**HTML NOTES**

**Introduction to HTML**

**What is HTML?**

* **HTML** stands for **HyperText Markup Language**.
* It is the **standard markup language** used to create web pages.
* HTML tells the browser **how to display content** such as text, images, videos, and links.

**Purpose of HTML**

* To **structure web content**.
* Defines elements like **headings, paragraphs, lists, links, tables**, and more.
* Acts as the **skeleton** of a webpage.

**Key Features of HTML**

* **Platform-independent** – works on all operating systems and browsers.
* Easy to learn and understand.
* Allows **embedding multimedia** (audio, video, images).
* Works together with **CSS (for styling)** and **JavaScript (for functionality)**.

**Components of HTML**

1. **Tags** – Keywords enclosed in angle brackets (e.g., <p>, <h1>)
2. **Elements** – A complete structure consisting of opening tag, content, and closing tag.
3. **Attributes** – Provide additional information about an element (e.g., href, src, alt)
4. **Document Structure** – Basic layout includes:
   * <!DOCTYPE html>: Declares HTML version
   * <html>: Root of the document
   * <head>: Metadata, title, links to CSS/JS
   * <body>: Visible page content

**🔹 Importance of HTML**

* **Essential skill** for web development.
* Forms the basis for building web applications and websites.
* Supported by all web browsers.

**Real-life Analogy**

Think of **HTML as the structure of a house**:

* Walls, doors, and windows = HTML elements (structure)
* Paint and decorations = CSS (style)
* Electric system and automation = JavaScript (functionality)

**HTML Elements & Tags**

**What is a Tag?**

* A **tag** is a keyword wrapped in **angle brackets (< >)**.
* Tags are used to **mark the start and end** of an HTML element.
* Tags are usually **paired**:
  + **Opening tag**: <p>
  + **Closing tag**: </p>

**What is an Element?**

* An **HTML element** is everything from the **start tag to the end tag**, including the content.
* **Example:**

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

Here, <p> is the opening tag, </p> is the closing tag, and the full structure is an **HTML element**.

**Types of HTML Elements**

1. **Block-level Elements**
   * Start on a **new line** and take up the full width.
   * Examples: <div>, <h1> to <h6>, <p>, <ul>, <li>, <table>
2. **Inline Elements**
   * Do **not start on a new line**, and only take up as much width as necessary.
   * Examples: <span>, <a>, <img>, <strong>, <em>
3. **Empty (Void) Elements**
   * Elements that **do not have closing tags**.
   * They do not contain content.
   * Examples: <br>, <img>, <hr>, <input>

**Common Tags & Their Purpose**

| **Tag** | **Purpose** |
| --- | --- |
| <h1> | Heading (largest) |
| <p> | Paragraph |
| <a> | Anchor (hyperlink) |
| <img> | Image |
| <br> | Line break |
| <div> | Division/block (layout) |
| <span> | Inline container |
| <ul> | Unordered list |
| <ol> | Ordered list |
| <li> | List item |

**Attributes in Tags**

* Tags can have **attributes** to provide extra information.
* Written inside the opening tag.
* Example:
  + <img src="image.jpg" alt="An image">
  + src and alt are attributes of the <img> tag.

**HTML Paragraph Tag (<p>)**

**What is the <p> Tag?**

* The <p> tag is used to define a **paragraph** in an HTML document.
* It is a **block-level** element, meaning it starts on a new line and adds vertical spacing.

**🔹 Syntax**

<p>Your paragraph text goes here.</p>

* Begins with an **opening tag** <p>.
* Ends with a **closing tag** </p>.
* The text in between is the **paragraph content**.

**Key Points**

* Browsers automatically add **space before and after** each paragraph.
* Paragraphs are used to **organize text** and improve readability.
* Multiple <p> tags can be used on a single page for different sections of text.

**HTML Heading Elements (<h1> to <h6>)**

**What are Heading Tags?**

* HTML provides **6 levels of headings** using the tags:  
  <h1>, <h2>, <h3>, <h4>, <h5>, and <h6>.
* Headings are used to **define titles and subtitles** on a webpage.

**Purpose of Headings**

* Organize content into **sections**.
* Improve **readability** and **SEO (Search Engine Optimization)**.
* Help screen readers and search engines understand page structure.

