

HTML: -

- HTML stands for Hyper Text Markup Language.
- It is the standard language used to create and structure content on the web.
- HTML defines the structure of a webpage by using a system of tags and attributes.
- It helps us to create web pages.
- The current version of html is html5.
- HTML means
 - Hyper – Reference/Link
 - Text – Information/Content
 - Markup - Predefined
 - Language - Communication

History of Html: -

- Html is invented by Tim Berner's LEE in the year 1991.
- HTML 1.0 – (1993) heading, lists, paragraphs, images. Minimal scope of styling
- HTML 2.0 – (1995) basic forms (text field, buttons), table
- HTML 3.2 – (1997) support for internal css, handling images
- HTML 4.01 – (1999) separate css file, field set, header, footer, row span colspan to table.
- HTML 5.0 – (2008) new form elements, audio tag, semantic tag.

Structure of html: -

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="description" content="Describe your website here">
  <title>Basic HTML Structure</title>
</head>
```

HTML Tags Explanation: -

1. <!DOCTYPE html>

Declares the document type.

Specifies that the document is an HTML5 document. Browsers use this to render the page correctly.

[Note: It must be the first line of your HTML file.](#)

2. <html lang="en">

<html>: -The root element that contains all other HTML elements. It represents the start of an HTML document.
lang="en": -

- Specifies the language of the content inside the document (in this case, English).
- Helps browsers and assistive technologies (like screen readers) determine the language for accessibility and SEO purpose.

3. <head>

- Contains metadata (information about the document) that is not displayed on the page itself.
- It can include:
 - Title of the document (<title>).
 - Links to stylesheets, fonts, or icons.
 - Metadata like descriptions, keywords, or authorship.

4. <meta charset="UTF-8">

- Declares the character encoding for the document.
- UTF-8 is a standard character encoding that supports all characters (letters, symbols, emojis, etc.).
- Ensures special characters are displayed correctly (e.g., accented letters like é or ñ).

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

- Makes the page responsive to different screen sizes (desktop, tablet, mobile) - width=device-width: Ensures the width of the content matches the device's screen width.
- initial-scale=1.0: Sets the initial zoom level to 100% when the page loads.

6. <meta name="description" content="Describe your website here">

- a brief description of your webpage content.

- Helps search engines like Google understand your page.
- Often displayed as the snippet in search engine results.

7. <title>

- Sets the title of the webpage.
- This title appears in the browser tab and is used by search engines for indexing.
- It should be descriptive and concise (e.g., "Welcome to My Website").

Tags:-

- Tags are the predefined keywords enclosed between angular braces (< >), having some particular purpose.
- The task of each tag has been already declared.
- Syntax:
`<tagname> content </tagname>`
- Example:
`<p> Hello world </p>`
` Hello world `

Types of Tags:-

- HTML tags were categorized into 2 ways.
- They are:
 1. Paired Tags
 2. UnPaired Tags

1. Paired Tags:

- A tag which is required of both opening and closing tags are called paired tags.
- It is also known as container tags.
- Example:
`<p> Hello Web Dev </p>`

2. UnPaired Tags:

- A Tag which does not require a closing tag is called unpaired tag.
- It is, also known as void tags or self-closing tags.
- Ex:
, <hr>, <input>

Heading Tags: -

- Heading Tags are basically used to provide title or subtitle to the content in the webpage.
- By default, they are in bold format.
- We have a set of heading tags from h1 to h6.
- They are in a set from h1 to h6 where h1 is the of highest size and h6 of lowest size.
- Default Sizes:
 - h1 → 2 em
 - h2 → 1.5 em
 - h3 → 1.17 em
 - h4 → 1 em
 - h5 → 0.83 em
 - h6 → 0.67 em

- Note: 1 em = 16px

Element: -

- An element is a combination of opening tag + content + closing tag.
- <tagname> content </tagname> collectively it is known as element in html
- Html Elements were classified into 3 types:
 1. Inline Level
 2. Block Level
 3. Inline-Block Level

1. Inline Level Element:

- These elements will occupy only content space.
- These elements will display in the same line.
