



# HTML

---

## 🌐 Web Development — Level 0: Setup & Fundamentals

---

### 🧱 Section 1: IDE (Integrated Development Environment)

#### 💡 Definition

An **IDE (Integrated Development Environment)** is software used by developers to write, edit, run, and debug code efficiently.

---

#### ✓ Key Benefits

- ⚡ Improves developer efficiency
  - 🧠 Code autocomplete support
  - 🔗 Version control integration (Git)
  - 🐞 Error detection and debugging tools
- 

#### 📦 Popular IDEs & Code Editors

IDE / Editor	Description
VS Code	Most popular, lightweight, powerful
Atom	Customizable editor (less used now)
Sublime Text	Fast and minimal
GitHub Codespaces	Cloud-based development environment

---



## Recommended VS Code Extensions

Extension	Purpose
Live Server	Run HTML files instantly in browser
Prettier	Auto-formats code



## Section 2: Client Side vs Server Side



### Concept Overview

Feature	Client Side	Server Side
Execution Location	User's browser	Remote server
Technologies	HTML, CSS, JavaScript	Node.js, Python, Java, PHP
Data Access	Cannot access database directly	Can access database and files
Speed	Faster UI interactions	Slower due to network



## Request Flow

User → Browser → Server → Database → Server → Browser → User



## Section 3: Role of Browser



### Responsibilities

- 💻 Displays web pages (renders HTML)
- 🎮 Enables user interaction
- ⚡ Executes JavaScript
- 📦 Loads resources (CSS, JS, Images)

## Section 4: What is HTML?

### Definition

HTML (HyperText Markup Language) is used to create and structure web pages.

---

### Purpose

HTML defines:

- Headings
  - Paragraphs
  - Buttons
  - Images
  - Links
- 

### Real-World Analogy

HTML = Skeleton 

It provides structure to the web page.

---

## Section 5: What is CSS?

### Definition

CSS (Cascading Style Sheets) is used to style HTML elements.

---

### Used for

- Colors 
- Layout 
- Fonts 

- Spacing 
  - Animations 
- 

## Real-World Analogy

CSS = Appearance 

It makes the skeleton look attractive.

---

## Section 6: What is JavaScript?

### Definition

JavaScript adds logic and interaction to web pages.

---

## Used for

- Button clicks
  - Form validation
  - Dynamic updates
  - API calls
- 

## Real-World Analogy

JavaScript = Brain 

It controls behavior.

---

## Section 7: HTML vs HTM

Feature	.html	.htm
Usage	Most common	Less common
Browser support	Full support	Full support

Feature	.html	.htm
Recommended	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No



## Section 8: Importance of index.html

### Definition

`index.html` is the default homepage.

### Example

```
https://example.com
```

Automatically loads:

```
https://example.com/index.html
```



### Folder Structure

```
root/
|
├── index.html
├── about.html
└── contact.html
```



### Importance

- Default entry point
- Required for websites
- Important for SEO

# Section 9: SEO (Search Engine Optimization)

## Definition

SEO improves website visibility in search engines.

---

## Benefits

- More visitors
  - Better ranking
  - Higher visibility
- 

# Section 10: How to Implement SEO

## 1. Proper HTML Structure

```
<h1>Main Heading</h1><p>Content</p>
```

---

## 2. Title Tag

```
<title>Learn Web Development</title>
```

---

## 3. Meta Description

```
<meta name="description" content="Learn HTML, CSS, JavaScript">
```

---

## 4. Proper Headings

```
<h1>Main Title</h1><h2>Section</h2><h3>Subsection</h3>
```

## ✓ 5. Use index.html

Default homepage file.

## ✓ 6. Alt Text for Images

```

```

## ✓ 7. Clean URLs

Good:

```
example.com/html-tutorial
```

Bad:

```
example.com/page?id=123
```

## 📎 Section 11: HTML Tags

### 💡 Definition

Tags define elements in HTML.