**Levels of Headings**

| **Tag** | **Purpose** | **Default Size (approx.)** |
| --- | --- | --- |
| <h1> | Main/Top-level heading | Largest |
| <h2> | Section heading | Smaller than <h1> |
| <h3> | Sub-section heading | Smaller than <h2> |
| <h4> | Minor heading within <h3> | Smaller than <h3> |
| <h5> | Smaller heading | Smaller than <h4> |
| <h6> | Smallest heading | Smallest |

**Syntax Example**

<h1>Main Title</h1>

<h2>Sub Title</h2>

<h3>Topic</h3>

<h4>Sub-topic</h4>

<h5>Note</h5>

<h6>Reference</h6>

### Best Practices

* Use **only one <h1>** per page (for main title).
* Follow a **logical hierarchy** — don’t skip heading levels.
* Avoid using headings just for **styling** (use CSS instead).
* Headings should be **descriptive and meaningful**.

**HTML Boilerplate Code**

**What is Boilerplate Code?**

* **Boilerplate** refers to a **standard starting template** used in most HTML documents.
* It provides the **basic structure** required for a valid HTML page.
* Every HTML file should start with this to ensure **browser compatibility** and proper rendering.

**Basic HTML Boilerplate Structure**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Page Title</title>

</head>

<body>

<!-- Page content goes here -->

</body>

</html>

**Explanation of Each Part**

| **Part** | **Description** |
| --- | --- |
| <!DOCTYPE html> | Declares the document type and version of HTML (HTML5 in this case). |
| <html lang="en"> | Root element of the document. lang="en" specifies the language as English. |
| <head> | Contains **meta-information** like title, character set, styles, scripts, etc. |
| <meta charset="UTF-8"> | Sets character encoding. UTF-8 supports all common characters and symbols. |
| <title> | Sets the title of the web page (appears in the browser tab). |
| <body> | Contains **visible content** shown on the browser (text, images, buttons, etc.). |

**Why Use Boilerplate Code?**

* Ensures the **page is rendered correctly** by browsers.
* Provides a **consistent and organized** structure.
* Makes it easier to **add CSS, JavaScript, and SEO elements** later.
* Essential for building professional and standard web pages.

## Lists in HTML

### What are Lists?

* Lists are used to **group related items** in an organized way.
* HTML provides three main types of lists:
  1. **Ordered List** (<ol>)
  2. **Unordered List** (<ul>)
  3. **Description List** (<dl>)

### 1.Ordered List (<ol>)

* Displays list items in a **numbered** format.
* Each item is placed inside a <li> (list item) tag.

#### Use Case:

* Steps in a process
* Rankings
* Instructions

#### Example:

<ol>

<li>Wake up</li>

<li>Brush teeth</li>

<li>Eat breakfast</li>

</ol>

### 2.Unordered List (<ul>)

* Displays list items with **bullets** (●) instead of numbers.
* Each item is also written inside a <li> tag.

#### Use Case:

* Features of a product
* Items in a bag
* Grocery list

#### Example:

<ul>

<li>Milk</li>

<li>Bread</li>

<li>Eggs</li>

</ul>

### 3.Description List (<dl>)

* Used for listing **terms and their definitions**.
* Uses:
  + <dt> for the **term**
  + <dd> for the **description**

#### Use Case:

* Glossaries
* FAQs
* Term-definition pairs

#### Example:

<dl>

<dt>HTML</dt>

<dd>HyperText Markup Language</dd>

<dt>CSS</dt>

<dd>Cascading Style Sheets</dd>

</dl>

### Nested Lists

* You can place a list **inside another list** (e.g., sub-points).
* Useful for hierarchical or multi-level data.

**Anchor Elements in HTML**

**What is the <a> Tag?**

* The <a> tag (anchor tag) is used to create **hyperlinks** in HTML.
* It allows users to **navigate** from one page to another, or to other websites, email, or sections within a page.

**Basic Syntax**

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

* href stands for **hypertext reference** and holds the URL or address.
* Link Text is the **clickable** text shown on the webpage.

