BASICS WEBTECH

M18 Batch WebTech Notes

Web

A system of interconnected public webpages accessible through the Internet refers to the World Wide Web (WWW). It is a vast network of websites that users can access via web browsers like Chrome, Firefox, or Safari.
In simple wordsWeb is a collection of information in the form of images , audios , videos , Webpages , websites , Web Application

Basic Terminologies

Internet: A global network that connects millions of devices.

Web: Short for the **World Wide Web**, a collection of web pages and resources accessible through the internet.

Browser: A software application (e.g., Chrome, Firefox, Safari) used to access and display websites.

Server

A **server** is a computer or a system where all the websites are hosted. It handles the request of the user 24 x 7. For a valid request server will give the response and it contains logics and business logics.

Client: A device (e.g., your computer or smartphone) that requests web content from a server.

IP Address: A unique numerical address assigned to each device connected to the internet.

URL (Uniform Resource Locator): The web address used to locate a specific webpage (e.g., https://example.com).

DNS (Domain Name System): Translates domain names (like `google.com`) into IP addresses.

HTTP/HTTPS: Protocols used for transferring data between a client and server.

- *HTTPS* adds a layer of security through encryption.

Difference between a Webpage, Website And WebApplication

Webpage → A single page within a website that displays specific content.
There are two types of Webpages are there 1 static 2 Dynamic

Static →Static pages are those pages in which the information will be same for each and every User .

Dynamic→Dynamic pages are those pages in which the information will be different for each and every user .

Website → A collection of interconnected web pages that primarily serve to display information.

Web Application → A software program that runs in a web browser, allowing users to interact with features and perform tasks beyond just viewing content. It is more dynamic and interactive than a regular website.
And they are providing some services to the users.

There are two types of Web Applications 1. Single page 2. Multipage Application

Single Page Application

1. Contains single html file
2. It will not reload in every request .
3. Request will always continuous from the previous request

MultPage Application

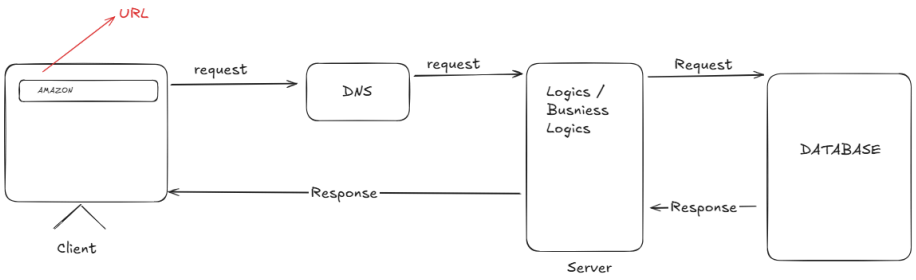
1. Contains multipages

2. It will reload in every Request made by user

Example →

- **Website:** A company's homepage with information about their products and services.
- **Webpage:** A specific page on that company website detailing a particular product.
- **Web application:** An online banking platform where users can view account balances, transfer funds, and pay bills.

How the Web Works?



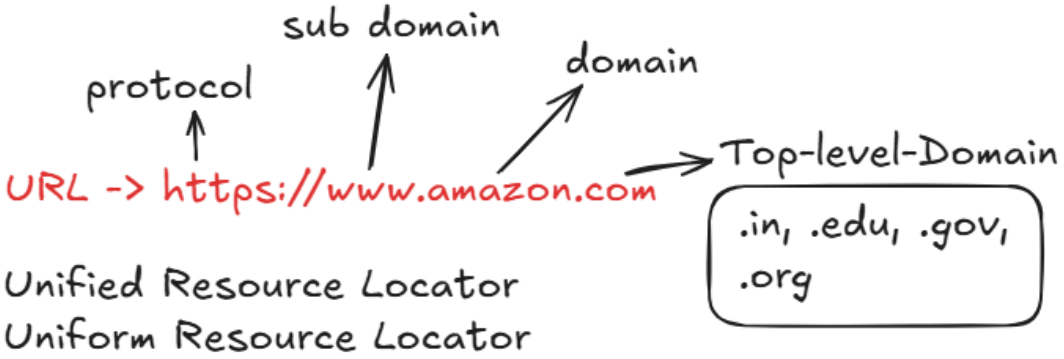
Steps

1. When you type a URL (website address) in the browser and press enter.

2. The browser sends a request to a DNS server (Domain Name System).
3. The DNS server translates the domain name (like www.google.com) into its corresponding IP address (a unique number assigned to a website).
4. Then, the DNS server sends back this IP address to the browser as a response.
5. Now, the browser sends an HTTP or HTTPS request to the server using this IP address.
6. The server processes the request and sends back a response to the browser. This response contains the website's files like HTML, CSS, and JavaScript.
7. The browser receives these files and displays the web page on the screen, allowing the user to see and interact with it.

What is the URL?

- It stands for **Uniform Resource Locator** or **Unified Resource Locator**.
- It is the address of the unique resource present on the internet



How Browsers understand HTML, CSS, JS?