## Example

```
<h1>Hello World</h1>
```

Tag	Purpose
<h1>	Heading
<p>	Paragraph
<img>	Image



## Section 12: Basic HTML Page Structure

```
<!DOCTYPE html><html lang="en"><head><title>My First Webpage</title></head><body><h1>Hello World!</h1></body></html>
```



### Explanation

Tag	Purpose
<!DOCTYPE html>	Defines HTML5
<html>	Root element
<head>	Metadata
<title>	Browser tab title
<body>	Visible content



### Quick Summary

Technology	Role
HTML	Structure
CSS	Design
JavaScript	Behavior
Browser	Executes and renders
Server	Provides data
index.html	Entry point
SEO	Improves visibility

## 2.4 MDN Documentation

### Visit Official Website

Visit the MDN official website:

<https://developer.mozilla.org/>

### What is MDN?

MDN (Mozilla Developer Network) is the **official and most trusted resource for web development**, especially for HTML, CSS, and JavaScript.

---

## 2.5 Comments in HTML

### What are Comments?

HTML comments are used to add **notes or explanations inside the code**. These notes are only visible to developers and are **not displayed on the web page**.

#### 1. Used to Add Notes in HTML Code

Comments help developers explain the purpose of code.

Example:

```
<!-- This is a heading --><h1>Hello World</h1>
```

#### 2. Not Displayed on the Web Page

Comments are ignored by the browser.

They are only visible in the source code.

Users cannot see comments on the webpage.

---

## 2.6 Case Sensitivity in HTML



## Definition

HTML is **case-insensitive**, which means tag names and attribute names work the same whether written in uppercase or lowercase.

## ★ Key Points

### 1. HTML Tag Names are Case-Insensitive

This means the browser treats uppercase and lowercase tags the same.

Example:

```
<html></html>
```

is the same as

```
<HTML></HTML>
```

Both work correctly.

### 2. Attribute Names are Also Case-Insensitive

Attributes can also be written in uppercase or lowercase.

Example:

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

is the same as

```
<INPUTTYPE="text">
```

Both are valid.



## 1.1 What are HTML Attributes?



## Definition

HTML attributes provide **additional information about HTML elements**. They modify the behavior or appearance of elements.

Attributes are always written inside the **opening tag**.

### Example:

```
<a href="https://google.com">Visit Google</a>
```

Here:

- `<a>` → Tag
- `href` → Attribute
- `"https://google.com"` → Value

### 1. Provides Additional Information

Attributes describe or modify HTML elements.

Example:

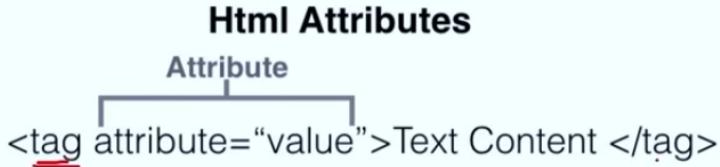
```

```

`src` tells the browser which image to load.

### 3. Common Examples of Attributes

Attribute	Purpose	Example
href	Link URL	<code>&lt;a href="link"&gt;</code>
src	Image source	<code>&lt;img src="image.jpg"&gt;</code>
alt	Image description	<code>&lt;img alt="Image"&gt;</code>
class	CSS styling	<code>&lt;div class="box"&gt;</code>
id	Unique identifier	<code>&lt;div id="header"&gt;</code>



## 1.2 id Property in HTML

### Definition

The **id attribute** is used to uniquely identify an HTML element within a webpage.

Each `id` must be **unique**, meaning it can only be used once per page.

### Basic Syntax

```
<tagid="uniqueName">Content</tag>
```

#### Example:

```
<divid="header">This is the header</div>
```

Here:

- `div` → HTML element
- `id` → Attribute
- `header` → Unique identifier

### 1. Unique Identifier

Each id must be unique in a webpage.

Correct example:

```
<divid="header"></div><divid="footer"></div>
```

Incorrect example (Do NOT do this):

```
<div id="header"></div><div id="header"></div><!-- X Wrong -->
```



## 2.1 Heading Tag in HTML



### Definition

Heading tags are used to define **titles and headings** in a webpage.