**Common Uses of <a> Tag**

1. **Link to another webpage**

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

1. **Link to another section on the same page** (internal linking)

<a href="#section1">Go to Section 1</a>

1. **Link to an email address**

<a href="mailto:example@email.com">Send Email</a>

1. **Link to a downloadable file**

<a href="file.pdf" download>Download PDF</a>

**Important Attributes of <a>**

| **Attribute** | **Purpose** |
| --- | --- |
| href | URL or link destination |
| target | Where to open the link (\_blank, \_self, etc.) |
| title | Tooltip text shown on hover |
| download | Allows file download instead of opening it |
| rel | Describes the relationship (e.g., nofollow, noopener) |

**target Attribute Values**

| **Value** | **Description** |
| --- | --- |
| \_self | Opens link in the **same tab** (default) |
| \_blank | Opens link in a **new tab or window** |
| \_parent | Opens in the **parent frame** |
| \_top | Opens in the **full body of the window** |

**HTML Image Element (<img>)**

**What is the <img> Tag?**

* The <img> tag is used to **embed images** in an HTML page.
* It is a **self-closing** (void) tag — meaning it **does not need a closing tag**.

**Syntax**

<img src="image.jpg" alt="Description of image">

**Important Attributes**

| **Attribute** | **Purpose** |
| --- | --- |
| src | **(Required)** – Specifies the **image path or URL** (source of the image) |
| alt | **(Required)** – Provides **alternative text** if the image fails to load or for screen readers |
| width | Sets the **width** of the image (in px or %) |
| height | Sets the **height** of the image |
| title | Shows **tooltip** text when hovering over the image |

**Example**

<img src="cat.jpg" alt="A cute cat" width="300" height="200">

This displays an image of a cat with specified width and height. If the image is missing, "A cute cat" is shown as alternative text.

**Image Sources**

You can use images from:

* **Local folders** (src="images/photo.png")
* **Online URLs** (src="https://example.com/logo.png")

**Best Practices**

* Always use the alt attribute for **accessibility** and **SEO**.
* Use optimized image formats like .jpg, .png, .webp.
* Resize large images using width and height for faster loading.
* Avoid stretching/distorting images — maintain **aspect ratio**.

**More HTML Tags You Should Know**

**<br> – Line Break**

* Inserts a **single line break** in the content.
* It’s a **void tag** (no closing tag).

<p>Hello<br>World!</p>

**<hr> – Horizontal Rule**

* Draws a **horizontal line** across the page.
* Used to separate content sections.

<hr>

**<strong> & <b>**

* <strong>: Makes text **bold** and gives it **semantic importance**.
* <b>: Makes text bold **without importance**.

<strong>Important</strong> vs. <b>Just bold</b>

**<em> & <i>**

* <em>: Makes text **italic** and emphasizes it semantically.
* <i>: Makes text italic **without emphasis**.

<em>Important</em> vs. <i>Just italic</i>

**<mark> – Highlight Text**

* Highlights text with a **yellow background** (by default).

<p>This is <mark>important</mark> content.</p>

**<code> – Code Snippets**

* Used to display **inline programming code**.

<p>Use the <code>printf()</code> function in C.</p>

**<pre> – Preformatted Text**

* Displays text **exactly as written** (including spaces and line breaks).

<pre>

Line 1

Line 2 (indented)

Line 3

</pre>

**<blockquote> – Quotation**

* Used to display **quoted text**, usually indented.

<blockquote>

"The journey of a thousand miles begins with one step."

</blockquote>

**<abbr> – Abbreviation**

* Shows the **full form** of an abbreviation on hover.

<abbr title="HyperText Markup Language">HTML</abbr>

**<sup> & <sub>**

* <sup>: Superscript (used in powers, footnotes).
* <sub>: Subscript (used in chemical formulas).

x<sup>2</sup> → x²

H<sub>2</sub>O → H₂O

**HTML <video> Tag**

**What is the <video> Tag?**

* The <video> tag is used to **embed video files** into an HTML document.
* It supports **multiple formats** and provides **playback controls**.

**Basic Syntax**

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

OR (with multiple sources for better compatibility):

<video controls width="400">

<source src="video.mp4" type="video/mp4">

<source src="video.webm" type="video/webm">

Your browser does not support the video tag.