A browser is not just a simple software—it has different components that work together to display web pages.

1. **Rendering Engine** : Responsible for understanding **HTML** and **CSS**.

What is HTML, CSS, JS?

HTML

- Html stands for Hyper Text Markup Language.
- It is used to structure a webpage.
- It also used to add content to the webpage.

CSS

- CSS stands for Cascading Style Sheets
- It is used to style and add alignment to the web page.

JavaScript

- It is used to add some functionality to our web pages

HTML

M18 Batch WebTech Notes

HTML

HTML (HyperText Markup Language) is the most basic building block of the Web. It is used to Create the structure of a webpage and along with that it will help out to add the content in webpage .
Content like → images , audios , text , videos etc

History Of HTML

- **1991**: Tim Berners-Lee invented HTML 1.0
- **1993**: HTML 1.0 was released
- **1995**: HTML 2.0 was published, which added new features to HTML 1.0
- **1997**: HTML 3.0 was invented, which improved HTML features and gave webmasters more capabilities
- **1999**: HTML 4.01 was released and became the official standard
- **2014**: HTML 5.0 was released and used worldwide

Structure of HTML

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

Doctype html:

It is used to declare the version of html as html5.

Html:

It is the root element which contains the head section and body section.
It is the container for all other HTML elements

Head

It holds the meta data about the html document. such as title, styles, and scripts

Title:

It will provide the title to the webpage.

Body:

This is the place where we have to write all the code which has to be displayed on UI.

Tags

HTML Tags are predefined keywords that are enclosed in angular brackets (< >). Each tag has a specific predefined task or purpose

Syntax:

<tagname> content </tagname>

Ex:
 Hello world
 Hello world

Types of Tags:

HTML tags were categorized into 2 ways.

They are:

1. Paired Tags
2. UnPaired Tags

Paired Tags:

A tag which is required of both opening and closing tags are called paired tags. Ex: Hello Web Dev

Unpaired Tags:

A Tag which does not require a closing tag is called unpaired tag.

Ex:
 , <hr> , <input>

Element

An element is a combination of opening tag + content + closing tag.

Html Elements were classified into 2 ways

1. Inline Level
2. Block Level
3. Inline-block Level

Inline Level:

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:

<b> , <i> , , etc…

Block Level:

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 elements.

Ex: <h1> , , <p> , etc…

Inline-Block-Level

Inline level Elements are those elements which are having both the behaviour block and inline level elements , They will take content height and content width only along with that they will accept height and width

Ex: <img> <button> <input>

Heading Tags

Heading Tags are basically used to provide title or subtitle to the content in the webpage.

We have a set of heading tags from h1 to h6.

Where h1 is highest in font size and h6 is lowest in font 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

Formatting Tags

Formatting Tags are used to display the content in different formats.

Formatting Tags are:

b → It will display the content in bold format.

strong → It will display the content in bold format and the browser will understand that information is very important.

i → It will display the content in Italic format.

em → It will display the content in Italic format and it is an alternative to the <i> tag.

u → It will provide an underline to the content.

ins → It will provide an underline to the content and it is an alternative to <u> tag.

strike → It will strike off the content.

del → It will strike off the content and it is the alternative to <strike> tag.

q → It will provide quotations to the content.

mark → It will highlight the content with yellow background color.

big → It will display the content with a big font-size.

small → It will display the content with a small font-size

sup → It will display the content as superscript.

sub → It will display the content as subscript.

code → It will display in the font-family "monospace".

pre → It will consider space also. It will also be known as a pre structured tag.

