Installing Visual Studio Code (VS Code)

(By FatimaTabassum)

Why Use VS Code : VS Code is a free, lightweight, and powerful code editor developed by Microsoft. It's widely used for web development and vast extension support.

Installation Steps:

1. **Download VS Code:**
    a. Visit the official VS Code website.
    b. Click on the **"Download for Windows"** button.
2. **Run the Installer:**
    a. Locate the downloaded file (usually in your **Downloads** folder) named something like `VSCodeUserSetup-x64-1.62.3.exe`.
    b. Double-click the file to start the installation.
3. **Setup Wizard:**
    a. In the setup window, select **"I accept the agreement"** and click **Next**.
    b. Choose the installation location or leave it as default, then click **Next**.
    c. Select additional tasks:
        i. **Add "Open with Code" action to Windows Explorer file context menu**
        ii. **Add to PATH (recommended)**
    d. Click **Next**, then **Install**.
4. **Finish Installation:**
    a. Once the installation is complete, click **Finish** to launch VS Code.

Recommended Extensions for Web Development:

- **Live Server:** Launch a local development server with live reload feature
- **Prettier - Code formatter:** Automatically formats your code for better readability.
- **ESLint:** Identifies and reports on patterns found in ECMAScript/JavaScript code.
- **HTML Snippets:** Provides a set of useful code snippets for HTML.
- **JavaScript (ES6) code snippets:** Provides JavaScript code snippets.

Creating HTML, CSS, and JavaScript Files

File Extensions:

- **HTML:** `.html` (e.g., `index.html`)
- **CSS:** `.css` (e.g., `styles.css`)
- **JavaScript:** `.js` (e.g., `script.js`)

Enable Auto Save: Auto Save helps you **automatically save** your code whenever you stop typing, click outside the editor, or switch files. This is useful so you don't forget to press (`Ctrl + S`) every time.

What is HTML?

- **HTML (HyperText Markup Language)** is the standard language used to create and structure content on the web. It allows you to define elements like headings, paragraphs, links, images,

and more, enabling browsers to render web pages appropriately.Originally introduced in the **1990s**.

- Basic structure for web content: headings, paragraphs, links, images, etc.
- Versions evolved: HTML → HTML 2.0 → HTML 3.2 → HTML 4.01 → **HTML5**

Basic Structure of an HTML Document

```
<!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>

</body>
</html>
```

Every HTML document follows a standard structure:

- `<!DOCTYPE html>`: Declares the document type and version of HTML.
- `<html>`: Root element of the HTML document.
- `<head>`: Contains metadata, links to stylesheets, and scripts.
- `<title>`: Sets the title of the document (shown in browser tab).
- `<body>`: Contains the visible content of the document.

HTML Key Concepts Tag vs Element vs Attribute

Tag: A **tag** is what you write in the code to define elements.

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

- `<p>` is an **opening tag**
- `</p>` is a **closing tag**

Element : An **element** is the whole thing: opening tag + content + closing tag.

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

This entire line is a **paragraph element**.

Attribute: An **attribute** adds extra information to a tag.

```
<img src="dog.jpg" alt="A dog" width="200">
```

Attributes:

- `src` tells the image source

- `alt` is alternative text
- `width` sets the width

📌 Attributes are always in the **opening tag** and follow `name="value"` format.

Block vs Inline vs Inline-Block

Block-Level Elements

- Take up the **full width** of the container
- Start on a **new line**
- Examples: `<div>`, `<p>`, `<h1>` to `<h6>`, `<section>`, `<article>`

```
<p>This is block 1</p>
<p>This is block 2</p>
```

Each paragraph appears on a new line.

Inline Elements

- Take up **only as much space** as needed
- Do **not** start on a new line
- Examples: `<span>`, `<a>`, `<strong>`, `<em>`, `<img>`

```
<span>This</span> <span>is inline</span>
```
They appear **side-by-side**.

