

This is table heading

Student		
ID	**Name**	**Age**
1	Luffy	23
2	Billy	24
This is footer		

5. **Graphic Tags :**

Used to add pictures to the web page.

a. We have two graphic tags that we can use on our webpage.
   i. Image
   ii. Figure
   a. Image : *An image tag is used to add a picture to the webpage.*
      a. To insert an image we use the image tag - <img src =" ">
   b. Figure : *A figure tag is used to add a picture along with the description.*
      a. We use the combination of the image tag and figure tag in order to add the description for the image.
      b. We have to place the image tag in <figure> tag and add the description tag <figcaption>.

6. **Link tags :**

Links in HTML are used to connect the two pages or resources to a webpage.

i. We can select or click on a link. When we click on it will take us to a new page.
ii. We select different options whether we need to open the link in the same page or a different page, using an attribute called target.
iii. To link the content in a page we need to use the anchor tag <a>.
iv. We use the href attribute in order to add the address or path.

7. **Multimedia Tags :**

Used to add multimedia data to our web page.

i. We have the following tags in order to add the multi-media data.
    1. **Audio** – add the audio to the web page.
    2. **Video** – add the video to the web page.

ii. We need to use the control tag to provide the playback options for both video and audio tags.
    1. ***src****: Specifies the URL of the audio/video file.*
    2. ***controls****: Adds audio/video controls (play, pause, volume, etc.).*
    3. ***autoplay****: Specifies that the audio/video should start playing automatically.*
    4. ***loop****: Specifies that the audio/video should restart when finished.*
    5. ***preload****: Specifies if and how the audio/video should be loaded when the page loads.*
    6. ***Poster*** *: Provides the poster before the video starts.*

Example *:*

```
    <video src="../multiMedia/5 Second Video_ Watch the Milky Way
Rise.mp4" controls poster="../../Lenskart/images/Spec1.webp" autoplay
loop></video>
```

Anchor tag : .

1. Internal Linking (Within Project):

Internal linking is used to create links within the same website or project. It helps users navigate between different pages or sections of the same project.

Syntax :

* content *

2. External Linking (From Our Project to External Pages):

External linking is used to create links that lead to pages outside of the current website or project.

Syntax :

* content *

3. Hash-Based Navigation (Within Page):

Hash-based navigation is used to create links that navigate to specific sections or elements within the same page. It is often used for smooth scrolling to different sections of a long webpage.

Syntax :

```
<!-- Linking to a section within the same page -->
<a href="#section-id">Go to Section</a>
```

In this case, the `#section-id` refers to the `id` attribute of an HTML element within the same page.

example:

```
<h2 id="section-id">Section Title</h2>
<p>This is the content of the section.</p>
```

Figure Tag :

The `<figure>` tag in HTML is used to encapsulate any content that is referenced from the main content, such as images, diagrams, photos, code listings, etc. It is often used in conjunction with the `<figcaption>` (figure caption) element to provide a caption for the content. The `<figure>` and `<figcaption>` elements help to associate a caption with the content it describes, providing context.

Ex :

```
<figure>
  <img src="example.jpg" alt="An example image">
  <figcaption>Caption for the image</figcaption>
</figure>
```

<figure>` Tag:

- Encloses the content being referenced (in this case, an image).
- It is a container for the referenced content.

<img>` Tag:

- The actual content being referenced (image, diagram, etc.).

- The `src` attribute specifies the URL of the content.

- The `alt` attribute provides alternative text for accessibility.

<figcaption>` Tag:

- Provides a caption for the content within the `<figure>` element.

- Describes or adds context to the content.

This structure is useful for semantic HTML and is particularly helpful for accessibility. Screen readers, for example, can use the association between the `<figure>` and `<figcaption>` to provide additional information about the content to users with visual impairments.

Example :

```
<figure>
  <code>
    <pre>
      // Some code snippet
      function example() {
        console.log("Hello, World!");
      }
    </pre>
  </code>
  <figcaption>Code snippet example</figcaption>
</figure>
```

<section>:

Functionality:

Represents a thematic grouping of content, typically with a heading.

Working:

It helps in organizing the content of a page into sections, making it semantically meaningful and improving accessibility.

<header>:

Functionality:

Represents introductory content at the beginning of a section or page.

Working:

Typically used to contain headings, logos, navigation menus, and other elements that are usually found at the top of a page or section.