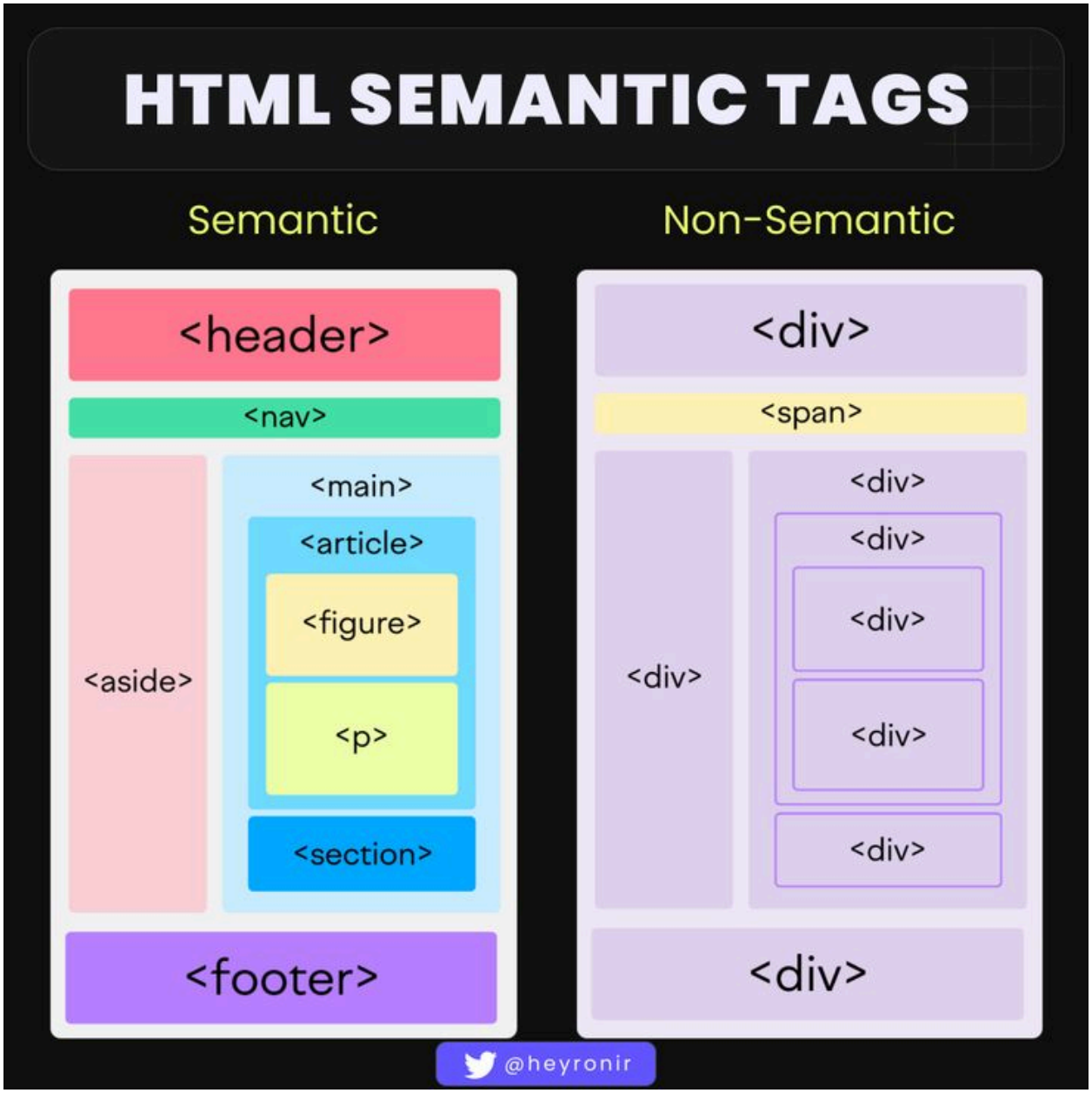
Deprecated Tags:

These are the tags which will be removed in further versions.

Vs code gives suggestions by displaying in red color.

Semantic tags

Semantic tags are those tags which will describe their meaning by their name only .



Examples

Main Document Structure

- `<header>` – Introductory content or navigation (top section).
- `<nav>` – Navigation links.
- `<main>` – Main content of the document (unique content).
- `<section>` – A thematic grouping of content.
- `<article>` – Self-contained content (e.g., blog post, news article).
- `<aside>` – Sidebar or related content.
- `<footer>` – Footer section (end of page or section).

Text Content

- `<h1> ... <h6>` – Headings.
- `<p>` – Paragraph.
- `<blockquote>` – Long quotation.
- `<q>` – Inline short quotation.
- `<address>` – Contact information.
- `<time>` – Date or time.
- `<mark>` – Highlighted text.
- `<strong>` – Important text.
- `<em>` – Emphasized text (usually italic).

Etc.. try to find out by yourself for better understanding with it .

Attributes

HTML attributes are used to define the characteristics of an HTML element.

- Attributes provide additional information about elements
- Attributes are always used in the opening tag of the element
- Attributes usually come in name/value pairs like: name="value"
- Attributes must be written in lower case

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

Types of Attributes

1.Core Attributes

These are basic attributes that can be applied to most HTML elements.

- **Style**: Used to add styles to an element, such as color, font, and size.
- **Title**: Provides additional information about an element and is often displayed as a tooltip when the mouse hovers over it.
- **Class**:`class` **attribute** is used to group multiple items together. It allows you to assign the same class name to multiple elements, making it easier to apply the same styles or manipulate them collectively using CSS or JavaScript.

  Unlike the ID attribute, the class attribute is not unique, and multiple elements can share the same class.

- **Id** The ID attribute is used to assign a unique identifier to an HTML element. The id should be unique for each element and the id attribute should have only one value

2.Internationalization Attributes:

These attributes help adapt the document to different languages and regions. Examples include `lang` and `dir`.

- lang

- dir

3.Generic Attributes

These are the element specific attributes .

- src
- href
- alt
- height and width

4.Global Attributes

- contenteditable
- tabindex
- Accesskey

1. `contenteditable`

The contenteditable attribute specifies whether the content of an element is editable by the user.

Ex →

```
<div contenteditable="true">You can edit this text!</div>
<div contenteditable="false">This text is not editable.</div>
```

3. `tabindex`

The **`tabindex`** attribute specifies the tab order of an element when navigating through the page using the `Tab` key. It controls the sequence of focusable elements.