- We cannot assign height and width properties for the inline elements.
- Ex: , <i>, , etc...

2. Block Level Element:

- These elements will occupy the whole viewport width.
- These Elements will display in the next line.

- We can assign height and width properties for the block element.
- Ex: <h1>, <div>, <p>, etc...

3. Inline-Block Level:

- These elements will occupy content space and along with that we can modify the height and width.
- These elements will display in the same line.
- We can assign height and width properties for the block element.
- Ex: <button>: Buttons.
 - <input>: Input fields.
 - <select>: Dropdown menus.
 - : Images.
 - <label>: Labels.

HTML Tags Notes: -

1. Tag

- Displays the content in bold format.
- Syntax:
`This is bold text`
- Example:
`Important Note: Submit your work on time.`

2. Tag

- Displays the content in bold format and indicates its importance to browsers and assistive technologies.
- Syntax:
`This is strong text`
- Example:
`Warning: Do not touch.`

3. <i> Tag

- Displays the content in italic format.
- Syntax:
`<i>This is italic text</i>`
- Example:
`<i>Book Title:</i> The Great Gatsby`

4. Tag

- Displays the content in italic format and provides emphasis, making it important for accessibility tools.
- Syntax:
`This is emphasized text`
- Example:
Please `read` the instructions carefully.

5. <u> Tag

- Underlines the content.
- Syntax: - `<u>This is underlined text</u>`
- Example: `<u>Underline this text for emphasis. </u>`

6. <ins> Tag

- Underlines the content and is an alternative to the <u> tag. It also represents inserted content.
- Syntax:
`<ins>This is inserted text</ins>`
- Example:
`<ins>New changes have been added. </ins>`

7. <strike> Tag

- Strikes off the content (not recommended as it is deprecated).
- Syntax:
`<strike>This text is struck off</strike>`
- Example:
`<strike>Outdated information. </strike>`

8. Tag

- Strikes off the content and is an alternative to <strike>.
- Syntax:
`This text is deleted`

- Example:
`Removed content`

9. `<q>` Tag

- Adds quotations to the content.
- Syntax:
`<q>This is a quote</q>`
- Example:
`<q>The early bird catches the worm. </q>`

10. `<mark>` Tag

- Highlights the content with a yellow background.
- Syntax:
`<mark>This text is highlighted</mark>`
- Example:
Please `<mark>note</mark>` this important point.

11. `<big>` Tag

- Increases the font size of the content.
- Syntax:
`<big>This text is bigger</big>`
- Example:
`<big>Attention:</big>` Follow the rules.

12. `<small>` Tag

- Decreases the font size of the content.
- Syntax:
`<small>This text is smaller</small>`
- Example:
`<small>Disclaimer:</small>` Terms and conditions apply.

13. `<sup>` Tag

- Displays the content as superscript.
- Syntax:
`^{This is superscript}`
- Example:
`X²` represents X squared.

14. `<sub>` Tag

- Displays the content as subscript.
- Syntax:
`_{This is subscript}`
- Example:
`H₂O` is the chemical formula for water.

15. `<pre>` Tag

- Displays preformatted text and preserves spaces, tabs, and line breaks.
- Syntax:
`<pre>`
This is preformatted text.
It preserves spaces and line breaks.
`</pre>`
- Example:
`<pre>`
HTML tags:
`Bold`
`<i>Italic</i>`
`</pre>`

Deprecated Tags: -

- These are the tags which will be removed in further versions.
- Vs code gives suggestions by displaying in red color.

Image Tag: -

- `Img` tag helps us to use images in web pages.

- Img tag is a self closing tag.
- It is an inline-block element.
- Syntax: -
- Attributes of image Tag:
 - Src → It will specify the path address of an image
 - Alt → It will provide an alternate name to the image.
 - Height → It helps to modify the height of an image.
 - Width → It helps to modify the Width of an image.

- path address we can provide by using two ways.

- 1.Absolute path
- 2.Relative Path

1. Absolute Path: -

- An absolute path is the full path to a file or directory starting from the root directory of the file system.
- It is independent of the current working directory and always provides a complete address for locating the resource.

2.Relative Path: -

- A relative path provides the location of a file or directory relative to the current working directory.
- It depends on the location where the command or operation is being executed.

Anchor Tag:-

- Anchor tag helps us to navigate from one web page to another webpage.
- By using anchor tags, we can create hyperlinks in a webpage.
- Syntax:


```
<a href="" path Address" target="_self"> content </a>
```
- Whatever we give in the content space that becomes a hyperlink.
- Content can be text, image, button, icon, etc.
- Attributes:
 1. href → It will specify the path address to where we have to navigate.
 - # - Render the page
 - Internal html file – render that particular html file
 - Website link (external url) – move to that particular web page of website
 - Id – for traversing in the same page or a particular tag of other web page
 2. download → It indicates to download the file
 3. target attribute – specify where the new web page has to open
 - _self – same tab (default)
 - _blank – new tab
 - _top – to your top most parent page
 - _parent – parent tag

Default colors:

1. Visited link → purple.
2. Unvisited link → blue.
3. Active link → red

Attributes:

- Attributes are also known as properties.
- Added extension of a particular tag, providing additional information to element.
- Attributes are case sensitive.
- Four core attributes that are used in a majority of html elements
 - 1.Id - specific unique identifier for element, to identify the element and its content.
 - If you have two elements with same tag, we can distinguish between them using id. Targeted with # symbol.
 - 2.Class – used for grouping elements together, to access them all together. Targeted with . symbol.
 - 3.Style – used for inline css
 - 4.Title – Used to provide tool tip which is shown on hover.

Lists: -

- HTML Lists allows to group a set of related items together.
- Group a set of related items/information together.
- It provides a well-structured and presentable content
- HTML lists were classified into 3 ways.
- They are:
 1. Ordered List
 2. Unordered List
 3. Description List

1. Ordered List: -

- Ordered list group a set of related items in a sequential format.
- As the name suggest, this list take into consideration the number/order in which the elements are entered.
- It is Also knows as sequential list. Defined using `` tag with list item in it.
- Tags:
 - i. `ol` → It indicates that we are creating an ordered List.
 - ii. `li` → It indicates List Items
- Attributes:
 - i. Type → Specifies the type of sequence.
Values = 1 (default), A , a, I , i
 - ii. Start → Specifies the Starting Position.
Values = 1 (default) , Any Number
 - ii. Reversed → It will reverse the sequence

2. Unordered List: -

- As the name suggest, this list doesn't consider the order of items present in the list.
- It uses symbols for the grouping of items.
- Unordered List is also known as Bulleted List
- Unordered List is used to display a set of related items with special symbols.
- Tags:
 - i. `ul` → It indicates that we are creating an Unordered List.
 - ii. `li` → It indicates List Items.
- Attributes:
 - i. Type → Specifies the type of symbol.
Values = disc (default), circle, square, none

3. Description List: -

- Description List is used to display the description data along with description terms.
- It is used to provide the description about the terms specified.
- The parent tag used is `<dl></dl>`. The child tags used are `dt` and `dd`
- Tags:
 - i. `dl` → Indicates Description List
 - ii. `dt` → Indicates Description term
 - iii. `dd` → Indicates Description Definition

Tables:-

- HTML tables allow user to arrange data in the form of rows and columns.
- In earlier days it was used for providing the alignment to web page.
- To define the table we use the table tag which is the parent tag along with that we use some child tags.
- To specify the rows we use the tr tag, each row will have some data for that we use the th tag if the data is heading otherwise we use the td tag for providing data to the cells
- If we want to apply any border we use the border attribute in table tag with numeric value specifying the width of border to the table and cells. This is the deprecated attribute.
- thead, tbody, tfoot – These tags don't provide any functionality, these are used to separate our headings, body, and footer in our table for the better reading of our code and easy understanding.
- If we want to merge cells over row, we use colspan attribute in the respective cell.
- If we want to merge cells over column we use rowspan attribute in the respective cell.
- If we want to specify the space between cells we use cell spacing attribute in table tag.
- If we want to specify the space between content and cell border, we use cell padding attribute in table tag.
- Caption tag is used to provide title to the table.

Tags:-

1. `table` → It defines the creation of a table.
2. `tr` → It defines table rows.
3. `td` → It defines table columns.
4. `th` → It defines table heading. (bold & aligns to center)
5. `thead` → It combines all the header cells together.
6. `tbody` → It combines all the body cells together.
7. `tfoot` → It combines all the footer cells together.
8. `caption` → It provides a title to the table tag.

Attributes:-

- border & cellpadding & cellspacing:
- These 3 attributes we have to use in a table tag.
- By default, the table will not display the border.

- To get the border to the table, we have to use border attribute.
- Cellspacing → It will provide the space between the cells.
- Cellpadding → It will provide the space between the content and cell border
- rowspan & colspan:
 - These 2 attributes we have to use in th and td tags.
 - Rowspan → It will merge the cells over the rows.
 - Colspan → It will merge the cells over the columns.

Semantic tags:-

- The semantic tags are those tags which will say the purpose of it .
- semantic tags were introduced in HTML5 version .
- There are various types of semantic tags .
- All the semantic tags are paired tags .
- All the semantic tags are blocked level elements .
- All semantic tags are :-

1. Header
2. Details
3. Aside
4. Main
5. Nav
6. Section
7. Time
8. Article
9. Footer
10. Figure / Figure caption

Forms:

- Used to collect data/information from user.
- It is block level element.
- It is a document which stores information from user on a webserver.
- It is the process of collecting info in an electronic format.
- most of the form elements are inline level.
- Syntax:


```
<form action=""> </form>
```

HTML Form Tags with Syntax and Examples:

1. <form>

- The <form> element is used to create an HTML form for user input.
- It acts as a container for various input fields, checkboxes, radio buttons, and other interactive elements.
- Syntax:


```
<form action="submit.php" method="post">
    <!-- Form elements go here -->
</form>
```
- Example:


```
<form action="submit.php" method="post">
    <label for="name">Name:</label>
    <input type="text" id="name" name="name">
    <button type="submit">Submit</button>
</form>
```
- This form collects user input for a name field and submits it using the POST method to submit.php.
- The action attribute specifies the server endpoint to which data is sent, and the method determines how the data is transmitted.

2. <label>

- The <label> tag defines a label for an <input> element, improving accessibility and usability by allowing users to click on the label to focus the associated input field.
- Syntax:


```
<label for="id">Label Text</label>
```
- Example:


```
<label for="email">Email:</label>
<input type="email" id="email" name="email">
```

- The for attribute in <label> links it to the corresponding input field using its id. This improves user interaction and accessibility.

3. <input>

- The <input> tag is used to create input fields for text, passwords, checkboxes, radio buttons, and other types of user input.

- Syntax:

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

- Example:

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

- This input field is for entering passwords, with a placeholder that guides the user.

4. <select>

- The <select> element creates a drop-down list, allowing users to choose from multiple options.

- Syntax:

```
<select name="options">
  <option value="1">Option 1</option>
  <option value="2">Option 2</option>
</select>
```

- Example:

```
<select name="fruit">
  <option value="apple">Apple</option>
  <option value="banana">Banana</option>
</select>
```

- Users can select a fruit from the dropdown menu.

5. <optgroup>

- The <optgroup> element groups related options within a <select> menu, making it easier to categorize selections.

- Syntax:

```
<optgroup label="Group Name">
  <option value="value1">Option 1</option>
</optgroup>
```

- Example:

```
<select name="cars">
  <optgroup label="German Cars">
    <option value="bmw">BMW</option>
    <option value="audi">Audi</option>
  </optgroup>
</select>
```

- the options are grouped under "German Cars" for better organization.

6. <option>

- The <option> tag defines individual selectable options in a <select> dropdown list.

- Syntax:

```
<option value="value">Option Text</option>
```

- Example:

```
<select name="colors">
  <option value="red">Red</option>
  <option value="blue">Blue</option>
</select>
```

- Each <option> represents a selectable choice.

7. <datalist>

- The <datalist> provides a list of predefined autocomplete suggestions for an <input> field.

- Syntax:

```
<datalist id="data">
  <option value="Item1">
</datalist>
```

- Example:

```
<input list="browsers" name="browser">
<datalist id="browsers">
  <option value="Chrome">
  <option value="Firefox">
```


`</datalist>`

- As users type, matching options appear as suggestions.

8. `<fieldset>`

- The `<fieldset>` element groups form fields together, making forms more organized.
- Syntax:

```
<fieldset>
    <legend>Group Name</legend>
    <!-- Form elements -->
</fieldset>
```

- Example:

```
<fieldset>
    <legend>Personal Information</legend>
    <label for="age">Age:</label>
    <input type="number" id="age" name="age">
</fieldset>
```

- Encapsulates related form fields under a common category.

9. `<legend>`

- The `<legend>` element provides a title for a `<fieldset>`.
- Syntax:

```
<legend>Caption</legend>
```

- Example:

```
<fieldset>
    <legend>Contact Details</legend>
    <label for="phone">Phone:</label>
    <input type="tel" id="phone" name="phone">
</fieldset>
```

- The legend text serves as a heading for grouped fields.

10. `<textarea>`

- The `<textarea>` tag is for multi-line text input, useful for feedback or comments.
- Syntax:

```
<textarea name="message"></textarea>
```

- Example:

```
<textarea name="feedback" rows="4" cols="50" placeholder="Enter your
feedback"></textarea>
```

- Creates a larger text box for user input.

11. `<button>`

- The `<button>` tag creates a clickable button, which can perform various actions.
- Syntax:

```
<button type="button">Click Me</button>
```

- Example:

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

- The submit button sends form data to the server.

`<input>` Attributes:

- The `<input>` tag supports various attributes that define its behavior and characteristics.

- Common Attributes:

- 1.type - Defines the type of input field (e.g., text, password, email, etc.).
- 2.id - Unique identifier for the input element.
- 3.name - Specifies the name of the input field, used when submitting form data.
- 4.value - Defines the default value of the input field.
- 5.list - Links the input field to a `<datalist>` for suggestions.
- 6.multiple - Allows selecting multiple values (used with file inputs and select elements).
- 7.readonly - Prevents the user from modifying the field.
- 8.disabled - Disables the input field.
- 9.required - Makes the field mandatory.
- 10.placeholder - Provides a hint to the user about the expected input.
- 11.min - Sets the minimum value (for number, date, etc.).
- 12.max - Sets the maximum value.
- 13.minlength - Specifies the minimum number of characters.
- 14.maxlength - Specifies the maximum number of characters.

- 15.step - Defines the interval for numeric input fields.
- 16.size - Specifies the width of the input field (in characters).
- 17.autofocus - Automatically focuses on the input field when the page loads.
- 18.autocomplete - Enables or disables browser auto-completion.
- 19.pattern - Defines a regular expression for validation.

Values of the type Attribute:

- The type attribute of <input> defines what kind of data the input field accepts.

Below are some common values:

- 1.text - A standard text input.
- 2.email - Ensures the user enters a valid email address.
- 3.password - Hides the input for security.
- 4.number - Allows only numeric values.
- 5.tel - Used for telephone numbers.
- 6.radio - Select one option from multiple choices.
- 7.checkbox - Select multiple options.
- 8.button - Creates a clickable button.
- 9.submit - Submits the form.
- 10.reset - Resets the form fields.
- 11.search - Provides a search field.
- 12.date - Allows selecting a date.
- 13.time - Allows selecting a time.
- 14.datetime-local - Allows selecting a date and time.
- 15.week - Selects a week.
- 16.month - Selects a month.
- 17.color - Opens a color picker.
- 18.image - Uses an image as a submit button.
- 19.file - Allows file uploads.
- 20.url - Ensures a valid URL format.
- 21.range - Allows selecting a value from a range.

HTML Input Types:

1. Text

- <input type="text" name="text">
- A standard text input field.

2. Email

- <input type="email" name="email">
- Ensures the user enters a valid email address.

3. Password

- <input type="password" name="password">
- Hides the input for security.

4. Number

- <input type="number" name="number">
- Allows only numeric values.

5. Telephone

- <input type="tel" name="tel">
- Used for telephone numbers.

6. Radio

- <input type="radio" name="option" value="1"> Option 1
- Select one option from multiple choices.

7. Checkbox

- <input type="checkbox" name="agree"> I Agree
- Select multiple options.

8. Button

- <input type="button" value="Click Me">
- Creates a clickable button.

9. Submit

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

- Submits the form.

10. Reset

- `<input type="reset" value="Reset">`
- Resets the form fields.

11. Search

- `<input type="search" name="search">`
- Provides a search field.

12. Date

- `<input type="date" name="dob">`
- Allows selecting a date.

13. Time

- `<input type="time" name="time">`
- Allows selecting a time.

14. Datetime-local

- `<input type="datetime-local" name="datetime">`
- Allows selecting a date and time.

15. Week

- `<input type="week" name="week">`
- Selects a week.

16. Month

- `<input type="month" name="month">`
- Selects a month.

17. Color

- `<input type="color" name="color">`
- Opens a color picker.

18. Image

- `<input type="image" src="submit.png" alt="Submit">`
- Uses an image as a submit button.

19. File

- `<input type="file" name="file">`
- Allows file uploads.

20. URL

- `<input type="url" name="website">`
- Ensures a valid URL format.

21. Range

- `<input type="range" name="range" min="0" max="100">`
- Allows selecting a value from a range.