<main>:

Functionality:

Represents the main content of the document.

Working:

It contains the primary content of the web page, excluding headers, footers, and sidebars. It's intended to directly relate to or be the central focus of the document.

<nav>:

Functionality:

Represents a section of the page dedicated to navigation links.

Working:

It's used to define a navigation menu. Inside a <nav> element, you might find a list of links or other navigation-related elements.

<footer>:

Functionality:

Represents a footer for a section or page.

Working:

It typically contains metadata, copyright information, links to related documents, or other information relevant to the section or page.

:

Functionality:

Represents an inline container that doesn't inherently represent anything.

Working:

It's used to apply styles or scripting to a specific part of the text without affecting the structure. It's often used with CSS to style specific portions of text.

<article>:

Functionality:

Represents a self-contained piece of content that could be distributed and reused independently.

Working:

It's used for content that makes sense on its own, such as a news article, blog post, or forum post. It often includes a heading and may contain images, paragraphs, or other elements.

<aside>:

Functionality:

Represents content that is tangentially related to the content around it, such as a sidebar.

Working:

It's often used for content like a sidebar, pull quote, or related links. It is not the primary content but provides additional context or supplementary information.

Form Tags :

This form uses the following attributes:

1. The <**form**> tag uses the action attribute to specify the URL of the page where the form data will be submitted, the method attribute to specify the HTTP method that will be used to submit the form data, and the id attribute to specify the ID of the form.
    a. In forms we have the following attributes
        i. **Action** : defines what type of action need to perform in that form
            1. ***Syntax*** : action = "Relative URL & Absolute URL or empty"

ii. **Method** : defines how the data needs to transfer.

1. ***Syntax*** :

method = “GET /POST /PUT /DELETE”

1. ***GET*** : Appends form data to the URL in a query string. Suitable for small amounts of data and non-sensitive information.
2. ***POST*** : Sends form data in the body of the HTTP request. Suitable for large amounts of data and sensitive information. It is more secure to transfer the data.

2. The <**input**> tags use the id, name, type, placeholder, required, max, min, maxlength, pattern, value, size, accept, alt, checked, disabled, readonly, src, and width attributes.
   a. We use different input types to accept the input from user.
      i. **Text** – accepts text from user
      ii. **Checkbox** – allows user to select multiple options
      iii. **Radio** – gives a radio button to select the between multiple options
      iv. **Submit Button** – acts as submit
      v. Here's a comprehensive list of input types:

1. ****Text Input:****

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

2. **Password Input:**

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

3. **Radio Buttons:**

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

4. **Checkboxes:**

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

5. **Submit Button:**

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

6. **Reset Button:**

```
<input type="reset" value="Reset">
```

7. **File Input:**

<input type="file" name="fileupload">

8. **Hidden Input:**

<input type="hidden" name="secret" value="hidden_value">

9. **Number Input:**

<input type="number" name="quantity" min="1" max="10" step="1" value="1">

10. **Range Input:**

<input type="range" name="volume" min="0" max="100" value="50">

11. **Date Input:**

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

12. **Time Input:**

<input type="time" name="meeting_time">

13. **Color Input:**

<input type="color" name="bgcolor" value="#ff0000">

14. **Email Input:**

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

15. **URL Input:**

<input type="url" name="website_url">

16. **Tel Input (Telephone Number):**

<input type="tel" name="phone_number">

17. **Search Input:**

<input type="search" name="search_query">

18. **Month Input:**

<input type="month" name="month_year">

19. **Week Input:**

<input type="week" name="week_number">

20. **Datetime Input:**

<input type="datetime-local" name="event_datetime">

21. **Textarea:**

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

22. **Button (Generic Button):**

<button type="button">Click me</button>

We have the following attributes for input tag

vi. **Type** : what type of input like text, checkbox, radio,…
   1. ***Syntax*** :
      type = ” Text/ radio/ checkbox/submit/ reset/file”

vii. **Name** : used to identify or group the set of data together
   1. ***Syntax*** :
      name = “DefineGrpName”

viii. **Placeholder** : used to give a name inside the textbox/ area so that the user can understand what needs to enter.
   1. ***Syntax*** :
      placeholder = “description”

ix. **Value** : used to give a certain value for the so that it will collect at server side
   1. ***Syntax*** :
      value = “name/value”

x. **Checked** : will the check the selected value when the form loaded by default.
   1. ***Syntax*** :
      <input type = “checkbox/radio” checked>