Inline-Block Elements

- Behave **like inline elements** (side-by-side)
- But you can set **width and height** (like block elements)

```
<span style="display: inline-block; width: 100px;">Box 1</span>
<span style="display: inline-block; width: 100px;">Box 2</span>
```

Comments in HTML: Use comments to explain your code. They don't appear on the webpage.

```
<!-- This is a comment --> (ctrl+/)
<p>This is visible.</p>
```

Self-Closing Tags: Some tags don’t need a closing tag.

```
<br />   <!-- Line break -->
<img src="dog.jpg" alt="Dog" />
<hr />   <!-- Horizontal line -->
<input type="text" />
```

🧠 What is a Favicon?

A favicon (short for **“favorite icon”**) is a small icon that appears in:

- Browser tab next to the page title
- Bookmarks/favorites list
- Browser history

Favicons help users **identify your website quickly**.

✅ How to Add a Favicon

Use the `<link>` tag inside the `<head>` section of your HTML:

```
<link rel="icon" href="favicon.ico" type="image/x-icon">
```

Formatting Tags: HTML formatting tags are used to **style text** — making it bold, italic, underlined, etc.

Tag	Meaning / Use	Typical Render	Example	Notes & A11y tips
`<b>`	Stylistic bold (no extra meaning)	Bold	`<b>Bold look</b>`	Prefer `<strong>` for importance.
`<strong>`	Strong importance	Bold	`<strong>Warning</strong>`	Conveys emphasis to screen readers.
`<i>`	Stylistic italics	Italic	`<i>Latin term</i>`	Prefer `<em>` for emphasis/meaning.
`<em>`	Emphasis	Italic	`<em>really</em>`	Nested `<em>` increases emphasis.
`<u>`	Unarticulated annotation	Underline	`<u>misspelling</u>`	Don’t fake links with it.
`<del>`	Deletion	Strike	`<del>Old text</del>`	Use with `<ins>` for diffs.
`<ins>`	Insertion	Underline	`<ins>New text</ins>`	Often paired with `<del>`.
`<mark>`	Highlighted reference	Highlight	`<mark>Match</mark>`	Good for search results.
`<small>`	Side comments / fine print	Smaller	`<small>T&Cs apply</small>`	Semantically “side note”.
`<sup>`	Superscript	Raised	`x<sup>2</sup>`	Use for exponents, footnotes.
`<sub>`	Subscript	Lowered	`H<sub>2</sub>O`	Chemical formulas, etc.
` `	Line break	New line	`Line 1 Line 2`	Don’t use for spacing.

<pre>	Preformatted text (preserves spaces and line breaks)	Keeps spacing & line breaks	<pre> Code Block</pre>	--

Examples :

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>Inline Text Tags Demo (No CSS)</title>
</head>
<body>
  <h1>Inline Text Tags Demo (No CSS)</h1>

  <p><b>&lt;b&gt;</b>: <b>This text is bold (presentational)</b></p>
  <p><strong>&lt;strong&gt;</strong>: <strong>This text is strong (semantic
emphasis)</strong></p>

  <p><i>&lt;i&gt;</i>: <i>This text is italic (presentational)</i></p>
  <p><em>&lt;em&gt;</em>: <em>This text is emphasized (semantic)</em></p>

  <p><u>&lt;u&gt;</u>: <u>This text is underlined</u></p>

  <p><mark>&lt;mark&gt;</mark>: Here is <mark>highlighted text</mark> inside a sentence.</p>

  <p><small>&lt;small&gt;</small>: <small>This is small print (e.g., fine print)</small></p>

  <p><del>&lt;del&gt;</del>: This item is <del>no longer available</del>.</p>

  <p><sub>&lt;sub&gt;</sub> and <sup>&lt;sup&gt;</sup>:
    H<sub>2</sub>O,  E = mc<sup>2</sup>
  </p>

  <h2><code>&lt;pre&gt;</code> (Preformatted)</h2>
  <pre>
Line 1 stays as-is.
    Line 2 keeps its spaces.
        Line 3 is indented.
Tabs and
line breaks are preserved.
  </pre>