</video>

**Common Attributes of <video>**

| **Attribute** | **Description** |
| --- | --- |
| src | Specifies the path to the video file (optional if using <source> tags) |
| controls | Adds play, pause, volume, etc. controls |
| autoplay | Automatically starts playing the video when the page loads |
| muted | Starts the video without sound |
| loop | Repeats the video automatically when it ends |
| poster | Specifies an image to show before the video plays |
| width/height | Sets the size of the video player |

**Supported Video Formats**

| **Format** | **File Extension** | **MIME Type** |
| --- | --- | --- |
| MP4 | .mp4 | video/mp4 |
| WebM | .webm | video/webm |
| Ogg | .ogg | video/ogg |

Using multiple formats inside <source> ensures **cross-browser compatibility**.

**HTML Comments**

**What are HTML Comments?**

* HTML comments are **non-visible notes** in the source code.
* Browsers **ignore** comments; they are **not shown** on the webpage.
* Used to **explain code**, **leave reminders**, or **temporarily disable elements** during testing.

**Syntax**

<!-- This is a comment -->

Everything between <!-- and --> is treated as a comment.

**Examples**

<!-- This is a single-line comment -->

<p>Hello</p> <!-- Comment after code -->

<!--

This is a multi-line comment.

It can span several lines.

-->

**Use Cases**

| **Use Case** | **Purpose** |
| --- | --- |
| Documenting Code | Explain what a section of code does |
| Hiding Temporary Code | Disable code without deleting it |
| Section Titles | Label parts of your page |
| Team Collaboration Notes | Leave notes for other developers |

**Common Mistakes**

| **Mistake** | **Why It's Wrong** |
| --- | --- |
| <!- comment -> | Incorrect syntax — missing dashes |
| Putting HTML code inside comment | Code inside <!-- --> is ignored |

**Is HTML Case Sensitive?**

**Short Answer:**

**No, HTML is not case sensitive.**

**Explanation:**

* In **HTML**, tags and attributes **can be written in uppercase, lowercase, or mixed case**, and they will still work.

<H1>Heading</H1> <!-- Valid -->

<h1>Heading</h1> <!-- Valid -->

<H1>Heading</h1> <!-- Still works, but not recommended -->

* However, **lowercase** is preferred and recommended because:
  + It follows the **HTML5 specification**.
  + It's consistent with **XHTML** and **XML** (which are case-sensitive).
  + It improves **readability and maintainability** of your code.

**Case Sensitivity Rules in HTML:**

| **Element** | **Case Sensitive?** | **Notes** |
| --- | --- | --- |
| Tags (<p>, <div>) | No | <P>, <p>, <P> all work |
| Attributes (href, src) | No | Works in any case |
| Attribute values (text, email) | Usually No | May depend on specific context |
| File names (src="image.JPG") | Yes (on some servers) | Depends on server (Linux is case-sensitive) |

**Best Practice**

**Always use lowercase** for:

* Tags: <html>, <body>, <img>
* Attributes: href, src, alt

This follows modern HTML standards and avoids confusion.

**What are HTML Elements?**

In HTML, every element behaves in a certain way in the document layout. Based on their behavior, HTML elements are categorized as:

* **Block-level elements**
* **Inline elements**

**Block-level Elements**

* These elements **start on a new line** and **take up the full width** available.
* They create a "block" or "box" around the content.
* Often used to structure the main parts of a web page (e.g., headings, paragraphs, divs).

**Examples**

<div>This is a div (block element)</div>

<p>This is a paragraph</p>

<h1>This is a heading</h1>

<ul><li>List item</li></ul>

**Inline Elements**

* These elements **do not start on a new line**.
* They only take up as much width as necessary.
* Typically used for **styling parts of text** or embedding small items within a block element.

**Examples:**

<span>This is a span (inline element)</span>

<a href="#">This is a link</a>

<strong>Bold text</strong>

<img src="image.jpg" alt="image">

**Key Differences Table**

| **Feature** | **Block Element** | **Inline Element** |
| --- | --- | --- |
| Starts on a new line | Yes | No |
| Takes full width | Yes | No |
| Contains other block tags | Often | Rarely |
| Common usage | Layout, sections, structure | Formatting small portions |
| Examples | <div>, <p>, <h1>, <ul> | <span>, <a>, <img>, <em> |

**What is a <div> Element?**

The <div> tag in HTML stands for **"division"**. It is a **block-level container** used to group elements and apply styles or scripts to them collectively.

It's one of the most commonly used tags in web development for **layout** and **structure**.