- **`0`**: The element is focusable in the natural tab order.
- **Positive Numbers**: The element is focusable and has a custom tab order. Lower numbers are focused first.
- **Negative Numbers**: The element is not focusable via `Tab`, but can be focused programmatically.

4.accesskey

The `accesskey` attribute specifies a shortcut key (keyboard combination) that can be used to activate or focus a specific HTML element, such as a link, button, or form field.

Lists

HTML Lists allows to group a set of items together.

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.

Ordered List is also known as Number List.

- **Tags:**

ol → It indicates that we are creating an ordered List.

li → It indicates List Items.

- **Attributes:**

Type → Specifies the type of sequence.

Values = 1 (default) , A , a , I , i

Start → Specifies the Starting Position.

Values = 1 (default) , Any Number

Reversed → It will reverse the sequence.

2. Unordered List:

Unordered List is used to display a set of items with special symbols.

Unordered List is also known as Bulleted List.

- **Tags:**

  ul → It indicates that we are creating an Unordered List.

  li → It indicates List Items.

- **Attributes:**

  - 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.

- **Tags:**

  dl → Indicates Description List

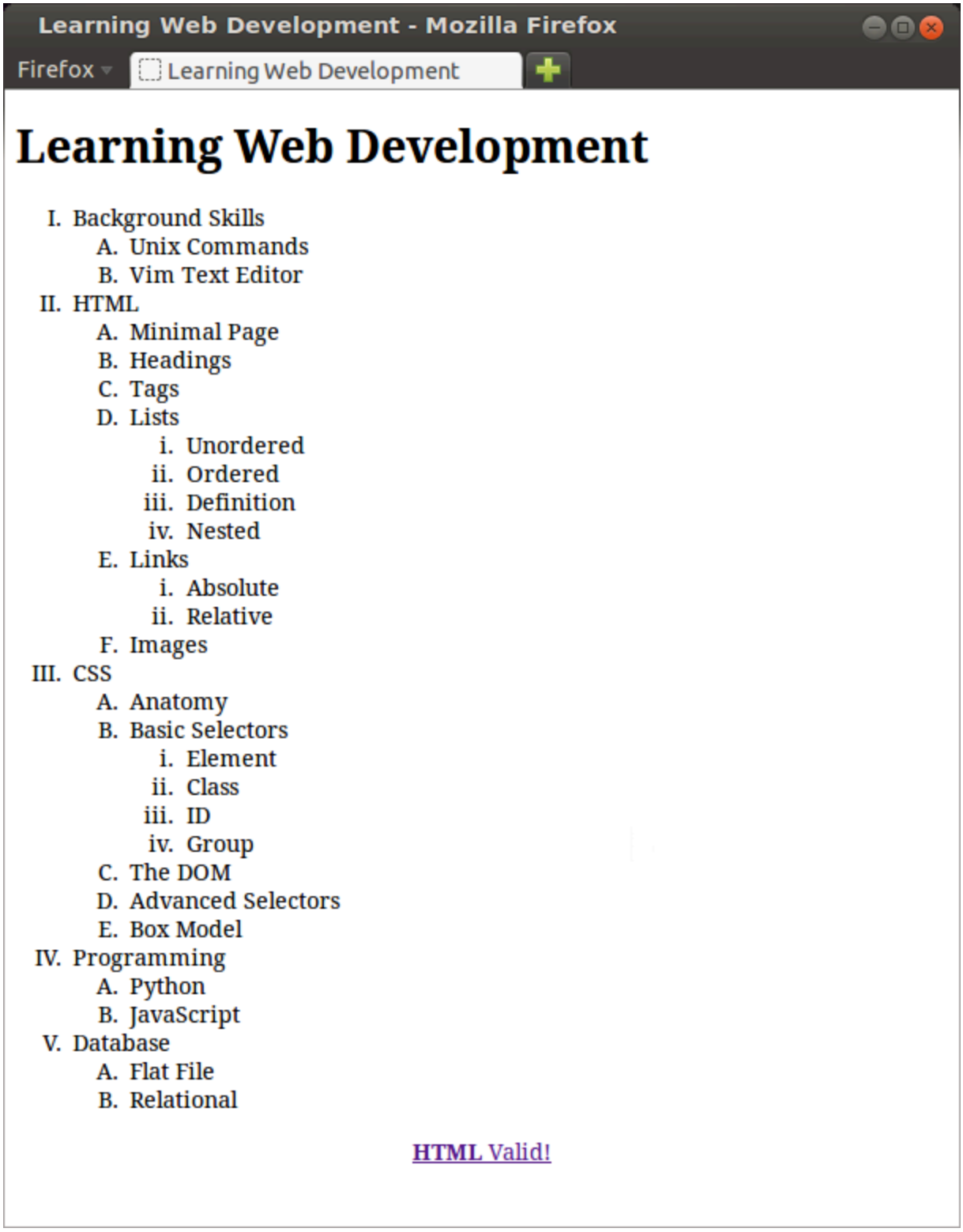
  dt → Indicates Description term

  dd → Indicates Description Definition.