  <hr />
  <p>All of the above uses only HTML—no CSS.</p>
</body>
</html>
```

What Is the <a> (Anchor) Tag?

The <a> tag stands for **“anchor”**. It is used to create **links** in an HTML page.

You can use it to:

- Go to another website 🌐
- Open a file 📄
- Jump to another section on the same page 📌
- Download files 📥
- Email someone 📧

✅ Basic Syntax:

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

This will create a clickable link that takes the user to Google.

Important Attributes of <a> Tag

Attribute	Purpose	Example
`href`	URL or location to link to	`href="https://example.com"`
`target`	Where to open the link (`_self`, `_blank`, tc.)	`target="_blank"` (opens in new tab)
`title`	Tooltip text shown on hover	`title="Go to Google"`
download	Prompts download instead of opening the e	`download="filename.pdf"`
`type`	Specifies type of the linked document	`type="application/pdf"`
`id`	Used with `href="#id"` to jump to a section	`id="about"` and `<a href=”#about”>`

Examples of Common Use Cases

🔗 1. Link to another website:

```
<a href="https://www.wikipedia.org">Visit Wikipedia</a>
```

🧭 2. Open link in a new tab:

```
<a href="https://youtube.com" target="_blank">YouTube</a>
```

🖱️ 3. Add tooltip on hover:

```
<a href="https://apple.com" title="Go to Apple">Apple</a>
```

4. Download a file:

```
<a href="files/resume.pdf" download>Download Resume</a>
```

🔁 5. Link to a section on the same page:

```
<!-- Somewhere on the page -->
<a href="#contact">Go to Contact</a>
```

6.we can navigate from one html page to another using anchor tag

Eg: Go To Contact Page

7. Send an email

```
<ahref="mailto:example@example.com">Email Someone</a>
```

Lists In HTML: There are **2 types of lists** in HTML:

Type	Tag Used	Purpose
Ordered list		For numbered lists (1, 2, 3…)
Unordered list		For bullet point lists (•, ◦, etc.)

Unordered List (<ul>): Displays a list with **bullets** (•)

```
<ul>
  <li>HTML</li>
  <li>CSS</li>
  <li>JavaScript</li>
</ul>
```

This shows Like this:

- HTML
- CSS
- JavaScript

Important Attributes:

<ul type=""> – (not recommended in HTML5, but still works in some browsers)

Type	Value	Example	Code Example
Bullet Style	Disc(by default)	•	<ul type="disc"> Item
Bullet Style	circle	◦	<ul type="circle"> Item
Bullet Style	square	■	<ul type="square"> Item

Ordered List (<ol>): Displays a list with **numbers** or **letters**(1, 2, 3 / A, B, C)

```
<ol>
  <li>Step One</li>
  <li>Step Two</li>
  <li>Step Three</li>
</ol>
```

1. Step One
2. Step Two

3. Step Three

Important Attributes:

Attribute	Value / Description	Example	Code Example
type	Defines the numbering style	1, 2, 3	<ol type="1"> Item
	A, B, C	A, B, C	<ol type="A"> Item
	a, b, c	a, b, c	<ol type="a"> Item
	I, II, III	I,II, III	<ol type="I"> Item
	i, ii, iii	i, ii, iii	<ol type="i"> Item
start	Starting number of list	5, 6, 7	<ol start="5"> Item
reversed	Reverse numbering	3, 2, 1	<ol start="3" reversed> Item

HTML Table: An HTML table is a way to **display data in rows and columns** – like a spreadsheet.

Table Structure

```
<table border="2px">
<caption>Monthly Sales Report</caption>
  <thead>
    <tr><th>Product</th><th>Price</th></tr>
  </thead>
  <tbody>
    <tr><td>Apples</td><td>$1</td></tr>
    <tr><td>Oranges</td><td>$2</td></tr>
  </tbody>
  <!-- optional -->
  <tfoot>
    <tr><td>Total</td><td>$3</td></tr>
  </tfoot>
</table>
```

Attributes:

Attribute	Description	Example Code
border	Specifies border width of the table	<table border="1">...</table>
cellpadding	Space between cell content and cell border	<table cellpadding="10">...</table>

cellspacing	Space between table cells	<table cellspacing="5">...</table>
width	Width of the table	<table width="500">...</table>
height	Height of the table	<table height="200">...</table>
align	Horizontal alignment of table (left, center, right)	<table align="center">...</table>
bgcolor	Background color of the table	<table bgcolor="lightblue">...</table>
summary	Summary of the table content (for accessibility)	<table summary="Sales report">...</table>
frame	Specifies which sides of the table border to show (void, above, below, hsides, vsides)	<table frame="box">...</table>
rules	Specifies internal borders (none, groups, rows, cols, all)	<table rules="all">...</table>
colspan	Number of columns a cell should span	<td colspan="2">Merged Cell</td>
rowspan	Number of rows a cell should span	<td rowspan="2">Merged Cell</td>

Example:

```
<table
  border="2"
  cellpadding="10"
  width="600"
  bgcolor="#f0f8ff"
  align="center"
  summary="Monthly Sales Report"
  frame="box"
  rules="all"
>
  <caption>
    <strong>Monthly Sales Report</strong>
  </caption>
  <thead>
    <tr>
      <th rowspan="2">Product</th>
      <th colspan="2">January</th>
    </tr>
    <tr>
      <th>Units</th>
      <th>Revenue</th>
    </tr>
  </thead>
  <tbody>
    <tr>
```

```
            <td>Apples</td>
            <td>100</td>
            <td>$200</td>
          </tr>
          <tr>
            <td>Oranges</td>
            <td>80</td>
            <td>$160</td>
          </tr>
          <tr>
            <td colspan="1">Total</td>
            <td>180</td>
            <td>$360</td>
          </tr>
        </tbody>
      </table>
```

Forms: An HTML form is used to **collect user input**. Common forms include:

- Login forms
- Contact forms
- Sign-up forms
- Surveys
- File uploads

Basic Form Tag:

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

- **`action`** → URL to send data
- **`method`** → `get` (URL) or `post` (request body)

🔹 **Label (`<label>`)**

Binds text to an input field (helps accessibility).

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

- **`for`** in label → matches input **`id`**
- Clicking text focuses input

Inputs:

1️⃣ **Text Input**

```
<input type="text" id="name" name="username" placeholder="Enter your name">
```

- `placeholder` → hint text
- `required` → mandatory
- `name` → key for JavaScript/server
- `id` → used with label

2️⃣ Password Input

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

- Hides input characters

3️⃣ Email Input

```
<input type="email" name="useremail" placeholder="Enter your email" required>
```

- Validates proper email format

4️⃣ Number Input

```
<input type="number" name="age" min="1" max="100">
```

- Only allows numeric input
- `min` and `max` define range

5️⃣ Date Picker

```
<input type="date" name="dob">
```

- Lets user select date via calendar

6️⃣ Radio Buttons

```
<p>Gender:</p>
<input type="radio" id="male" name="gender" value="male">
<label for="male">Male</label>
<input type="radio" id="female" name="gender" value="female">
<label for="female">Female</label>
```

- Only **one** option per `name`
- `value` → data sent when selected

7️⃣ Checkbox

```
<p>Hobbies:</p>
<input type="checkbox" id="reading" name="hobbies" value="Reading">
<label for="reading">Reading</label>
<input type="checkbox" id="sports" name="hobbies" value="Sports">
<label for="sports">Sports</label>
```

- Multiple selections allowed

Buttons & Textarea

1️⃣ Submit Button

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

- Sends form data to `action` URL

2️⃣ Reset Button

```
<button type="reset">Clear</button>
```

- Clears form fields

3️⃣ Normal Button

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

- Can be used with JS

4️⃣ Textarea

```
<label for="message">Message:</label>
<textarea id="message" name="message" rows="4" cols="30" placeholder="Type your message"></textarea>
```

- Multi-line text input

✅ Dropdown (`<select>`)

```
<label for="country">Country:</label>
<select id="country" name="country">
  <option value="india">India</option>
  <option value="usa">USA</option>
  <option value="uk">UK</option>
</select>
```

- User picks **one** item
- `value` → data sent

✅ File Upload

```
<form action="/upload" method="post" enctype="multipart/form-data">
  <label for="file">Upload File:</label>
  <input type="file" id="file" name="resume">
  <input type="submit" value="Upload">
</form>
```

- `type="file"` → select file

- `enctype="multipart/form-data"` → required for file uploads

✅ Grouping Fields

Use `<fieldset>` and `<legend>` for better structure:

```
<fieldset>
  <legend>Personal Info</legend>
  Name: <input type="text" name="name">
  Email: <input type="email" name="email">
</fieldset>
```

Attributes:

Attribute	Description	Example Code
`type`	Specifies input type (`text`, `password`, `email`, etc.)	`<input type="text" name="username">`
`name`	Name of the input, key used in form submission	`<input type="email" name="useremail">`
`id`	Unique identifier for input, used with `<label>`	`<input type="text" id="name">`
`value`	Predefined value of input	`<input type="text" name="city" value="Delhi">`
`placeholder`	Hint text shown inside input	`<input type="text" placeholder="Enter your name">`
`required`	Makes input mandatory	`<input type="email" required>`
`readonly`	Input cannot be changed by user	`<input type="text" value="Read Only" readonly>`
`disabled`	Input is disabled	`<input type="text" disabled>`
`maxlength`	Maximum number of characters	`<input type="text" maxlength="10">`
`min`	Minimum value (number/date inputs)	`<input type="number" min="1">`
`max`	Maximum value (number/date inputs)	`<input type="number" max="100">`
`step`	Increment steps for number/date	`<input type="number" step="5">`
`size`	Width of input in characters	`<input type="text" size="20">`
`autocomplete`	Enable/disable browser autocomplete (on/off)	`<input type="text" autocomplete="off">`
`autofocus`	Input automatically focused on page load	`<input type="text" autofocus>`

`multiple`	Allows multiple files or multiple selection (for file/select)	`<input type="file" multiple>`
`accept`	Accepted file types (for file input)	`<input type="file" accept=".pdf,.docx">`

✅ Full Example Form:

```
<!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>
    <form action="/register" method="post" enctype="multipart/form-data">
      <fieldset>
        <legend>Registration Form</legend>
        <label for="username">Username:</label>
        <input type="text" id="username" name="username" required /><br /><br />
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required /><br /><br />
        <label for="age">Age:</label>
        <input
          type="number"
          id="age"
          name="age"
          min="10"
          max="99"
        /><br /><br />
        <p>Gender:</p>
        <input type="radio" id="male" name="gender" value="male" /> Male
        <input type="radio" id="female" name="gender" value="female" />
        Female<br /><br />
        <p>Hobbies:</p>
        <input type="checkbox" name="hobbies" value="reading" /> Reading
        <input type="checkbox" name="hobbies" value="sports" />
        Sports<br /><br />
        <label for="message">Message:</label><br />
        <textarea
          name="message"
          rows="4"
          cols="30"
          placeholder="Type your message"
        ></textarea
        ><br /><br />
        <label for="file">Upload Resume:</label>
        <input type="file" name="resume" /><br /><br />
```

```
        <label for="country">Country:</label>
        <select name="country">
          <option value="india">India</option>
          <option value="usa">USA</option></select
        ><br /><br />
        <button type="submit">Submit</button>
        <button type="reset">Clear</button>
      </fieldset>
    </form>
  </body>
</html>
```

🧠 **"Media" in HTML:** Media in web development refers to content that isn't plain text, such as:

- 🖼️ Images
- 🎵 Audio (music, sound effects)
- 🎥 Videos

HTML provides special tags to **embed and control media**.

`<img>` — Embedding Images

Syntax: `<img src="image.jpg" alt="Description" width="300" height="200">`

- Always use **`alt`** for images for accessibility.the alt text will be shown when image is broken

Attributes:

Attribute	Description
`src`	URL or path of the image
`alt`	Text shown if the image fails to load (also helps screen readers)
`width` & `height`	Set image size (pixels or %)
`loading`	`lazy` or `eager` (lazy loads images for performance)

🎵 **`<audio>` — Embedding Sound**

Syntax:

```
<audio controls>
  <source src="song.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
</audio>
```

Key Points:

- `controls` → adds play/pause buttons

- `<source>` → defines file and format
- `autoplay`, `loop`, `muted` can also be used

🎥 `<video>` — Embedding Videos

Syntax:

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  Your browser does not support the video tag.
</video>
```

Attributes:

Attribute	Description
`poster`	Image before video starts

`<iframe>` — Embedding External Content

Used to embed another webpage or external content inside your page.