### Important Points

- `<h1>` → Most important (largest)
- `<h6>` → Least important (smallest)
- Used to **structure content**
- Important for **SEO**
- Helps browser and search engines understand page hierarchy



### Example

```
<h1>Main Title</h1><h2>Section Title</h2><h3>Subsection Title</h3>
```



## 2.2 Paragraph Tag in HTML



### Definition

The `<p>` tag is used to define **paragraphs** in a webpage.



### Syntax

```
<p>This is a paragraph</p>
```

## ★ Important Points

- Used to write **paragraph text**
- Always written inside `<p>` and `</p>`
- Adds **automatic spacing** before and after
- Text automatically moves to the next line
- Used for **text content**



## 2.4 Image Tag in HTML



### Definition

The `<img>` tag is used to **display images** on a webpage.



### Syntax

```

```

## ★ Important Points

- Used to **embed images**
- `src` → Specifies image path
- `alt` → Alternative text (important for SEO)
- `width` and `height` → Resize image
- Self-closing tag (no closing tag)

## Example

```

```

```

```



## 2.5 Video Tag in HTML

### Definition

The `<video>` tag is used to **display videos** on a webpage.

### Syntax

```
<video src="video.mp4" controls></video>
```

## ★ Important Points

- Used to **embed videos**
- `src` → Video URL/path
- `controls` → Shows play, pause, volume
- `autoplay` → Plays automatically
- `loop` → Repeats video
- Supports formats: MP4, WebM

```
<video width="400" height="250" controls>
    <source src="https://www.w3schools.com/html/mov_bbb.mp4" type="video/mp4">
        Your browser does not support the video tag.
    </video>
```

## To Remove or Hide a Controls

```
<video
    width="400"
    autoplay
    loop
    muted
>
    <source src="https://www.w3schools.com/html/mov_bbb.mp4"
    type="video/mp4">
</video>
```



## Anchor Tag in HTML ( `<a>` )



### Definition

The `<a>` tag is used to create **links** to another page, website, file, or section.

---

## Syntax

```
<a href="URL">Link Text</a>
```

### Basic Example

```
<a href="https://google.com">Visit Google</a>
```

Clicking this opens Google.

## Summary Table

Target	Opens In	Usage
<code>_self</code>	Same tab	Default
<code>_blank</code>	New tab	External links 
<code>_parent</code>	Parent frame	Frames
<code>_top</code>	Full window	Break iframe

## Best Practice (Security)

Always use this with `_blank` :

```
<a href="https://google.com" target="_blank" rel="noopener noreferrer">  
  Open Securely</a>
```

This prevents security issues.

---

# Bold / Italic / Underline / Strikethrough Tags

## Definition

These tags are used to **style text** in HTML.

---

## Important Tags

### 1. Bold Text → `<b>`

```
<b>This is bold text</b>
```

**Output:** This text appears bold

---

### 2. Italic Text → `<i>`

```
<i>This is italic text</i>
```

**Output:** Text appears italic

---

### 3. Underline Text → `<u>`

```
<u>This is underlined text</u>
```

**Output:** Text appears underlined

---

### 4. Strikethrough Text → `<s>` or `<strike>`

```
<s>This is strikethrough text</s>
```

**Output:** Text appears with a line through it

---



## 2.8 Pre Tag in HTML

### Definition

The `<pre>` tag is used to display **preformatted text** exactly as written.

### Important Points

- Preserves **spaces**
- Preserves **line breaks**
- Useful for **displaying code**
- Shows text in **same format as written**

### Syntax

```
<pre>  
Your text here</pre>
```

### Example

```
<pre>  
This is line 1
```

```
This is line 2 with spaces  
This is line 3</pre>
```

Output shows same spacing and line breaks

---

## Summary

Tag	Purpose
<pre>	Preserves text formatting

## Common Use Case

```
<pre>  
function hello() {  
    console.log("Hello World");  
}</pre>
```

Used for displaying code.