**Key Features**

* It is **block-level** — starts on a new line and takes full width.
* It groups multiple HTML elements logically.
* Typically used with **CSS or JavaScript** for styling or dynamic behavior.
* Doesn’t change how content looks by default.

**Basic Example:**

<div>

<h2>This is a Heading</h2>

<p>This is a paragraph inside a div.</p>

</div>

**Use Cases**

| **Use Case** | **Example** |
| --- | --- |
| Grouping content | Group headings, images, and paragraphs into a section |
| Applying CSS styles | Use class or id to apply CSS to a particular section |
| Structuring layout | Used in columns, rows, boxes |
| DOM manipulation | Helps JavaScript target sections for interaction |

### Notes:

* A <div> itself doesn’t provide meaning — it’s **generic**.
* Use **semantic HTML** tags like <section>, <article>, or <main> when meaning matters.

**What is a <span> Element?**

The <span> tag is an **inline container** used to **style or target a part of the text or inline elements** without affecting layout.

Unlike <div>, which is block-level, <span> does not break content into a new line.

**Key Features**

* **Inline-level** element — does **not** start on a new line.
* Used for **styling text** or **scripting specific words**.
* Perfect for **short text snippets** inside paragraphs or headings.

**Basic Example:**

<p>This is a <span style="color: red;">red</span> word.</p>

Output: This is a **red** word.

**Use Cases**

| **Use Case** | **Example** |
| --- | --- |
| Inline text styling | Coloring, bolding, changing font |
| JavaScript targeting | Select a span by id or class |
| Semantic tagging | Highlighting specific phrases |

**What is the <hr> Tag?**

The <hr> tag stands for **Horizontal Rule**.  
It is used to **insert a horizontal line** (a thematic break) across the page.

**Key Features**

* Represents a **thematic break** between paragraph-level elements.
* It is a **self-closing tag** — does **not** need a closing tag.
* Renders a **horizontal line** on the page.
* Treated as a **block-level** element.
* Often used to separate **sections or topics** visually.

**Basic Example:**

<h2>Introduction</h2>

<p>This is the first section.</p>

<hr>

<h2>Next Section</h2>

<p>This is the second section after the horizontal line.</p>

**Notes:**

* HTML5 redefined <hr> to represent a **thematic break** (not just a line).
* It improves **content readability** by visually separating related topics.

**What is Semantic Markup?**

**Semantic HTML (or semantic markup)** uses HTML5 tags that **convey the meaning of the content** within them.

These tags **clearly describe** their purpose, making it easier for:

* Developers to read the code,
* Search engines to understand the structure,
* Screen readers to interpret content for accessibility.

**Why Use Semantic HTML?**

* Improves **code readability**
* Enhances **SEO (Search Engine Optimization)**
* Supports **accessibility**
* Reduces need for excessive <div> and <span> usage (known as “div soup”)

**Common Semantic Tags**

| **Tag** | **Description** |
| --- | --- |
| <header> | Defines the top section of a page or section |
| <nav> | Contains navigation links |
| <main> | Represents the main content area |
| <section> | Defines a section of related content |
| <article> | Represents an independent, self-contained piece of content |
| <aside> | Side content like ads or sidebars |
| <footer> | Bottom section of a page or section |
| <figure> | Wraps images or illustrations with captions |
| <figcaption> | Provides a caption for a <figure> |
| <time> | Represents date or time |

**Non-Semantic vs Semantic**

| **Non-Semantic Tag** | **Semantic Equivalent** |
| --- | --- |
| <div id="header"> | <header> |
| <div id="nav"> | <nav> |
| <div id="content"> | <main> |
| <div id="footer"> | <footer> |

**What are Semantic Tags?**

**Semantic tags** in HTML are tags that **clearly describe their meaning and purpose** in the layout of a web page.

Unlike non-semantic tags like <div> and <span>, semantic tags tell both the **browser** and **developers** what kind of content they contain.