Use Cases:

1. 🎬 YouTube Videos

```
<iframe src="https://www.youtube" width="560" height="315" controls></iframe>
```

2. 🗺️ Google Maps

```
<iframe src="https:/maps/" width="600" height="450"></iframe>
```

3. 📄 PDFs or Documents

```
<iframe src="file.pdf" width="600" height="400"></iframe>
```

4. 🌐 Another Website

```
<iframe src="https://example.com" width="800" height="600"></iframe>
```

5. 🧪 Reusable HTML Content (like calculators or forms)

```
<iframe src="calculator.html" width="300" height="400"></iframe>
```

🧱 Semantic Tags in HTML

Semantic tags describe their meaning for both **humans and machines**, improving **accessibility and SEO**.

Tag	Description
`<header>`	Top section of page (logo, nav)
`<nav>`	Navigation menu or links
`<main>`	Main content of the page
`<section>`	Thematic group of content
`<article>`	Standalone content (blog, news)
`<aside>`	Sidebar or related content
`<footer>`	Bottom section (copyright, contact)

Additional Semantic Tags:

Tag	Description
`<figure>`	Wraps media content
`<figcaption>`	Caption for `<figure>`
`<time>`	Represents a date/time
`<details>`	Collapsible content box
`<summary>`	Title for `<details>`
`<abbr>`	Abbreviation with full meaning on hover
`<address>`	Contact information
`<code>`	Displays computer code

Example With Semantic Tags:

```
<!DOCTYPE html>
<html>
<head>
  <title>Semantic HTML Example</title>
</head>
<body>
<header>
  <h1>My Blog</h1>
  <nav>
    <a href="#">Home</a> | <a href="#">About</a>
  </nav>
</header>
<main>
  <article>
    <h2>What is Semantic HTML?</h2>
    <p><mark>Semantic HTML</mark> makes your code easier to read and understand.</p>
    <figure>
      <img src="semantic.png" alt="Semantic layout" width="200">
      <figcaption>Fig: Example of semantic structure</figcaption>
    </figure>
    <p><time datetime="2025-06-03">Published: June 3, 2025</time></p>
```

```
  </article>
</main>
<aside>
  <h3>Related Topics</h3>
  <ul>
    <li>HTML Basics</li>
    <li>CSS Styling</li>
  </ul>
</aside>
<footer>
  <p>&copy; 2025 My Blog | <address>Contact: me@example.com</address></p>
</footer>
</body>
</html>
```

🧠 What is Accessibility ?

Accessibility in web development means designing websites so that **everyone can use them**, including people with:

- Visual impairments (blind, low vision)
- Hearing impairments
- Motor disabilities
- Cognitive or learning disabilities

Goal: Make web content **perceivable, operable, understandable, and robust** for all users.

🛠️ HTML Accessibility Features

1. **Alt Text for Images (`alt`)**

- Describes images for screen readers.

```
<img src="logo.png" alt="Company Logo">
```

- Decorative images: use `alt=""` to skip them in screen readers.

2. **Labels for Form Inputs (`<label>`)**

- Connect text with input fields for screen readers.

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

3. **Fieldset & Legend for Grouped Inputs**

- Helps understand **groups of related form fields**.

```
<fieldset>
  <legend>Personal Info</legend>
  <input type="text" name="name">
</fieldset>
```

4. **ARIA (Accessible Rich Internet Applications) Attributes**

- Adds extra accessibility information when HTML alone isn't enough.

ARIA Attribute	Purpose	Example
`role`	Defines element role	`<div role="button">Click</div>`
`aria-label`	Label for screen readers	`<button aria-label="Close menu">X</button>`
`aria-hidden`	Hides element from screen readers	`<div aria-hidden="true">Hidden</div>`
`aria-live`	Announces dynamic updates	`<div aria-live="polite">Message</div>`

5. **Keyboard Navigation**

- Ensure all interactive elements can be **accessed via `Tab`** key.

```
<a href="page.html">Go to Page</a>
<button>Click Me</button>
```

6. **Semantic HTML**

- Use **semantic tags** to convey meaning to screen readers and search engines.
- Examples: `<header>`, `<nav>`, `<main>`, `<section>`, `<article>`, `<aside>`, `<footer>` instead of always using `<div>` and `<span>`