---



## 2.10 Superscript and Subscript Tags



### Definition

These tags are used to display text **above or below normal text line**.

---

### Important Tags

#### 1. Superscript → `<sup>`

Displays text slightly **above** the normal line.

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

**Output:**  $x^2$

---

## 2. Subscript → `<sub>`

Displays text slightly **below** the normal line.

```
H<sub>2</sub>O
```

**Output:** H<sub>2</sub>O

### Example

```
<p>(a + b)<sup>2</sup> = a<sup>2</sup> + b<sup>2</sup> + 2ab
</p><p>CH<sub>4</sub> + O<sub>2</sub> → H<sub>2</sub>O + CO<s
ub>2</sub></p>
```

### Summary Table

Tag	Purpose	Example
<code>&lt;sup&gt;</code>	Superscript	x <sup>2</sup>
<code>&lt;sub&gt;</code>	Subscript	H <sub>2</sub> O



## 3.1 Character Entity Reference in HTML



### Definition

Character entities are used to display **special or reserved characters** in HTML.

### Syntax

```
&entityName;
```

Starts with `&` and ends with `;`

## ★ Common Examples

Entity	Output	Meaning
&lt;	<	Less than
&gt;	>	Greater than
&	&	Ampersand
&nbsp;	(space)	Non-breaking space
&copy;	©	Copyright
&reg;	®	Registered
&euro;	€	Euro

## 🧠 Example

```
<p>5&lt; 10</p><p>Tom&amp; Jerry</p><p>&copy; 2026 My Website</p>
```

### Output:

5 < 10

Tom & Jerry

© 2026 My Website

## 🧭 Summary

Symbol	Entity
<	&lt;
>	&gt;
&	&amp;
Space	&nbsp;



## Level 3: Browser Tools

Browser tools help developers **inspect, debug, and edit web pages**.

---

## ★ 1. View Page Source

- Shows original HTML code
  - Right click → **View Page Source**
- 

## ★ 2. Inspect Element

- Inspect and edit HTML, CSS, JS
  - Right click → **Inspect**
  - Used for debugging
- 

## ★ 3. Responsive Design

- Test website on different screen sizes
  - Mobile, tablet, desktop view
  - Shortcut: `Ctrl + Shift + M`
- 

## ★ 4. Live Edit Code

- Edit HTML, CSS, JS in browser
  - Changes are temporary (not saved)
  - Useful for testing quickly
- 

## ★ 5. Validate Web Pages

- Check HTML errors using:  
<https://validator.w3.org/>
- 

## 🧠 Summary

Tool	Purpose
View Source	See HTML code
Inspect	Debug and edit
Responsive Mode	Test screen sizes
Live Edit	Temporary editing
Validator	Check HTML errors

## 🎯 Most Important Shortcut

- **Inspect Element:** `Right Click → Inspect`
- **Open DevTools:** `F12`

Note —

That Chrome extension is **Web Developer (by Chris Pederick)** 📱

It adds a toolbar with options like:

- Disable All Styles
- Disable Inline Styles
- Disable Browser Default Styles
- View CSS
- Resize Browser
- View Forms, Images, Cookies, etc.

These exact menu items are signature features of this extension.

## 🔗 Chrome Extension Name

**Web Developer**

## 📥 Chrome Web Store Link

<https://chrome.google.com/webstore/detail/web-developer/>

## **Different Screen Sizes (Responsive Design)**



### **Definition**

Responsive design makes a website **adapt to different screen sizes** like mobile, tablet, and desktop.

---

### **Important Points**

- Adjusts layout for **mobile, tablet, desktop**
  - Uses **flexible layouts**
  - Optimizes **images and content**
  - Improves **user experience**
- 

### **Common Screen Types**

Device	Width Example
Mobile	320px – 480px
Tablet	768px – 1024px
Laptop	1024px – 1440px
Desktop	1440px+

---

### **Important Meta Tag (Required)**

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

This makes website responsive.