**Benefits of Semantic Tags**

* Improves **code clarity**
* Enhances **SEO (Search Engine Optimization)**
* Supports **screen readers** and **accessibility tools**
* Better organization of page content

**Common Semantic HTML5 Tags and Their Usage**

| **Semantic Tag** | **Description** |
| --- | --- |
| <header> | Defines the **top section** of a document or a section |
| <nav> | Defines a **navigation menu** |
| <main> | Represents the **main content** of the page |
| <section> | Groups related content into **logical sections** |
| <article> | Represents an **independent, self-contained** piece of content |
| <aside> | Defines content **outside the main flow**, like a sidebar |
| <footer> | Defines the **bottom section** of a document or a section |
| <figure> | Wraps **media** content like images with optional captions |
| <figcaption> | Provides a **caption** for a <figure> |
| <summary> | Defines a **summary heading** for <details> |
| <details> | Used to create a **collapsible section** |
| <mark> | Highlights or **marks important text** |
| <time> | Represents **date or time values** |
| <address> | Contains **contact information** for the author |

**What are HTML Entities?**

In HTML, some characters are **reserved** (like <, >, &, "), and cannot be used directly in content because the browser might confuse them with HTML tags or syntax.

**HTML Entities** are **special codes** that allow you to display these reserved characters properly.

**Syntax of an Entity**

&entity\_name; or &#entity\_number;

* & – Starts the entity
* ; – Ends the entity
* You can use either the **entity name** (e.g., &lt;) or **entity number** (e.g., &#60;)

**Commonly Used HTML Entities**

| **Character** | **Entity Name** | **Entity Number** | **Description** |
| --- | --- | --- | --- |
| < | &lt; | &#60; | Less-than symbol |
| > | &gt; | &#62; | Greater-than symbol |
| & | &amp; | &#38; | Ampersand |
| " | &quot; | &#34; | Double quotation mark |
| ' | &apos; | &#39; | Single quotation mark |
| © | &copy; | &#169; | Copyright symbol |
| ® | &reg; | &#174; | Registered trademark |
| ₹ | &#8377; | &#8377; | Indian Rupee symbol |
| (space) | &nbsp; | &#160; | Non-breaking space |

**Example:**

<p>Use &lt;div&gt; instead of &lt;span&gt; for layout.</p>

Output:  
**Use <div> instead of <span> for layout.**

**Non-breaking Space (&nbsp;)**

* Adds a space that **won’t break into a new line**
* Useful when you want to insert multiple spaces (regular spaces collapse in HTML)

<p>This&nbsp;&nbsp;&nbsp;has&nbsp;extra&nbsp;spaces.</p>

**What is Emmet?**

**Emmet** is a **powerful toolkit** (built into most modern code editors like VS Code) that allows you to **write HTML and CSS code faster** using short **abbreviations** that expand into full code snippets.

It greatly boosts your productivity by **reducing repetitive typing**.

**Key Features**

* Type short **abbreviations** and hit Tab (or Enter) to expand them.
* Works in HTML, CSS, and JSX.
* Saves time during development, especially when creating repetitive layouts.

**Basic Emmet Abbreviations (HTML)**

| **Abbreviation** | **Expands To** |
| --- | --- |
| html:5 | Basic HTML5 boilerplate |
| div | <div></div> |
| p | <p></p> |
| ul>li\*3 | Creates a list with 3 items |
| div.container | <div class="container"></div> |
| div#main | <div id="main"></div> |
| input:email | <input type="email" /> |
| a:link | <a href="http://"></a> |

**Nesting Example**

ul>li\*3

Expands to:

<ul>

<li></li>

<li></li>

<li></li>

</ul>

**Grouping and Siblings**

div.header+div.nav+div.content

Expands to:

<div class="header"></div>

<div class="nav"></div>

<div class="content"></div>

**Emmet with Attributes**

input:text[name="username"]

Expands to:

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

## ****HTML Tables****

### What is an HTML Table?

An **HTML table** is used to display data in **rows and columns**, similar to a spreadsheet or database.

### ****Basic Table Structure****

<table>

<tr>

<th>Header 1</th>

<th>Header 2</th>

</tr>

<tr>

<td>Data 1</td>

<td>Data 2</td>

</tr>

</table>

* <table>: Starts the table.
* <tr>: Table row (horizontal).
* <th>: Table header cell (bold & centered by default).
* <td>: Table data cell.

