Introduction to HTML

HTML stands for **HyperText Markup Language**. It is the standard language used to create and structure content on the web. It defines the structure of a webpage by using a series of elements, tags, and attributes to organize text, images, links, and other multimedia elements.

- HTML is a markup language, not a programming language, meaning it annotates text to define how it is structured and displayed by web browsers.
- It forms the building blocks of all websites and is complemented by CSS for style and JavaScript for interactivity.
- It is a **static language**, meaning that it does not inherently provide interactive features but can be combined with CSS for styling and JavaScript for interactivity.

In a nutshell, HTML is all about **organizing and displaying information** on a webpage. We can think of it as the **bones** or **structure** of a webpage.

Basic HTML Code Example

```
<!DOCTYPE html>
<html>

<head>
   <title>My First Webpage</title>
</head>

<body>
   <h1>Welcome to My Webpage</h1>
   <p>This is my first paragraph of text!</p>
</body>

</html>
```

Output:

In this example:

- **<!DOCTYPE html>**: Declares the document type and version (HTML5).
- **<html>**: The root element that wraps all HTML content.
- **<head>**: Contains meta-information about the webpage, like the title.

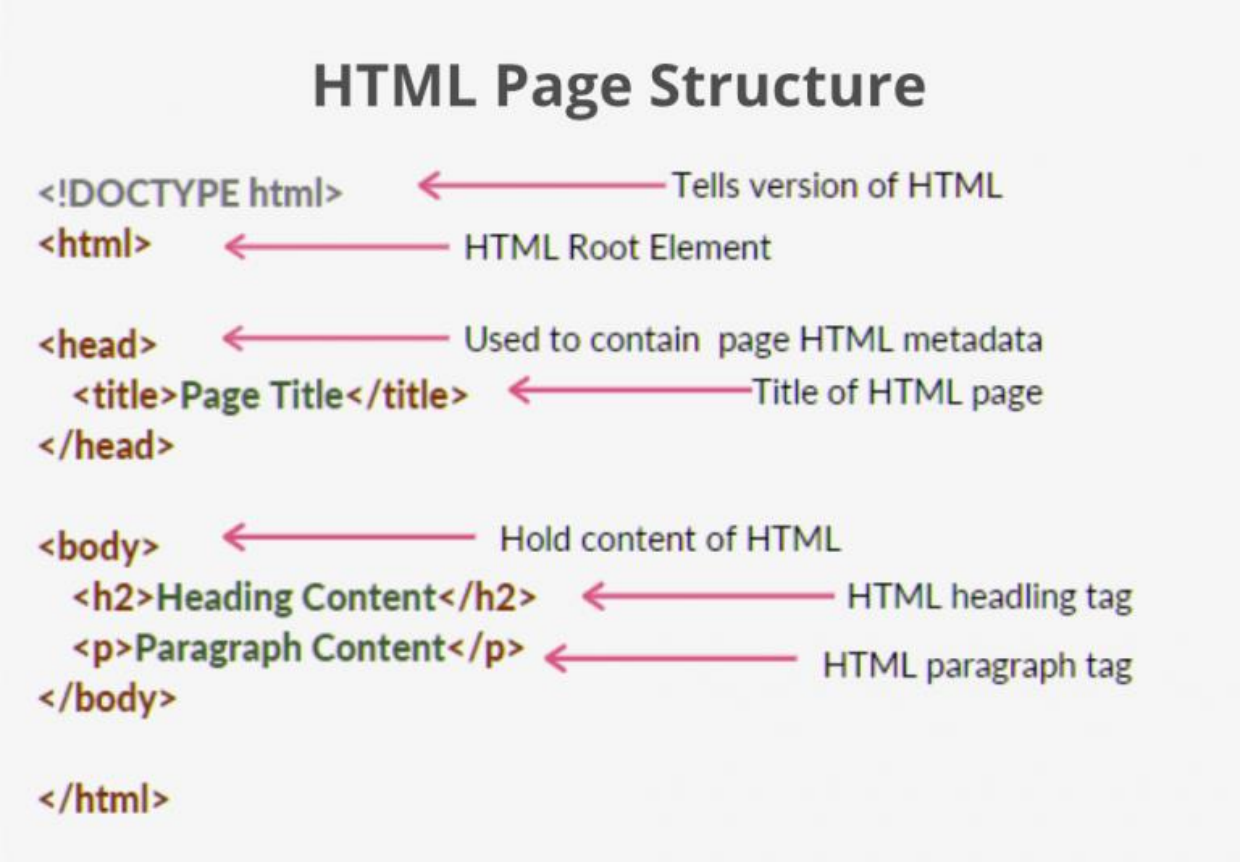
- **<title>:** Specifies the title of the webpage (appears in the browser tab).
- **<body>**: Contains the visible content of the webpage.
- **<h1>**: Represents the main heading on the page ("Welcome to My Webpage").
- **<p>**: Defines a paragraph of text ("This is my first paragraph of text!").

Key Features of HTML

- **Markup Language:** HTML uses tags to markup content. Each tag defines different elements, such as headings, paragraphs, tables, links, etc.
- **Semantics:** HTML provides semantic tags that describe the meaning of the content. For example, <article>, <footer>, <header>, and <nav> describe different types of content on a webpage.
- **Responsive Web Design**: HTML supports various features for building responsive websites, including media queries and the ability to embed multimedia content.
- **Interactive Content:** HTML can embed interactive content using JavaScript, which allows for dynamic changes in the content.

HTML Page Structure

The basic structure of an HTML page is shown below. It contains the essential building-block elements (i.e. doctype declaration, HTML, head, title, and body elements) upon which all web pages are created.