---

## Example

```
<!DOCTYPE html><html><head><meta name="viewport" content="width=device-width, initial-scale=1.0"></head><body><h1>Responsive Website</h1><p>This works on mobile and desktop</p></body></html>
```

## How to Test in Browser

Open DevTools → Press:

Ctrl + Shift + M

Select devices like:

- iPhone
- Pixel
- iPad

Chrome Extension for Mobile Simulator

<https://chromewebstore.google.com/detail/mobile-simulator-responsi/ckejmhbmlajgoklhgbapkiccekfoccmk>



## Using validator.w3.org (HTML Validator)



### Definition

validator.w3.org is used to **check HTML code for errors and correctness.**

Website:

<https://validator.w3.org/>

## Important Benefits

1. Ensures **correct HTML syntax**
  2. Prevents **browser compatibility issues**
  3. Improves **SEO**
  4. Helps in **debugging errors**
  5. Improves **website performance**
- 



## Semantic vs Non-Semantic Tags



### Semantic Tags

Semantic tags have **meaning** and describe the content.

#### ★ Benefits

- Improves **SEO**
- Better **accessibility**
- Easier to understand structure



### Examples

```
<header></header><nav></nav><section></section><article></article><footer></footer>
```



### Non-Semantic Tags

Non-semantic tags have **no specific meaning**. Used mainly for styling.

#### ★ Purpose

- Used for layout and design
- No SEO benefit

## Examples

```
<div></div><span></span><b></b><i></i>
```

## Difference Table

Semantic	Non-Semantic
Has meaning	No meaning
SEO friendly	Not SEO friendly
Better structure	Used for styling
Example: <header>	Example: <div>

## Semantic Tags (Meaningful Tags)

These tags describe the **purpose of the content**.

### Layout Semantic Tags

```
<header><nav><main><section><article><aside><footer>
```

### Text Semantic Tags

```
<h1> to<h6><p><strong><em><mark><blockquote><cite><time><abbr>
```

### Media Semantic Tags

```
<figure><figcaption>
```

### Form Semantic Tags

```
<form><label><fieldset><legend>
```



## Non-Semantic Tags (Generic Tags)

These tags **do not** describe content meaning.

### ★ Layout Non-Semantic Tags

```
<div><span>
```

### ★ Text Non-Semantic Tags

```
<b><i><u><s><font><!-- deprecated --><big><!-- deprecated -->
<small>
```

### ★ Generic Container Tags

```
<div><span>
```



## Important Body Semantic Tags

These tags are used inside the `<body>` to structure a webpage.

### 1. `<header>`



#### Purpose

Top section of website (logo, title, menu)

```
<header><h1>My Website</h1></header>
```

## 2. <nav>

### Purpose

Navigation menu (links)

```
<nav><a href="#">Home</a><a href="#">About</a></nav>
```

## 3. <main>

### Purpose

Main content area

```
<main><p>Main content here</p></main>
```

## 4. <section>

### Purpose

Group related content

```
<section><h2>News Section</h2></section>
```

## 5. <article>

### Purpose

Independent content (blog, post)

```
<article><h3>Blog Post</h3></article>
```

## 6. <aside>

## Purpose

Sidebar content (ads, links)

```
<aside><p>Sidebar content</p></aside>
```

## 7. `<footer>`

## Purpose

Bottom section (copyright, contact)

```
<footer><p>© 2026 My Website</p></footer>
```

## Full Example Layout

```
<body><header><h1>Website Header</h1></header><nav><a href="#">Home</a><a href="#">About</a></nav><main><section><article>Article 1</article><article>Article 2</article></section><aside>Sidebar Content</aside></main><footer>Copyright 2026</footer></body>
```

## Most Important Semantic Tags (Must Know)

```
headernavmainsectionarticleasidefooter
```

These are used in **every professional website.**

# Recommended Folder Structure (Website)

## Purpose

Organize website files properly for easy management and scalability.