### ****Key Table Tags****

| **Tag** | **Meaning** |
| --- | --- |
| <table> | Starts the table |
| <tr> | Table row |
| <td> | Table data |
| <th> | Table header |
| <thead> | Table header section |
| <tbody> | Table body section |
| <tfoot> | Table footer section |
| colspan | Merge columns |
| rowspan | Merge rows |

### ****Attributes in**** <table>

| **Attribute** | **Description** |
| --- | --- |
| border | Sets the border width |
| cellspacing | Space between cells |
| cellpadding | Space inside cell |
| width | Table width |
| height | Table height |
| align | Alignment (left, right, center) |

### ****Special: Merging Cells****

#### colspan Example

<td colspan="2">Merged Cell</td>

→ Merges two columns into one.

#### rowspan Example

<td rowspan="2">Merged Cell</td>

→ Merges two rows into one.

**Semantics in HTML Tables**

**What is Semantics?**

**Semantics** in HTML means using tags that clearly describe their **meaning and purpose** to both the browser and developers.

In the context of **tables**, semantic tags help structure the data meaningfully and **improve accessibility**, especially for screen readers and assistive technologies.

**Semantic Table Tags**

| **Tag** | **Description** |
| --- | --- |
| <table> | The main container for tabular data |
| <thead> | Groups the header rows of a table |
| <tbody> | Groups the body (main content) rows |
| <tfoot> | Groups the footer rows |
| <tr> | Table row |
| <th> | Table header cell (bold and centered by default) |
| <td> | Table data cell |
| scope | Attribute used on <th> to define whether it applies to a row or column |

**Why Use Semantic Tags in Tables?**

1. **Accessibility**  
   Screen readers can understand data better (e.g., <th scope="col"> tells the screen reader it's a column header).
2. **Clean Structure**  
   Separating headers (<thead>), body (<tbody>), and footers (<tfoot>) organizes code better.
3. **Improved SEO & Indexing**  
   Search engines can understand and categorize your content more accurately.

**Important Semantic Attribute: scope**

Used in <th> to define if the header relates to a **row** or a **column**:

| **Usage** | **Example** |
| --- | --- |
| scope="col" | Applies to a column header |
| scope="row" | Applies to a row header |

<th scope="row">Total</th>

<th scope="col">Price</th>

## ****HTML Forms – Complete Notes****

### 1. ****What is an HTML Form?****

An HTML **form** is used to collect user input. Data entered in a form is usually sent to a server for processing.

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

<!-- form elements go here -->

</form>

* action: URL where the form data is sent.
* method: HTTP method (GET or POST).

### 2. ****Input Element (****<input>****)****

The <input> tag is the most used form element. It supports different type attributes:

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

| **Type** | **Purpose** |
| --- | --- |
| text | Single-line input |
| email | Email address |
| password | Password input |
| number | Numbers only |
| date | Date picker |
| range | Slider |
| checkbox | Multiple selections |
| radio | Single selection from group |
| submit | Submit form |

### 3. ****Placeholders and Labels****

#### Placeholder

Displays **hint text** inside the input field.

<input type="text" placeholder="Enter your name">

#### Label

Describes what the input is for and improves accessibility.

<label for="username">Username:</label>

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

### 4. ****Button Element****

Used to **submit**, **reset**, or **trigger actions**.

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

<button type="reset">Reset</button>

<button type="button">Click Me</button> <!-- No form action -->

### 5. ****Name Attribute****

The name attribute is **important for sending form data** to the backend. It acts as a key in the key-value pair.

<input type="text" name="email">

### 6. ****Checkbox Input****

Used to allow **multiple selections**.

<label><input type="checkbox" name="interest" value="coding"> Coding</label>

<label><input type="checkbox" name="interest" value="music"> Music</label>

* Use the **same name** if you want to group checkboxes together.

### 7. ****Radio Input****

Used for **single selection** from a group.

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

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

* **Only one** radio button in the group (same name) can be selected.

### 8. ****Select Dropdown****

Creates a **drop-down menu**.

<label for="city">Choose City:</label>

<select name="city" id="city">

<option value="pune">Pune</option>

<option value="mumbai">Mumbai</option>

<option value="delhi">Delhi</option>

</select>

* You can use multiple attribute to allow multi-selection.

### 9. ****Range Input (Slider)****

Used to select a numeric value within a range.

<label for="volume">Volume:</label>

<input type="range" id="volume" name="volume" min="0" max="100">

* Add step="10" to control the jump.

### 10. ****Text Area****

Used for **multi-line text** input.

<label for="message">Your Message:</label><br>

<textarea id="message" name="message" rows="5" cols="30"></textarea>