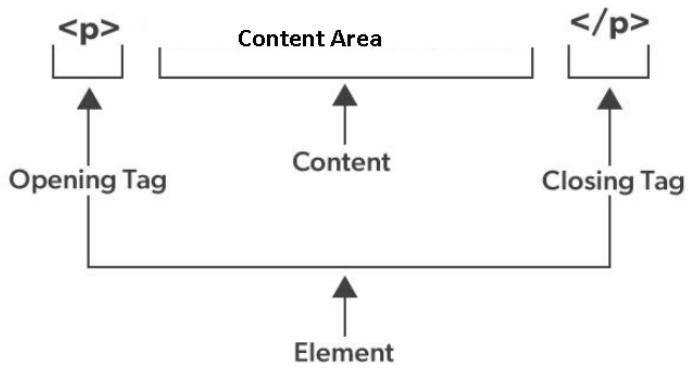


- **<!DOCTYPE html>** – This is the document type declaration, not a tag. It declares that the document is an HTML5 document.
- **<html>** – This is called the HTML root element. All other elements are contained within it.
- **<head>** – The head tag contains the "behind the scenes" elements for a webpage. Elements within the head aren't visible on the front end of a webpage. Typical elements inside the <head> include:
    - **<title>:** Defines the title displayed on the browser tab.
    - **<meta>**: Provides information like the character set or viewport settings.
    - **<link>**: Links external stylesheets or resources.
    - **<style>**: Embeds internal CSS styles.
    - **<script>**: Embeds JavaScript for functionality.
- **<title>** – The title is what is displayed on the top of your browser when you visit a website and contains the title of the webpage that you are viewing.
- **<h2>** – The <h2> tag is a second-level heading tag.
- **<p>**– The <p> tag represents a paragraph of text.
- **<body>** – The body tag is used to enclose all the visible content of a webpage. In other words, the body content is what the browser will show on the front end.

An HTML document can be created using an **HTML text editor**. Save the text file using the "**.html"** or "**.htm"** extension. Once saved as an HTML document, the file can be opened as a webpage in the browser.

HTML Elements and HTML Tag

HTML Elements and HTML Tags are related but distinct. An **HTML element** is the complete structure, including the opening tag, content (if any), and the closing tag (if applicable).
On the other hand, A **tag** is the actual keyword or name enclosed in angle brackets (< >) that tells the browser what kind of content to expect.



Tag	Description
<html>	The root element of an HTML document

<head>	Contains meta-information about the webpage
<body>	Contains the visible content of the webpage
<h1> to <h6>	Headings of various levels (h1 being the largest)
<p>	Defines a paragraph
<a>	Defines a hyperlink
<img>	Embed an image
	Defines an unordered list
	Defines an ordered list
	Defines a list item
<table>	Defines a table
<form>	Defines an HTML form

HTML Attributes

Attributes provide additional information about an element. They are placed inside the opening tag and are written as name=”value”. Common attributes include class, id, href, and src.

Example:

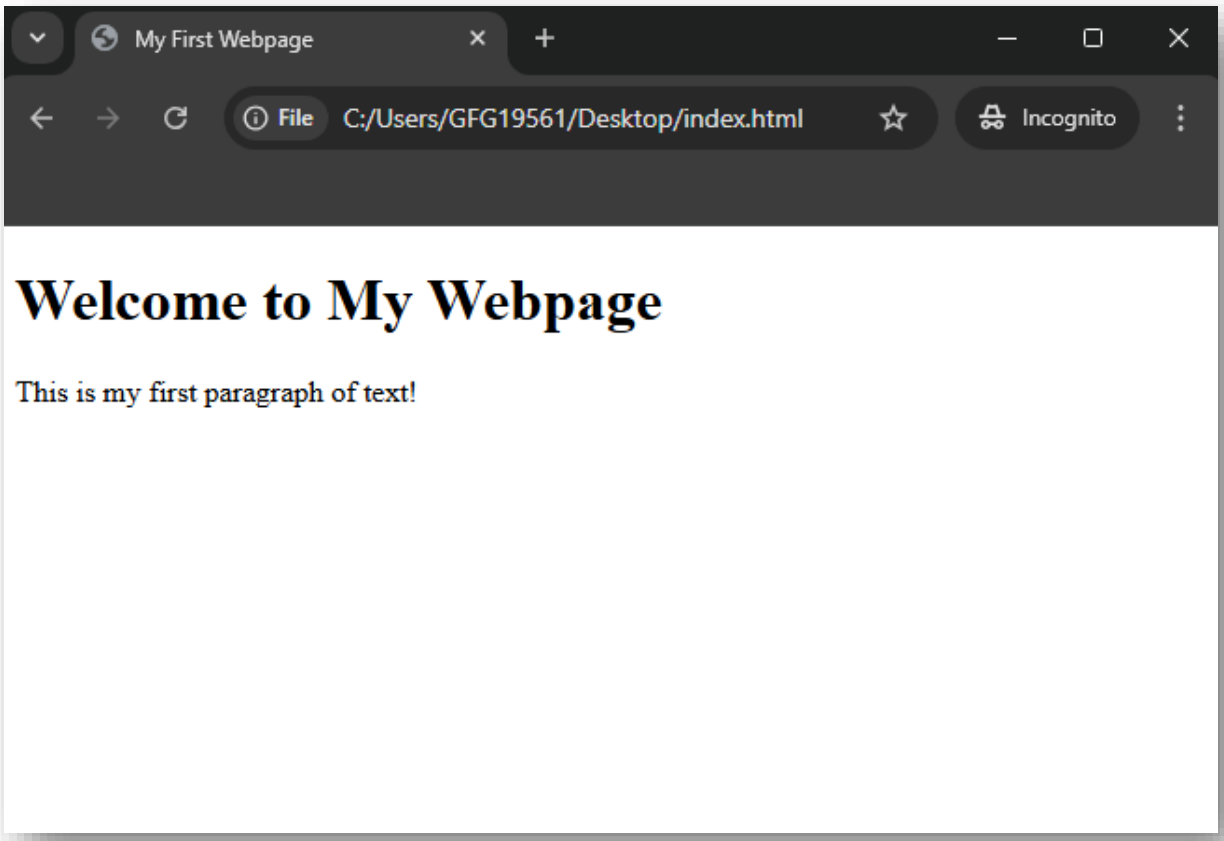
```
<a href="https://www.example.com">Visit Example</a>
```

- href is an attribute of the <a> tag that defines the URL of the link.