3. The <**label**> tags use the for attribute to specify the ID of the input field that the label is associated with.
4. The <**select**> tag uses the id, name, size, multiple, and disabled attributes.
   a. The <**option**> tags use the value, label, and selected attributes.

b. Syntax :

```
<select>
    <option> dropDown value 1</option>
    <option> dropDown value 2</option>
    <option> dropDown value 3</option>
    ……
    <option> dropDown value n</option>
</select>
```

Note : We can use the selected attribute to select a specific option by default.

5. The <**textarea**> tags use the id, name, rows, cols, placeholder, and wrap attributes.
    a. Will give a some text area to enter the data to the client.
    b. There are some attributes we can use for the text area input tag
        i. ***Rows*** : will create the rows to enter the data (**height**).
        ii. ***Cols*** : will create the columns to enter the data (**width**).
6. The <**button**> tags use the type, form, value, and disabled attributes.
    a. We can use the same submit and reset input types to this one as well.
7. The <**fieldset**> tags use the id, name, and disabled attributes. Group related elements together.
    a. The <**legend**> tag uses the align attribute.
8. The <**datalist**> tag uses the id attribute.
    a. Is similar to select tag but the only difference is it will give suggestions instead of values.
    b. Due to that we can enhance the user experience.
        i. Syntax :

```
<input list="listName" >
 <datalist id=" listName ">
    <option value="value 1">
    <option value="value 2">
    <option value="value 3">
```

……

<option value="value N">

</datalist>

9. We can also use the **disabled** attribute to disable the input field.

Address :

Address is a tag that it is styled specially by the browser which will represents the author or the article owner address.

Example :

```
<address>
   Here we write the address/ contact info
      <p>CodingWiz Academy</p>
   <p>India</p>
</address>
```

Output :

Block Quote :

<blockquote> : Defines a block of quoted content in a single place.

Example :

```
<blockquote>
   <q>The only way to do great work is to love what you
do.</q>
```

```
            <footer>- Steve Jobs</footer>
        </blockquote>
```

Output :

Here is a famous quote:

"The only way to do great work is to love what you do."
- Steve Jobs

Cite :

Defines the title of a creative work (e.g., a book, movie, song, etc.).

Example :

```
        <cite>
         This is title
        </cite>
```

Output :

This is title

Time :

It is a semantic tag which is used for the structural purpose.

Example :

```
    <time datetime="YYYY-MM-DD">1999-11-18</time>
```

Output :

1999-11-18

Abbr :

It is used to define the abbreviation and it will use the dotted lines to highlight

Example :

```
        <abbr title="info">This is a abbrevation</abbr>
```

Output :

This is a abbrevation

Data :

It's typically used when you want to include data that isn't necessarily meant to be displayed to the user but could be accessed by scripts or other programs.

Example :

```
<data value="34">34</data>
```

Output :

34

Code :

Used to write the computer programming languages or some code snippets.

With the help of pre it will make it display as we write on the file

Example :

```
 <code>
   <pre>
       let repeat = 0;
       for(let num = 1; num <= 100; num++) {
           repeat += num;
       }
       console.log(repeat);
   </pre>
</code>
```

Output :

```
let repeat = 0;
for(let num = 1; num <= 100; num++) {
    repeat += num;
}
console.log(repeat);
```

Samp :

Used to provide the output for the code mentioned in the code block.

Example :

```
<h4>output : </h4>
 <samp>
    Here is the output for the above code
 </samp>
```

Output :

output :

Here is the output for the above code

Kbd :

It is used to represent the keyboard keys which will provide a font with the monospace which will differ them from the other tags.

Example :

```
Lorem ipsum dolor, sit amet consectetur adipisicing elit.
Laudantium error                    ducimus
<b><q><kbd>ctrl</kbd> </q> + <q><kbd>S</kbd></q></b>, ab
aperiam
```

Output:

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Laudantium error ducimus **“ctrl ”** + **“S”**, ab aperiam

Sup :

It is used to represents the superscripts

Example :

```
x <sup>2</sup>
```

Output :

x^2

Sub :

It is used to represents the subscripts

Example :

```
log <sub>2</sub>
```

Output :

$\log_2$