## Standard Folder Structure

```
project/
|
├── index.html          ← MainHTML file
├── pages/              ← OtherHTML pages
│   ├── about.html
│   └── contact.html
|
├── css/                ← CSS files
│   └── style.css
|
├── scripts/             ← JavaScript files
│   └── script.js
|
├── images/              ← Images
│   └── logo.png
|
└── assets/              ← Fonts, videos, icons
    └── font.ttf
```

## Important Points

- `index.html` → Main entry file
- `css/` → All CSS files
- `scripts/` → All JavaScript files

- `images/` → All images
  - `assets/` → Fonts, videos, icons
  - `pages/` → Extra HTML pages
- 

## Example Linking Files

### Link CSS

```
<link rel="stylesheet" href="css/style.css">
```

### Link JS

```
<script src="scripts/script.js"></script>
```

### Add Image

```

```

## Best Practice

Always keep files organized like this for real projects.

---



## Navigation Tag in HTML (`<nav>`)



### Definition

The `<nav>` tag is used to define **navigation links (menu)**.

---

## Example

```
<nav><a href="#">Home</a><a href="#">About</a><a href="#">Contact</a></nav>
```

## ★ Important Points

- Used for **menu / navigation**
- Contains **links** (`<a>`)
- Semantic tag (SEO friendly)
- Usually placed inside `<header>`

## ⌚ Common Use

```
<header><nav><a href="#">Home</a><a href="#">Services</a><a href="#">Contact</a></nav></header>
```

## 🧭 Summary

Tag	Purpose
<code>&lt;nav&gt;</code>	Navigation menu

# 📘 Block vs Inline Elements

## ◆ Block Elements

### 💡 Definition

Start on a **new line** and take **full width**.

### ★ Features

- Start on new line

- Take full width
- Can set width & height
- Can use margin & padding

## Examples

```
<div></div><p></p><h1></h1><section></section><ul></ul>
```

## Inline Elements

### Definition

Stay in the **same line** and take only **required width**.

### Features

- No new line
- Take only needed width
- Used inside text

## Examples

```
<span></span><a></a><img><strong></strong><em></em>
```

## Difference Table

Block	Inline
New line	Same line
Full width	Content width
Example: <code>&lt;div&gt;</code>	Example: <code>&lt;span&gt;</code>

### Example

```
<div>This is block</div><div>This is block</div><span>This is  
inline</span><span>This is inline</span>
```



## Ordered List in HTML ( `<ol>` )



### Definition

Used to create a **numbered list**.



### Syntax

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



### Example

```
<ol><li>Wake up</li><li>Brush teeth</li><li>Take a shower</li>  
><li>Have breakfast</li></ol>
```

**Output:**

1. Wake up
2. Brush teeth
3. Take a shower
4. Have breakfast



### Important Points

- `<ol>` → Ordered list
- `<li>` → List item
- Automatically shows **numbers**

- Used when **order matters**
- 



## Types of Ordered Lists ( `<ol>` )

Use the `type` attribute to change numbering style.

---

### ★ 1. Numeric (Default)

```
<ol type="1"><li>Apple</li><li>Banana</li></ol>
```

Output:

1. Apple
  2. Banana
- 

### ★ 2. Uppercase Letters

```
<ol type="A"><li>Apple</li><li>Banana</li></ol>
```

Output:

- A. Apple
  - B. Banana
- 

### ★ 3. Lowercase Letters

```
<ol type="a"><li>Apple</li><li>Banana</li></ol>
```

Output:

- a. Apple
  - b. Banana
-

## ★ 4. Uppercase Roman Numbers

```
<ol type="I"><li>Apple</li><li>Banana</li></ol>
```

Output:

- I. Apple
  - II. Banana
- 

## ★ 5. Lowercase Roman Numbers

```
<ol type="i"><li>Apple</li><li>Banana</li></ol>
```

Output:

- i. Apple
  - ii. Banana
- 

## ⌚ Summary Table

Type	Result
type="1"	1, 2, 3
type="A"	A, B, C
type="a"	a, b, c
type="I"	I, II, III
type="i"	i, ii, iii