Web Browsers

Unlike other programming languages, HTML does not show output on the compiler. Web browsers show the results of an HTML code.

It reads HTML files and determines how to show content with the help of HTML tags. Any web browser (**Google, Safari, Mozilla Firefox**, etc) can be used to open a **HTML file** and view the results.



Why Learn HTML?

Here are **5 common reasons to learn HTML**:

1. **Build Websites**: HTML is the basic building block for creating any website. Learning HTML can help you pursue a career in web development.
2. **Customize Content**: Allows you to edit or tweak web pages, emails, or templates to fit your needs.
3. **Understand how the web works**: This helps you grasp how the internet works and how web pages are structured.
4. **Employment Opportunities**: According to the Bureau of Labor Statistics, projects that employment for web developers will grow 16% between 2022 and 2032, which is much faster than the average across all occupations.
5. **Learn Easily**: HTML is beginner-friendly, making it a great first step into the world of coding and technology.

Applications of HTML

- **Web Development:** HTML is the backbone of every webpage. It structures the content and integrates multimedia, hyperlinks, and more.

- **Web Applications:** HTML, in combination with CSS and JavaScript, powers complex web applications (e.g., Google Docs, Trello).
- **Emails:** HTML emails use table-based layouts and embedded media to deliver rich, interactive content.
- **Mobile App Development:** HTML5 is used with frameworks like PhoneGap to build mobile apps for iOS and Android.

Limitations of HTML

- **No Logic or Functionality:** HTML cannot handle complex logic, interactivity, or dynamic content on its own. It requires JavaScript for such tasks.
- **SEO Limitations:** While HTML provides structure, it's not enough by itself for search engine optimization (SEO). Proper metadata and content structuring, as well as external SEO practices, are necessary.
- **Limited Styles:** While HTML can handle basic styles via the style attribute, it is typically complemented by CSS for complex styling and layout.

HTML5: Enhancements and New Features

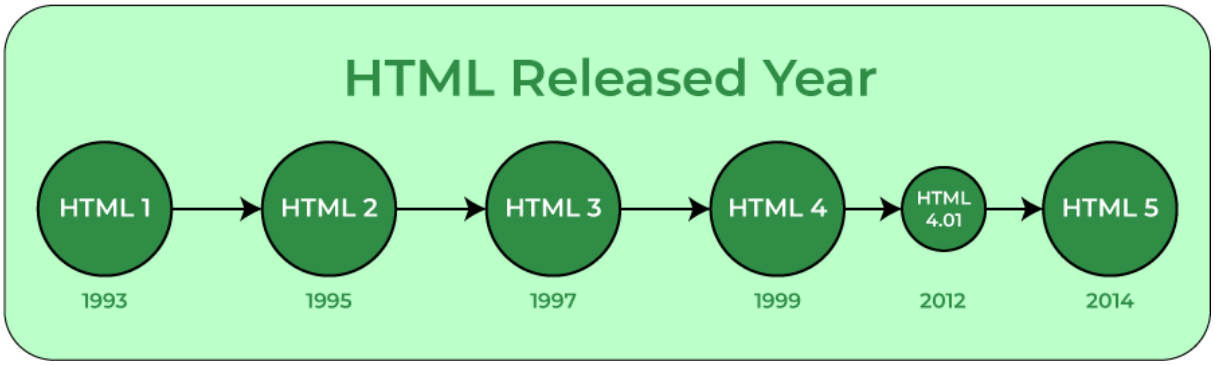
HTML5 introduced several powerful features that improve the structure and functionality of web pages, including:

- **Semantics:** New tags like <article>, <footer>, <header>, and <section> to improve the meaning of the content.
- **Multimedia**: <audio> and <video> tags for embedding audio and video without plugins.
- **APIs:** New APIs like Geolocation, Web Storage, and Canvas allow for more dynamic content and interactive websites.

HTML History

Currently, we are using **HTML5**, which is the latest and most advanced version of HTML.

- HTML was initially created by **Tim Berners-Lee** in 1991 as a way to share and structure documents on the web.
- The first-ever version was **HTML 1.0**, a basic and limited version. However, the first standardized version, **HTML 2.0**, was published in 1995, laying the foundation for web development as we know it today.



Conclusion

In conclusion, mastering HTML is a fundamental step in your web development journey. This guide serves as a comprehensive resource for understanding HTML, from the basics to more advanced topics. Remember, HTML is more than just a markup language – it's a powerful tool for creating engaging, accessible, and SEO-friendly websites.

HTML Lists

An HTML List allows you to organize data on web pages into an ordered or unordered format to make the information easier to read and visually appealing. HTML Lists are very helpful for creating structured, accessible content in web development.

Types of HTML Lists

There are three main types of lists in HTML:

- **Unordered Lists (<ul>):** These lists are used for items that do not need to be in any specific order. The list items are typically marked with bullets.
- **Ordered Lists (<ol>):** These lists are used when the order of the items is important. Each item in an ordered list is typically marked with numbers or letters.
- **Description Lists (<dl>):** These lists are used to contain terms and their corresponding descriptions.