3. Nested List :

- List inside another list is nothing but the Nested List , it can be Oredered , unordered , Description .

Examples for List



Learning Web Development

I. Background Skills
    A. Unix Commands
    B. Vim Text Editor
II. HTML
    A. Minimal Page
    B. Headings
    C. Tags
    D. Lists
        i. Unordered
        ii. Ordered
        iii. Definition
        iv. Nested
    E. Links
        i. Absolute
        ii. Relative
    F. Images
III. CSS
    A. Anatomy
    B. Basic Selectors
        i. Element
        ii. Class
        iii. ID
        iv. Group
    C. The DOM
    D. Advanced Selectors
    E. Box Model
IV. Programming
    A. Python
    B. JavaScript
V. Database
    A. Flat File
    B. Relational

HTML Valid!

Table

Table in html is used to display the data in the structured and organized manner on the ui .
In the form of rows and columns .
Whatever will be displayed on the ui it will be inside the cell . And the cell is nothing but the intersection of rows and columns .

Tags : → <Table>, <tr> , <td> , <th> , <thead> , <tbody> , <tfoot>
Attributes → cellspacing , cellpadding , border , rules , rowspan , colspan , dir

Important Table Tags

Tag	Description
`<table>`	Main container to create a table.
`<tr>`	Table row – defines a row in the table.
`<td>`	Table data – represents a normal cell inside a row.
`<th>`	Table header – similar to `<td>` but makes text **bold & centered**.
`<thead>`	Groups the header content of the table.
`<tbody>`	Groups the body (main data) of the table.
`<tfoot>`	Groups the footer content of the table (like totals).

Important Table Attributes

Attribute	Description
`border`	Adds a border around the table and cells.
`cellspacing`	Space **between** cells.

`cellpadding`	Space **inside** a cell (padding).
`rules`	Defines internal borders (rows, cols, or all).
`rowspan`	Makes a cell span across multiple rows.
`colspan`	Makes a cell span across multiple columns.
`dir`	Defines text direction (`ltr` → left to right, `rtl` → right to left).

Task on table

A test table with merged cells

	Average		Red eyes
	height	weight	
Males	1.9	0.003	40%
Females	1.7	0.002	43%

Athletics day time table					
Area	**10am-11am**	**11am-12pm**	**12pm-1pm**	**1pm-2pm**	**2pm-3pm**
Track	400m	200m	**Lunch**	800m	100m
	Hurdles	1500m		Free	80m
Field	Discus	Javelin		Shot put	Hammer throw

Image

- `<img>` tag is used to **display images** on a webpage.
- It is an **empty tag** (no closing tag).

Attributes of `<img>`

Attribute	Description
`src`	Specifies the **path (location)** of the image file.
`alt`	Alternative text – shown when image is not loaded, also used by screen readers (for accessibility).

Src attribute will accept two types of path 1. Relative Path , 2. Absolute Path

1. **Relative path**
   This is a path having some relation with my current file or folder and it will start with . and . represents coming out from any file or folder , / it represents move inside amny file or folder .

2. **Absolute Path**
   This is the Path which is not having any relation with my current file or folder .

Form

HTML form is a container element (`<form>`) used to collect user input and send it to a server for processing

Syntax

```
<form action="">

  <!-- Form elements go here -->

</form>
```

Tags Used in Forms

1. `<form>`
2. `<label>`
3. `<input>`
4. `<select>`
5. `<optgroup>`
6. `<option>`
7. `<datalist>`
8. `<fieldset>`
9. `<legend>`
10. `<textarea>`
11. `<button>`

Explanation of Form Elements

1. `<form>`

- The `<form>` tag is a **semantic element** used to contain form elements.

2. `<label>`

- Specifies the purpose of an input field.
- Improves accessibility by linking with an `<input>` field using the `for` and `id` attributes.

Example: Label and Input

```
<label for="userName">Name:</label>

<input type="text" id="userName">


<label for="userEmail">Email:</label>
```

```
<input type="email" id="userEmail">
```

3. `<select>`

- Creates a **dropdown list** with a set of options.

4. `<optgroup>`

- Groups related `<option>` elements inside a `<select>`.

5. `<option>`

- Defines individual options inside a dropdown.

Example: Select, Optgroup, and Option

```
<label>Skills:</label>
<select name="userSkills" multiple>
  <optgroup label="Frontend">
    <option value="React">React</option>
    <option value="Angular">Angular</option>
  </optgroup>
  <optgroup label="Database">
    <option value="SQL">SQL</option>
    <option value="MongoDB">MongoDB</option>
  </optgroup>
  <optgroup label="Backend">
    <option value="Python">Python</option>
    <option value="Java">Java</option>
```

```
    <option value="NodeJs">NodeJs</option>
  </optgroup>
</select>
```

6. <datalist>

- Similar to <select>, but allows users to filter options by typing.
- Requires an <input> field.
- The value attribute and content should be the same.