---



## Table Tags in HTML

1. **<tr>** → Table Row



Purpose

Defines a **row** in a table.

```
<tr></tr>
```

## 2. **<th>** → Table Header



### Purpose

Defines a **header cell** (bold by default).

```
<th>Name</th><th>Age</th>
```

Output: **Name, Age**

## 3. **<td>** → Table Data



Defines **data cell** in table.

```
<td>John</td><td>30</td>
```

## 🧠 Complete Example

```
<table border="1"><tr><th>Name</th><th>Age</th></tr><tr><td>John</td><td>30</td></tr><tr><td>Jane</td><td>25</td></tr></table>
```

## 🧭 Summary Table

Tag	Purpose
<tr>	Table row
<th>	Table header

Tag	Purpose
<td>	Table data



## Complete Example

```
<table border="1">

<thead>
<tr>
    <th>Name</th>
    <th>Age</th>
    <th>Email</th>
</tr>
</thead>

<tbody>
<tr>
    <td>John</td>
    <td>30</td>
    <td>john@email.com</td>
</tr>

<tr>
    <td>Jane</td>
    <td>25</td>
    <td>jane@email.com</td>
</tr>
</tbody>

<tfoot>
<tr>
    <td colspan="3">Total Users: 2</td>
</tr>
</tfoot>
```

```
</table>
```

## Structure Diagram

```
table
└─thead   ← headings
└─tbody   ← main data
└─tfoot   ← summary/footer
```



### `<caption>` Definition

The `<caption>` tag is used to give a **title to the table**.

It appears at the **top of the table**.

```
<table border="1">

<caption>User Details Table</caption>

<thead>
<tr>
    <th>Name</th>
    <th>Age</th>
    <th>Email</th>
</tr>
</thead>

<tbody>
<tr>
    <td>John</td>
    <td>30</td>
    <td>john@email.com</td>
</tr>
</tbody>
```

```

</tr>

<tr>
    <td>Jane</td>
    <td>25</td>
    <td>jane@email.com</td>
</tr>
</tbody>

<tfoot>
<tr>
    <td colspan="3">Total Users: 2</td>
</tr>
</tfoot>

</table>

```



## colspan in HTML Table



### Definition

`colspan` is used to make a cell **span across multiple columns**.

It merges columns horizontally.

---



### Syntax

```
<td colspan="number">Content</td>
```



## rowspan in HTML Table (Vertical Merge)



## Definition

`rowspan` is used to make a cell **span across multiple rows**.

It merges cells **vertically**.

---



## Syntax

```
<td rowspan="number">Content</td>
```



# Forms in HTML



## Definition

HTML forms are used to **collect user input** like name, email, password, etc.

---



## Basic Syntax

```
<form>
    input elements</form>
```



## Simple Example

```
<form>

    <label>Name:</label>
    <input type="text">

    <br><br>

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

```
<br><br>

<input type="submit">

</form>
```

## ★ Important Form Elements

Tag	Purpose
<form>	Form container
<input>	Input field
<label>	Label for input
<textarea>	Multi-line text
<select>	Dropdown
<button>	Button

## █ Common Input Types

```
<input type="text">      <!-- Text -->
<input type="email">     <!-- Email -->
<input type="password">   <!-- Password -->
<input type="number">    <!-- Number -->
<input type="radio">     <!-- Radio button -->
<input type="checkbox">   <!-- Checkbox -->
<input type="submit">    <!-- Submit button -->
```

## 🧠 Complete Example

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

```

<body>

<h2>Registration Form</h2>

<form>

<label>Name:</label>
<input type="text"><br><br>

<label>Email:</label>
<input type="email"><br><br>

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

<label>Gender:</label>
<input type="radio" name="gender"> Male
<input type="radio" name="gender"> Female

<br><br>

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

</form>

</body>
</html>

```

## Important Attributes

Attribute	Purpose
action	Where to send form data
method	GET or POST

Attribute	Purpose
name	Input name
placeholder	Hint text
required	Mandatory field

## Example with attributes

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

<input type="text" name="username" placeholder="Enter name" required>

<input type="submit">

</form>
```

## 🎯 Most Important Tags

form  
input  
label  
textarea  
select  
button



## name and value Property in HTML

Used inside form inputs to identify and send data to the server.