Basic Example of HTML Lists

```
<!DOCTYPE html>
<html>

<body>
   <h2>Welcome to NAST College</h2>
   <h5>List of available courses</h5>
   <ul>
      <li>Data Structures & Algorithm</li>
      <li>Web Technology</li>
      <li>Aptitude & Logical Reasoning</li>
      <li>Programming Languages</li>
   </ul>

   <h5>Data Structures topics</h5>
   <ol>
      <li>Array</li>
      <li>Linked List</li>
```

```
      <li>Stacks</li>
      <li>Queues</li>
      <li>Trees</li>
      <li>Graphs</li>
   </ol>

</body>

</html>
```

HTML List Tags

Tag	Description
	Defines an unordered list.
	Defines an ordered list.
	Defines a list item.
<dl>	Defines a description list.
<dt>	Defines a term in a description list.
<dd>	Details the term in a description list.

Details about the types of List

1. Using HTML Unordered List or Bulleted List

Unordered lists are ideal for scenarios where the sequence of the items is not important.

The unordered list items are marked with bullets, also known as bulleted lists. An unordered list starts with the <ul> tag, and each list item begins with the <li> tag.

Syntax:

```
<ul>
   <li>Item 1</li>
   <li>Item 2</li>
   <li>Item 3</li>
</ul>
```

Attribute: This tag contains two attributes which are listed below:

- compact: It will render the list smaller.
- type: It specifies which kind of marker is used in the list.

Example:

This example describes the unordered list.

```
<!DOCTYPE html>
<html>
```

```
<body>
   <h2>Grocery list</h2>
   <ul>
      <li>Bread</li>
      <li>Eggs</li>
      <li>Milk</li>
      <li>Coffee</li>
   </ul>
</body>

</html>
```

2. Using HTML Ordered List

Ordered lists are used when the items need to follow a specific sequence.

In an ordered list, all list items are marked with numbers by default. An ordered list starts with the <ol> tag, and each list item begins with the <li> tag.

```
<ol>
  <li>Item1</li>
  <li>Item2</li>
  <li>Item3</li>
</ol>
```

Attributes:

- compact: It defines the list should be compacted (compact attribute is not supported in HTML5. Use CSS instead.).
- reversed: It defines that the order will be descending.
- start: It defines from which number or alphabet the order will start.
- type: It defines which type(1, A, a, I, and i) of the order you want in your list of numeric, alphabetic, or roman numbers.

Example

This example describes the ordered list with the use of reverse, type, and start attribute.

```
<!DOCTYPE html>
<html>
<head>
   <title>HTML ol tag</title>
</head>

<body>
```

```
    <h1 style="color: green">NAST College</h1>
    <h3>HTML ol tag</h3>

<p>reversed attribute</p>
    <ol reversed>
        <li>HTML</li>
        <li>CSS</li>
        <li>JS</li>
    </ol>

<p>start attribute</p>
    <ol start="5">
        <li>HTML</li>
        <li>CSS</li>
        <li>JS</li>
    </ol>
<p>type attribute</p>
    <ol type="i">
        <li>HTML</li>
        <li>CSS</li>
        <li>JS</li>
    </ol>
</body>

</html>
```

Output:

HTML <ol> tag

reversed attribute

3. HTML
2. CSS
1. JS

start attribute

5. HTML
6. CSS
7. JS

type attribute

i. HTML
ii. CSS
iii. JS

Ordered List with different list style

3. Using HTML Description List

A description list is a list of terms, with a description of each term. Description lists are less common but very useful for definitions, glossaries, or any other key-value pairs of items.

The <dl> tag defines the description list, the <dt> tag defines the term name, and the <dd> tag describes each term.

Syntax:

```
<dl>
  <dt>Item 1</dt>
  <dd>Description of Item 1 </dd>
  <dt>Item 2</dt>
  <dd>Description of Item 2</dd>
</dl>
```

Here, <dt> (description term) is used for the term being defined, and <dd> (description details) is used for the description.

Example

This example describes the HTML Description List.

```
<!DOCTYPE html>
<html>

<body>
```

```
  <h2>A Description List</h2>
  <dl>
    <dt>Coffee</dt>
    <dd>- 500 gms</dd>
    <dt>Milk</dt>
    <dd>- 1 ltr Tetra Pack</dd>
  </dl>
</body>

</html>
```

HTML Links Hyperlinks

HTML Links, also known as **hyperlinks**, are defined by the **<a>** tag in HTML, which stands for "anchor." These links are essential for navigating between web pages and directing users to different sites, documents, or sections within the same page.

The basic attributes of the <a> tag include **href**, **title**, and **target**, among others.

Basic Syntax of an HTML Link:

NAST Website

Note: A hyperlink can be represented by an image or any other HTML element, not just text.

Types of Hyperlinks

- Internal
- External
- Part of HTML section

Internal Link Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Internal Linking Example</title>
</head>
<body>
  <nav>
    <ul>
      <li><a href="home.html">Home</a></li>
      <li><a href="about.html">About</a></li>
      <li><a href="gallery.html">Gallery</a></li>
    </ul>
```

```
  </nav>

</body>
</html>
```

An External HTML Link Example

```
<!DOCTYPE html>
<html>

<head>
  <title>HTML Links</title>
</head>

<body>
  <p>Click on the following link</p>
  <a href="https://www.nast.edu.np">
    NAST College
  </a>
</body>

</html>
```

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

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

HTML Links – Target Attribute

The target attribute in the <a> tag specifies where to open the linked document. It controls whether the link opens in the same window, a new window, or a specific frame.

Attribute	Description
_blank	Opens the linked document in a new window or tab.
_self	Opens the linked document in the same frame or window as the link. (Default behavior)

Attribute	Description
_parent	Opens the linked document in the parent frame.
_top	Opens the linked document in the full body of the window.
framename	Opens the linked document in a specified frame. The frame's name is specified in the attribute.