Example: Datalist

```
<label>Skills:</label>
<input list="skillsList">
<datalist id="skillsList">
  <option value="React">React</option>
  <option value="Angular">Angular</option>
  <option value="SQL">SQL</option>
  <option value="MongoDB">MongoDB</option>
  <option value="Python">Python</option>
  <option value="Java">Java</option>
  <option value="NodeJs">NodeJs</option>
</datalist>
```

7. <fieldset>

- Groups related form fields and displays a **border** around them.

8. `<legend>`

- Provides a **title** for a `<fieldset>`.

Example: Fieldset and Legend

```
<fieldset>
  <legend>Personal Details</legend>
  <!-- Form elements go here -->
</fieldset>
```

9. `<textarea>`

- Used for **multiline text input**.
- `rows` and `cols` specify the visible height and width.

Example: Textarea

```
<textarea name="comment" rows="10" cols="30"></textarea>
```

10. `<button>`

- Creates a **button** with different types (e.g., submit, reset).

Example: Button

```
<button type="reset">Reset</button>
<button type="submit">Submit</button>
```

Attributes of `<input>` Tag

1. `type` – Specifies the input type (e.g., text, email, password).
2. `id` – Unique identifier for the input field.
3. `name` – It is used to retrieve specific form data after submission
4. `value` – Predefined value in the input field.
5. `list` – Links to a `<datalist>`.
6. `multiple` – Allows multiple selections.
7. `readonly` – Prevents the user from editing the field.
8. `disabled` – Disables the field.
9. `required` – Makes the field mandatory.
10. `placeholder` – Displays hint text inside the field.
11. `min` / `max` – Defines minimum and maximum values.
12. `minlength` / `maxlength` – Defines min/max length for text input.
13. `step` – Specifies increments for number/date inputs.
14. `size` – Defines the width of the input field.
15. `autofocus` – Automatically focuses the input field.
16. `autocomplete` – Suggests previously entered values.
17. `pattern` – Defines a regex pattern for validation.

Values of the `type` Attribute

1. `text`
2. `email`
3. `password`
4. `number`
5. `tel`
6. `radio`
7. `checkbox`
8. `button`
9. `submit`
10. `reset`
11. `search`
12. `date`
13. `time`
14. `datetime-local`
15. `week`
16. `month`

17. `color`
18. `image`
19. `file`
20. `url`
21. `range`

Attributes of the `<form>` Tag

1. **`action`**
    - Specifies the URL where the form data should be sent.
2. **`method`**
    - Defines how data is sent:
        - `GET` – Data is sent in the URL, **as a Query Parameter**.
        - `POST` – Data is sent in the request body, **secure**.
3. **`novalidate`**
    - Prevents form validation before submission.
4. **`autocomplete`**
    - Controls autofill behavior (`on` / `off`).

FORM Code

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

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

    <title>Document</title>

  </head>

  <body>
```

```
<h1>Job Application</h1>


<form action="" method="post">
  <fieldset>
    <legend>Basic Details</legend>
    <div>
      <label for="name">Name </label>
      <input
        type="text"
        id="name"
        placeholder="enter name"
        name="name"
        minlength="2"
        autofocus
      />
    </div>


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


    <div>
```

```
          <label for="contact">Contact</label>

          <input type="tel" id="contact" name="contact"
pattern="[0-9]{10}" />

        </div>



        <div>

          <label for="dob">Dob</label>

          <input

            type="date"

            id="dob"

            name="dob"

            min="1990-01-01"

            max="2007-12-31"

          />

        </div>



        <div>

          <label for="gender">Gender</label>

          Male<input type="radio" id="gender" name="gender" value="male"
/>

          Female<input type="radio" id="gender" name="gender"
value="female" />

          Others<input type="radio" id="gender" name="gender"
value="others" />

        </div>
```

```
        <div>
          <label for="skills">Skills</label>
          HTML<input type="checkbox" name="skills" value="html" />
CSS<input
            type="checkbox"
            name="skills"
            value="css"
          />
          JS<input type="checkbox" name="skills" value="js" />
        </div>
      </fieldset>

      <fieldset>
        <legend>Educational Details</legend>
        <div>
          <label for="highestDegree">Highest Degree</label>
          <select name="Highest Degree" id="">
            <option value="" disabled>Select degree</option>
            <option value="B.tech">B.tech</option>
            <option value="BCA">BCA</option>
            <option value="MBA">MBA</option>
            <option value="B PHARMA">B PHARMA</option>
            <option value="">MCA</option>
```

```
        </select>
      </div>

      <div>
        <label for="yop">Yop</label>
        <input type="number" id="yop" name="Yop" min="2020" max="2025"
/>
      </div>
      <div>
        <label for="percentage">Percentage</label>
        <input type="number" name="percentage" min="33" max="99" />
      </div>

      <div>
        <label for="univeristy">univeristy</label>
        <select name="univeristy" id="">
          <option value="">Select univeristy</option>
          <option value="">A univeristy</option>
          <option value="">B univeristy</option>
          <option value="">C univeristy</option>
          <option value="">M univeristy</option>
        </select>
      </div>
```