### 1. name Property

## Purpose

Identifies the input field when form is submitted.

Server receives data using this name.

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

If user enters: **John**

Server receives:

```
username = John
```

## 2. **value** Property

### Purpose

Sets the **default value** of input.

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

Input box will already contain:

```
John
```

## Complete Example

```
<form action="/submit" method="POST"><label>Name:</label><input type="text" name="username" value="John"><br><br><input type="submit" type="submit"></form>
```

## What gets sent to server

If user submits form:

```
username = John
```

## Real Example (Multiple Inputs)

```
<form><input type="text" name="name" value="John"><input type="email" name="email" value="john@email.com"><input type="submit"></form>
```

Data sent:

```
name = Johnemail = john@email.com
```

## Summary Table

Property	Purpose
name	Identifies input
value	Default or submitted value



## <iframe> Tag in HTML



### Definition

`<iframe>` is used to **embed another webpage inside your webpage.**

It creates a small window showing another page.

## Basic Syntax

```
<iframe src="URL"></iframe>
```

## Example 1: Embed a Website

```
<iframe src="https://example.com"></iframe>
```

## Example 2: Set Width and Height

```
<iframe src="https://example.com" width="600" height="400"></iframe>
```

## Example 3: Embed YouTube Video

```
<iframe width="560" height="315" src="https://www.youtube.com/embed/dQw4w9WgXcQ" allowfullscreen></iframe>
```

## Important Attributes

Attribute	Purpose
src	URL to embed
width	Width of iframe
height	Height of iframe
title	Accessibility title
allowfullscreen	Allow fullscreen

## Real Example

```
<!DOCTYPE html><html><body><h2>Google Website</h2><iframe src="https://www.wikipedia.org" width="800" height="400"></iframe></body></html>
```

## Uses in Real Projects

- Embed YouTube videos
  - Embed Google Maps
  - Embed other webpages
  - Embed dashboards
- 

## Summary

```
iframe = webpage inside webpage
```

## To Open Inside a Popup

```
<!DOCTYPE html>
<html>
<head>
<title>Popup Example</title>

<script>
function openPopup() {
    window.open(
        "https://www.wikipedia.org",
        "popupWindow",
        "width=600,height=400"
    );
}
</script>

</head>
<body>

<button onclick="openPopup()">
    Open Popup

```

```
</button>

</body>
</html>
```

## To Open in a Same Page

---

```
<h2>Wikipedia inside iframe</h2>

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

<h3>Click link to open inside iframe:</h3>

<a href="https://www.wikipedia.org" target="myframe">
    Open Wikipedia
</a>

<br><br>

<iframe
    name="myframe"
    width="600"
    height="400">
</iframe>

</body>
</html>
```

## Optional Notes

---

### What is Version Control?

#### Definition

Version Control is a system used to **track changes in files over time**.

---

#### Why it is used

- Track file history
  - Work in teams
  - Undo mistakes
  - Manage different versions
- 

#### Example

```
Version1 → index.html (basic page) Version2 → added CSS  
→ added JavaScript
```

You can go back to any version anytime.

---

#### Types

##### 1. Centralized Version Control

- Example: SVN
- One central server

##### 2. Distributed Version Control

- Example: Git 
  - Each user has full copy
-

## ★ Important Concepts

Term	Meaning
Commit	Save version
Branch	Separate work
Merge	Combine work
Undo	Go back to old version

## ★ Real Tool Used

Git (Most popular)  
GitHub (Cloud storage for Git)

## 🧠 Real-life Example

Like **Google Docs** version history

You can see old versions and restore them.

## 🧭 Summary (1 line)

**Version Control = Track + Manage + Restore file versions**

## How to Create a CodeSpace in Github