```
<!DOCTYPE html>
<html>

<head>
   <title>Target Attribute Example</title>
</head>

<body>
   <h3> Various options available in the Target Attribute </h3>

   <p>
      If you set the target attribute to "_blank", the link will open in a new browser window or tab.
   </p>
   <a href="https://www.nast.edu.np" target="_blank">NAST College</a>

   <p>
      If you set the target attribute to "_self", the link will open in the same window or tab.
   </p>
   <a href="https://www.nast.edu.np"  target="_self">NAST College</a>

   <p>
      If you set the target attribute to "_top", the link will open in the full body of the window.
   </p>
   <a href="https://www.nast.edu.np" target="_top"> NAST College </a>

   <p>
      If you set the target attribute to "_parent", the link will open in the parent frame.
   </p>
   <a href="https://www.nast.edu.np" target="_parent"> NAST College </a>
```

```
</body>

</html>
```

Linking Different HTML Elements

Element to Interlink	Specific Code
Linking to an image	<img src="image.jpg" alt="Image">
Link to an Email Address	Send Email
Phone Number	Call Now
Button	 <button>Visit Website</button>
Link to Download File	Download File
Link Title	Website Link Text

Linking Part of Section

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Linking to Anchors</title>
</head>
<body>
  <nav>
    <ul>
      <li><a href="#section1">Go to Section 1</a></li>
      <li><a href="#section2">Go to Section 2</a></li>
      <li><a href="#section3">Go to Section 3</a></li>
```

```
    </ul>
  </nav>

  <h2 id="section1">Section 1</h2>
  <p>This is the content of section 1.</p>

  <h2 id="section2">Section 2</h2>
  <p>This is the content of section 2.</p>

  <h2 id="section3">Section 3</h2>
  <p>This is the content of section 3.</p>
</body>
</html>
```

HTML Images

The **HTML <img> tag** is used to embed an image in web pages by linking them. It creates a placeholder for the image, defined by attributes like src, width, height, and alt, and does not require a closing tag.

There are **two ways** to insert the images into a webpage:

- By providing a full path or address (URL) to access an internet file.
- By providing the file path relative to the location of the current web page file.

Basic Example of the <img> Tag:

```
<!DOCTYPE html>
<html>

<body>
  <img src="https://nast.edu.np/wp-content/uploads/2022/08/DSC_2151-min-scaled.jpg"
    alt="BCA HOD Photo" />
</body>

</html>
```

In this example:

- The <img> tag is used to embed an image into the webpage.
- src attribute: Specifies the source URL of the image, which in this example is https://nast.edu.np/wp-content/uploads/2022/08/DSC_2151-min-scaled.jpg. The image is loaded from this URL when the webpage is accessed.

- alt attribute: Provides alternative text for the image, “BCA HOD Photo” which describes the image content. If, for any reason, the image cannot be displayed, the text “BCA HOD Photo” will be shown instead.

Various HTML <img> Tag Attributes

Attribute	Description
src	Specifies the path to the image file.
alt	Provides alternate text for the image, useful for accessibility and when the image cannot be displayed.
height	Specifies the height of the image.
width	Specifies the width of the image.

Set Image Size – Width and Height, title Attribute

The width and height attributes are used to specify the width and height of an image. The attribute values are specified in pixels by default. The width and height attributes are always declared in pixels

```
<!DOCTYPE html>
<html>

<body>
      <img src="https://nast.edu.np/wp-content/uploads/2022/08/DSC_2151-min-scaled.jpg"
       alt="BCA HOD Photo"
       title=” Sunil Bahadur Bist, BCA HOD”
       width="300"
       height="300" />
</body>

</html>
```

HTML Tables

HTML Tables allow you to arrange data into rows and columns on a web page, making it easy to display information like schedules, statistics, or other structured data in a clear format. An HTML table is created using the <table> tag.

Basic HTML Table Structure

An HTML table is created using the <table> tag. Inside the table, we use:

<tr>: Represents a row within the table.

<th>: Represents the header cells in the table.

<td>: Represents a regular data cell within the table.

Each <tr> represents a row, and within each row, <th> or <td> tags represent the cells in that row, which can contain text, images, lists, or even another table.

HTML Table Code Example

```
<!-- index.html -->
<!DOCTYPE html>
<html>

<body>
  <table>
    <tr>
      <th>Firstname</th>
      <th>Lastname</th>
      <th>Age</th>
    </tr>
    <tr>
      <td>Priya</td>
      <td>Sharma</td>
      <td>24</td>
    </tr>
    <tr>
      <td>Arun</td>
      <td>Singh</td>
      <td>32</td>
    </tr>
    <tr>
      <td>Sam</td>
      <td>Watson</td>
      <td>41</td>
    </tr>
  </table>
</body>

</html>
```

s

Firstname	Lastname	Age
Priya	Sharma	24
Arun	Singh	32
Sam	Watson	41

In this Example:

<table>: This tag starts the table. Everything between the opening <table> and closing </table> tags makes up the table.

<tr>: Stands for "table row". Each <tr> tag defines a row in the table.

<th>: Stands for "table header". It's used for the headers of the columns. In this case, "**Firstname**", "**Lastname**", and "**Age**" are headers. Text in <th> tags is usually bold and centered by default.

<td>: Stands for "table data". This tag is used for actual data cells under each column. For instance, "**Priya**" is the data under the "**Firstname**" header, "**Sharma**" under the "**Lastname**", and "**24**" under the "**Age**".

The first <tr> has three <th> elements, setting up the column titles.

The subsequent <tr> tags contain three <td> elements, representing the data for each person listed in the table.