```
    <div>
      <label for="backlogs">No. Backlogs</label>
      <input type="number" name="backlogs" min="0" max="3" />
    </div>
  </fieldset>
  <fieldset>
    <legend>Address Details</legend>

    <div>
      <label for="city">City</label>
      <input
        type="text"
        list="citydata"
        placeholder="Select city"
        name="city"
      />
      <datalist id="citydata">
        <option value="Delhi"></option>
        <option value="Chandigarh"></option>
        <option value="Noida"></option>
        <option value="Sonipat"></option>
        <option value="gaya"></option>
        <option value="Patna"></option>
```

```
    <option value="muzaffarnagar"></option>
  </datalist>
</div>

<div>
  <label for="State">State</label>
  <input type="text" list="stateList" name="State" />

  <datalist id="stateList">
    <option value="Delhi"></option>
    <option value="Haryana"></option>
    <option value="Up"></option>
    <option value="Uttrakhand"></option>
    <option value="Jharkhand"></option>
    <option value="sikkim"></option>
    <option value="Bihar"></option>
  </datalist>
</div>

<div>
  <label for="">Pincode</label>
  <input type="tel" name="pincode" pattern="[0-9]{6}" />
</div>
```

```
  <div>
    <label for="">Complete Address</label>
    <textarea name="Complete Address" id=""></textarea>
  </div>
</fieldset>
<fieldset>
  <legend>Others Details</legend>

  <div>
    <label for="">Salary Expectations</label>
    <input type="range" name="Salary" min="0" value="5" max="20" />
  </div>

  <div>
    <label for="">CV</label>
    <input type="file" name="cv" />
  </div>

  <div>

    <label for="">Tshirt color</label>
```

```
  <input type="color" name="tshirt color" />
</div>
<div>
  <label for="">Select cab</label>
  <select name="Cab" id="">
    <option value="">Select Cab</option>
    <optgroup label="Honda">
      <option value="Honda City">Honda City</option>
      <option value="Honda civic">Honda civic</option>
      <option value="Honda Amaze">Honda Amaze</option>
    </optgroup>
    <optgroup label="Tata">
      <option value="">Nano</option>
      <option value="">Harrier</option>
      <option value="">Nexon</option>
    </optgroup>
    <optgroup label="Mahindra">
      <option value="">Scorpio</option>
      <option value="">Xuv 700</option>
      <option value="">Be 6</option>
    </optgroup>
  </select>
</div>
```

```
        </fieldset>


        <button>Submit</button>
    </form>
  </body>
</html>
```

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:

 content

Whatever we give as the content to the anchor tag that becomes a hyperlink.

Content can be text, image, button, icon, etc.

To create the hyperlink , href attribute is mandatory

To mail a person we have to use mailto protocol ("mailto:example@gmail.com")

Email Us

To call a person we have to use tel protocol ("tel:+911234567890")

Attributes:

1. href → It will specify the path address to where we have to navigate.
2. download → It indicates to download the file.

Download File

3. Target → it specify where to open the linked document

(_blank , _parent , _top , _self)

Default colors:

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

```
<!-- Usage -->
<!-- 1.Link a web page -->
<!-- 2.Link to a section of the same page / another page -->
<!-- 3. Send Email -->
<!-- 4.Link to phone caller-->
<!-- 5. Link to a script -->
<!-- 6.Download a file-->
<!-- 7.Opening the page into new window or new tab -->
<!--8. target attribute -->
<!-- 9. four states of the anchor tag →
<!--10.access key attribute-->
<center>
```

Anchor Zip code below

Anchor.zip

1 Files • 6.33KB

Audio Tag

The <audio> tag is used to include audio files on a webpage.

Syntax: <audio src="path address"></audio>

Attributes:

1. **src** → It will specify the path address
2. **controls** →Displays play, pause, and volume controls for the user.
3. **autoplay** → It will play the song automatically when the page loads
4. **loop** → Makes the audio replay continuously after finishing.
5. **muted** → Starts the audio with the volume muted.

Video Tag

The <video> tag is used to include video files on a webpage.

Syntax: <video src="path address"></video>

Attributes:

1. **src** → It will specify the path address
2. **controls** →Displays play, pause, volume, and fullscreen controls for the user
3. **autoplay** →Automatically starts playing the video when the page loads (should include `muted` for autoplay).
4. **loop** → Makes the video replay continuously after finishing.
5. **muted** → Starts the video without sound.
6. **Poster** → Specifies an image to display before the video starts playing.(thumbnail)

Marquee Tag

Marquee tag implements scrollable content in our webpages.

Syntax:

```
<marquee behavior="" direction=""> Content </marquee>
```

Whatever we provide in the content place will scroll, content like text / images / anchor tag / icons / buttons etc…

Attributes:

direction → It specifies the direction of scrolling content.

Values = left (default) / right / up / down

Behavior → It specifies the behavior of marquee element.

Values = scroll (default) / slide / alternate

Scrollamount → It specifies the speed of an element

Values = Numbers (default is 6)

Loop → It specifies how many times the content has to scroll.

Height → It Specifies the height of an element.

Width → It Specifies the width of an element.

Audio Tag

The `<audio>` tag is used to include audio files on a webpage.

Syntax: `<audio src="path address"></audio>`

Attributes:

1. **src** → It will specify the path address
2. **controls** →Displays play, pause, and volume controls for the user.
3. **autoplay** → It will play the song automatically when the page loads
4. **loop** → Makes the audio replay continuously after finishing.
5. **muted** → Starts the audio with the volume muted.

Video Tag

The `<video>` tag is used to include video files on a webpage.

Syntax: `<video src="path address"></video>`

Attributes:

1. **src** → It will specify the path address
2. **controls** →Displays play, pause, volume, and fullscreen controls for the user
3. **autoplay** →Automatically starts playing the video when the page loads (should include `muted` for autoplay).
4. **loop** → Makes the video replay continuously after finishing.
5. **muted** → Starts the video without sound.
6. **Poster** → Specifies an image to display before the video starts playing.(thumbnail)
7. **`width` and `height`**: Define the dimensions of the video player.

Iframe Tag

Iframe tag is used to embed another web page into the current web page. We can embed internal web pages as well as external web pages like youtube , google maps etc…

Syntax : `<iframe src="" frameborder="0" height="" width=""></iframe>`

Attributes:

1. **src** → It will specify the path address
2. **frameborer**→Specifies whether the iframe should have a border (`0` for no border, deprecated in HTML5).
7. **`width` and `height`**: Define the dimensions of the iframe.

Embedding Map

Steps to Embed a Map

1. Open Google Maps
2. Get the Embed Code
3. Add the Code to Your Webpage

Generating QR Code

You can use free APIs like **Google Chart API** or **QR Code Generator APIs** to dynamically create QR codes.

1. Go to goqr website and get the api (qr code generator api)
   https://goqr.me/api/
   https://api.qrserver.com/v1/create-qr-code/?size=150x150&data=Example
2. Replace the Example with the your website path address
3. Use img tag and pass the complete path to src
4. You can change the qr size also by using height and width attribute or passing values to the size in path address

HTML Entities

Some characters are reserved in HTML.

If you use the less than (<) or greater than (>) signs in your HTML text, the browser might mix them with tags.

Entity names or entity numbers can be used to display reserved HTML characters.

Example →

To display a less than sign (<) we must write: < or <

CSS

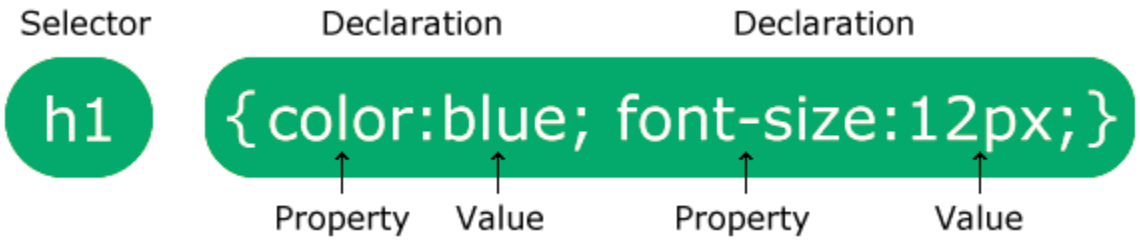
CSS

CSS stands for Cascading Style Sheets
CSS describes how elements in a web page should be displayed on screen
CSS is used to style and provide layout to the webpage

CSS History

1994
Håkon Wium Lie, a Norwegian technologist at CERN, proposed the idea of CSS after noticing that there was no way to style documents on the web.
1996
The first version of CSS, CSS1, was adopted by the World Wide Web Consortium (W3C).
1998
CSS 2 was released, and work on CSS 3 began.
2011
CSS 3 was released

CSS Syntax



The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

Ways to add CSS

There are three ways to add CSS .

1. Inline CSS
2. Internal CSS
3. External CSS

1. Inline CSS

An inline style is used to apply a unique style for a single element.

To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

```
<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>
```

2. Internal CSS

It is a way of adding the CSS in the same html file itself. By using Style Tag.
We can use style tag anywhere in the html file. It's recommended to use in a head tag.

Ex :

```
<html>
 <head>
```

```
    <title>Document</title>
    <style>
      h1 {
        color: white;
        background-color: blue;
      }
    </style>
  </head>
  <body>
    <h1>Let's dive into CSS</h1>
  </body>
</html>
```

3. External CSS

It is a way of writing all the css code in one file which is saved with .css extension.
That css file you can link with multiple html files by using the link tag inside the <head> section
It makes the stylesheet reusable

```
<!-- index.html →
<html>
  <head>
    <title>Document</title>
    <link rel="stylesheet" href="./style.css" />
  </head>
  <body>
    <h1>Let's dive into CSS</h1>
  </body>
</html>
/* style.css */
h1 { color: white;
  background-color: blue;
 }
```

Note :

1. The first will always be taken by inline CSS. because of the most recent update.
2. Internal vs External depends on code flow which is written at last.