Tags used in HTML Tables

HTML Tags	Descriptions
<table>	Defines the structure for organizing data in rows and columns within a web page.
<tr>	Represents a row within an HTML table containing individual cells.
<th>	Shows a table header cell that typically holds titles or headings.
<td>	Represents a standard data cell, holding content or data.
<caption>	Provides a title or description for the entire table.
<thead>	Defines the header section of a table, often containing column labels.
<tbody>	Represents the main content area of a table, separating it from the header or footer.
<tfoot>	Specifies the footer section of a table, typically holding summaries or totals.
<col>	Defines attributes for table columns that can be applied to multiple columns simultaneously.
<colgroup>	Groups together a set of columns in a table to which you can apply formatting or properties collectively.

Another Example of a border spacing HTML Table:

Creating a simple table in HTML using a table tag.

```
<!-- index.html -->
<!DOCTYPE html>
<html>

<body>
  <table>
    <tr>
```

```
        <th>Book Name</th>
        <th>Author Name</th>
        <th>Genre</th>
      </tr>
      <tr>
        <td>The Book Thief</td>
        <td>Markus Zusak</td>
        <td>Historical Fiction</td>
      </tr>
      <tr>
        <td>The Cruel Prince</td>
        <td>Holly Black</td>
        <td>Fantasy</td>
      </tr>
      <tr>
        <td>The Silent Patient</td>
        <td> Alex Michaelides</td>
        <td>Psychological Fiction</td>
      </tr>
  </table>
</body>

</html>
```

Output:

Book Name	Author Name	Genre
The Book Thief	Markus Zusak	Historical Fiction
The Cruel Prince	Holly Black	Fantasy
The Silent Patient	Alex Michaelides	Psychological Fiction

HTML Forms

HTML Forms use the <form> tag to collect user input through various interactive controls. These controls range from text fields, numeric inputs, and email fields to password fields, checkboxes, radio buttons, and submit buttons.

Syntax:

```
<form>
    <!--form                                                    elements-->
</form>
```

Form Elements

The HTML <form> comprises several elements, each serving a unique purpose. For instance, the <label> element defines labels for other <form> elements. On the other hand, the <input> element is versatile and can be used to capture various types of input data such as text, password, email, and more simply by altering its type attribute.

Elements	Descriptions
<label>	It defines labels for <form> elements.
<input>	It is used to get input data from various types such as text, password, email, etc by changing its type.
<button>	It defines a clickable button to control other elements or execute a functionality.
<select>	It is used to create a drop-down list.
<textarea>	It is used to get input long text content.
<fieldset>	It is used to draw a box around other form elements and group the related data.
<legend>	It defines a caption for fieldset elements
<datalist>	It is used to specify pre-defined list options for input controls.
<output>	It displays the output of performed calculations.
<option>	It is used to define options in a drop-down list.
<optgroup>	It is used to define group-related options in a drop-down list.

Commonly Used Input Types in HTML Forms

In HTML forms, various input types are used to collect different types of data from users. Here are some commonly used input types:

Input Type	Description
<input type="text">	Defines a one-line text input field
<input type="password">	Defines a password field
<input type="submit">	Defines a submit button
<input type="reset">	Defines a reset button
<input type="radio">	Defines a radio button
<input type="email">	Validates that the input is a valid email address.
<input type="number">	Allows the user to enter a number. You can specify min, max, and step attributes for range.
<input type="checkbox">	Used for checkboxes where the user can select multiple options.
<input type="date">	Allows the user to select a date from a calendar.
<input type="time">	Allows the user to select a time.
<input type="file">	Allows the user to select a file to upload.

HTML Forms Example

Example 1: Basic HTML Forms

Example: This HTML form collects user personal information such as username and password with a button to submit the form.

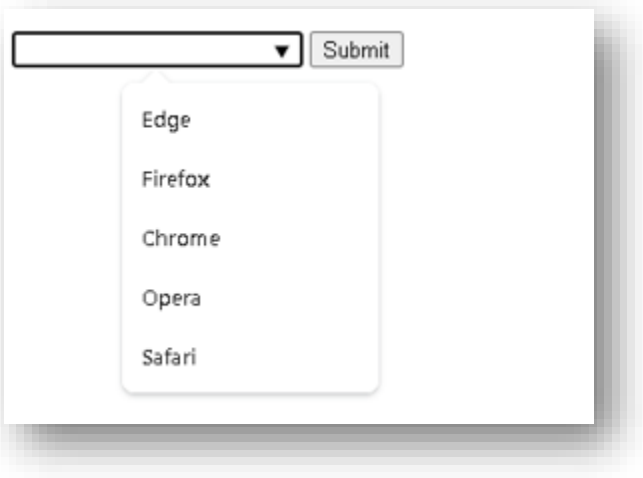
```
<html>
<head>
   <title>Html Forms</title>
</head>

<body>
   <h2>HTML Forms</h2>
   <form>
      <label for="username">Username:</label><br>
      <input type="text" id="username" name="username"><br><br>

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

      <input type="submit" value="Submit">
   </form>
</body>
</html>
```

Output:

In this example:

- **HTML Structure:** The code has a basic HTML structure with a title “HTML Forms.”
- **Heading**: The <h2> tag displays “HTML Forms” as the main heading on the page.
- **Form Tag:** The <form> tag defines a form for user input.
- **Username Field:** A text input field for the username with a label.

- **Password Field & Submit:** A password input field and a submit button to send the form data.




```
<form action="action_page.py">
  <input list="browsers" name="browser">
  <datalist id="browsers">
    <option value="Edge">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
  <input type="submit">
</form>
```


```
<form>
   <table>
      <tr>
         <td>
            <label for="uname">Name</label>
         </td>
         <td>
            <input type="text" id="uname" name="username">
         </td>
      </tr>
      <tr>
         <td>
            <label for="uemail">Email</label>
         </td>
         <td>
            <input type="text" id="uemail" name="usermail">
            <button type="button">Check</button>
         </td>
```

```
</tr>
<tr>
   <td>
      <label for="age">Age</label>
   </td>
   <td>
      <input type="text" name="userage" id="age" size="2" maxlength="2">
   </td>
</tr>
<tr>
   <td>
      <label>Country</label>
   </td>
   <td>
      <input type="text" value="India" name="country" disabled>
   </td>
</tr>
<tr>
   <td>
         <label for="pass">Password</label>
   </td>
   <td>
         <input type="password" id="pass">
   </td>
</tr>
<tr>
   <td>
      <label for="res">Resume</label>
   </td>
   <td>
      <input type="file" id="res">
   </td>
</tr>
<tr>
   <td>
      <label>Hobbies</label>
   </td>
   <td>
      <label>
         <input type="checkbox" checked> Cricket
```

```
        </label>
        <label>
            <input type="checkbox"> Football
        </label>
    </td>
</tr>
<tr>
    <td>
        <label>Gender</label>
    </td>
    <td>
        <label>
            <input type="radio" value="f" name="gender"> Female</label>
            <label>
            <input value="m" type="radio" name="gender"> Male</label>
    </td>
</tr>
<tr>
    <td>
            <label for="city">City</label>
    </td>
    <td>
            <select id="city" name="city">
                <option disabled selected>--Choose City--</option>
                    <optgroup label="Metros">
                        <option>New Delhi</option>
                        <option>Mumbai</option>
                        <option>Channai</option>
                        <option>Kolkata</option>
                    </optgroup>
                    <optgroup label="Others">
                        <option>Noida</option>
                        <option>Gurgram</option>
                        <option>Faridabad</option>
                        <option>Gaziabad</option>
                    </optgroup>
            </select>
    </td>
</tr>
<tr>
```

```
        <td>
          <label>Address</label>
        </td>
        <td>
          <textarea rows="4" cols="40"></textarea>
        </td>
      </tr>
      <tr>
        <td></td>
        <td>
          <input type="submit" value="Submit">
          <input type="reset">
        </td>
      </tr>
    </table>
</form>
```

Assignment Work

HTML5 Semantics

HTML5 introduced a range of **semantic elements** that clearly describe their purpose in human and machine-readable language. Unlike non-semantic elements, which provide no information about their content, semantic elements clearly define their content.

For instance, **<form>**, **<table>**, and **<article>** tags clearly define the content and purpose, to the browser.

Why Use Semantic HTML Tags?

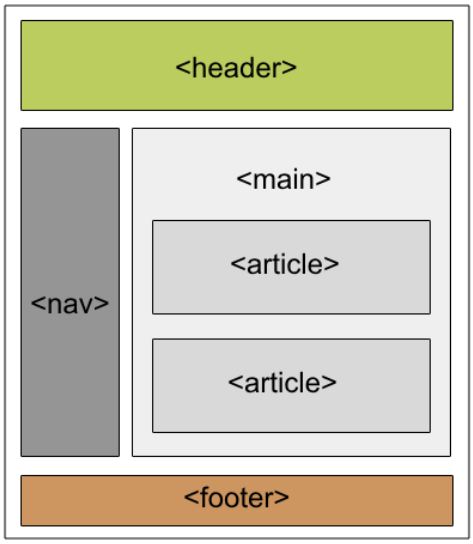
- **Accessibility:** Semantic elements make web pages more accessible. Screen readers and other assistive technologies can interpret the structure and navigate the content more efficiently.

- **SEO:** Better structured data leads to better SEO. Search engines prioritize well-structured content that uses semantic elements correctly, as it's easier to index.
- **Maintainability:** Semantic HTML helps create a logically structured document, which is easier to read and maintain.

Semantic Elements

Here are some of the fundamental HTML5 semantic elements that you should use to structure your web content:

<header>, <footer>, <nav>, <article>, <section>



HTML header Tag

The **<header> tag** is a semantic HTML element that is used to define the introductory or navigational content of a webpage or a section. Typically, a header contains elements like:

- *The website or page title*
- *Logo or branding*
- *Navigation menus*
- *Search bar*
- *Any introductory information relevant to the page or section*

Note: Header tag cannot be placed within a <footer>, <address>, or another <header> element.

Syntax:

```
<header> ...</header>
```

HTML footer Tag

The **<footer>** tag in HTML is used to define the footer section of an HTML document.

- The footer section typically contains information such as contact information, sitemap, back-to-top links, related documents, copyright, etc.

- The footer tag is a semantic tag included in HTML (in 2014) along with other tags like article, nav, header, etc.
- It is not mandatory, but adds to clear structure to the document and useful for SEO.

```
<footer>

</footer>
```

HTML <nav> Tag

The <nav> tag in HTML is used to define navigation sections on a webpage. It typically contains links to important sections of the website, such as menus, tables of contents, or indexes, and is often structured within unordered lists (<ul>) or as standalone links.

How to Use the <nav> Tag

Links within the <nav> tag can either be standalone or structured within an unordered list (<ul>) for better organization. While it's common to use lists, it's not a strict requirement.

Syntax:

```
<nav>
  <!-- Your navigation links here -->
</nav>
```

```
<nav>
      <a href="#">Home</a> |
      <a href="#">Interview</a> |
      <a href="#">Languages</a> |
      <a href="#">Data Structure</a> |
      <a href="#">Algorithm</a>
</nav>
```

HTML article Tag

The **HTML <article> tag** defines a self-contained, independent piece of content like a blog post, news article, or comment. It is designed for content that can be independently distributed, shared, or reused, providing semantic meaning to the content.

This tag is introduced in HTML5.

```
<!DOCTYPE html>
<html>

<body>
   <article>
```

```
      <h3>My First Blog Post</h3>
      <p>
         This is a brief introduction to my blog.
         I will share my thoughts on various topics.
      </p>
   </article>
</body>

</html>
```

HTML <section> Tag

The Section tag defines the section of documents such as chapters, headers, footers, or any other sections. The section tag divides the content into sections and subsections.

The section tag is used when requirements of two headers or footers or any other section of documents are needed. Section tag grouped the generic block of related contents. The main advantage of the section tag is, it is a semantic element, that describes its meaning to both the browser and the developer.

Note: Section tag is used to distribute the content i.e., it distributes the sections and subsections.

Syntax

```
<section> Section Contents </section>
```

Example 1: The implementation of the section tag

```
<!DOCTYPE html>
<html>

<body>
   <!-- html section tag is used here -->
   <section>
      <h1>
         Section 1
      </h1>
      <p>
         Content of section 1
      </p>
   </section>
   <section>
      <h1>
         Section 2
      </h1>
```

```
      <p>
        Content of section 2
      </p>
    </section>
    <section>
      <h1>
        Section 3
      </h1>
      <p>
        Content of section 3
      </p>
    </section>
</body>

</html>
```

<audio> Tag

We use the <audio> tag to embed audio in HTML,. Before HTML5, audio cannot be added to web pages in the Internet Explorer era. To play audio, we used web plugins like Flash. After the release of HTML5, it is possible. This tag supports Chrome, Firefox, Safari, Opera, and Edge in three audio formats – MP3, WAV, OGG. Only Safari browser doesn't support OGG audio format.

Syntax:

```
<audio>
   <source src="file_name" type="audio_file_type">
</audio>
```

Attributes of <audio> tag

Attribute	Value	Description
Autoplay	Autoplay	When the page is loaded. It specifies to play audio as soon as possible.
Controls	Controls	It displays audio control.
Loop	Loop	It will start the audio again when it is finished.
Muted	Muted	When the page is loaded audio will be automatically muted.
Preload	auto metadata none	It specifies how the author thinks the audio will be loaded when the page is ready.
Src	URL	It specifies the URL of the audio file.

For example,

```
<audio controls preload="auto" autoplay muted loop src="ads.mp3"></audio>
```

<video> Tag

To embed video in HTML, we use the <video> tag. It contains one or more video sources at a time using <source> tag. It supports MP4, WebM, and Ogg in all modern browsers. Only Ogg video format doesn't support in Safari browser.

Syntax

```
<video>
   <source src="file_name" type="video_file_type">
</video>
```

Attributes of <video> tag

Attribute	Value	Description
autoplay	autoplay	When the page is loaded. It specifies to play video as soon as possible.
controls	controls	It displays video control such as play, pause, and stop.
loop	loop	It will start the video again when it is finished.
muted	muted	When the page is loaded video will be automatically muted.
poster	URL	It specifies an image will be shown until video play.
preload	auto metadata none	It specifies how the author thinks the video will be loaded when the page is ready.
src	URL	It specifies the URL of the audio file.
width	pixels	It specifies the width of the video area. The default value of width is 'auto'.
height	pixels	It specifies the height of the video area. The default value of height is 'auto'.

For example,

```
<video controls preload="auto" autoplay muted loop width="600" height="400" src="vds.mp4"></video>
```

** and Tag**

In HTML, and are both generic containers used for grouping and styling content, but they have different display behaviors. is a block-level element, meaning it starts on a new line and takes up the full width available, while is an inline element, fitting within the flow of text and only taking up as much space as the content within it requires.

Here's a more detailed breakdown:

** (Block Element):**

- **Purpose:**
  Used to create sections or divisions in a document, grouping other elements together and often used for layout purposes.
- **Display:**
  display: block by default, meaning it starts on a new line and takes up the full available width.
- **Example:**
  You might use a to structure a webpage into a header, main content area, and footer.
  Code

```
<div>
  <h1>My Web Page</h1>
  <p>This is some content. </p>
</div>
```

** (Inline Element):**

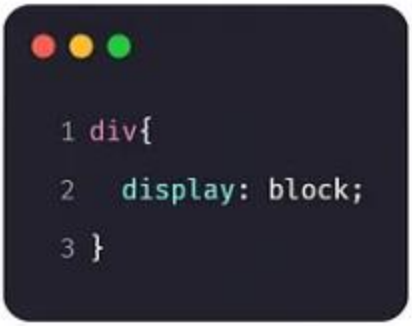
- **Purpose:**
  Used to group inline text elements and apply styling or scripts to specific parts of the text.
- **Display:**
  display: inline by default, meaning it flows within the text and only occupies the space needed by its content.
- **Example:**
  You might use a to highlight specific words in a paragraph or change their color.
  Code

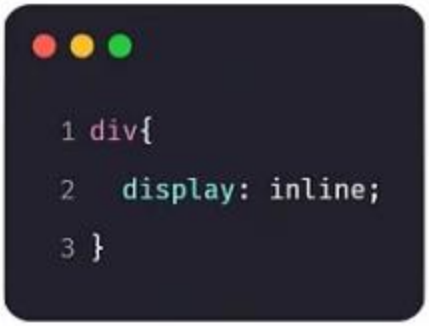
```
<p>This is some <span style="color: blue;">text</span> in a paragraph. </p>
```

Key Differences Summarized:

Feature		
Display	Block (takes up full width)	Inline (flows with text)
Usage	Structuring sections and layout	Styling inline elements or text
Newline	Starts on a new line	Does not start on a new line
Content	Can contain other block or inline elements	Can only contain inline elements

In essence: Use for structuring sections and organizing larger blocks of content, and use for styling or grouping parts of text within a line.





References

https://www.geeksforgeeks.org/
https://www.w3schools.com