



HTML(Hyper Text Markup Langauge)

Shorts Notes



Er. Rajesh Prasad(B.E, M.E)
Founder: TWF & RID Org.

- **RID ORGANIZATION** यानि **Research, Innovation and Discovery** संस्था जिसका मुख्य उद्देश्य हैं आने वाले समय में सबसे पहले **NEW (RID, PMS & TLR)** की खोज, प्रकाशन एवं उपयोग भारत की इस पावन धरती से भारतीय संस्कृति, सभ्यता एवं भाषा में ही हो।
- देश, समाज, एवं लोगों की समस्याओं का समाधान **NEW (RID, PMS & TLR)** के माध्यम से किया जाये इसके लिए ही मैं राजेश प्रसाद **इस RID संस्था** की स्थपना किया हूँ।
- Research, Innovation & Discovery में रुचि रखने वाले आप सभी विधार्थियों, शिक्षकों एवं बुधीजिवियों से मैं आवाहन करता हूँ की आप सभी **इस RID संस्था** से जुड़ें एवं अपने बुद्धि, विवेक एवं प्रतिभा से दुनियां को कुछ नई **(RID, PMS & TLR)** की खोजकर, बनाकर एवं अपनाकर लोगों की समस्याओं का समाधान करें।

त्वक्सा HTML के इस ई-पुस्तक में आप HTML से जुड़ी सभी बुनियादी अवधारणाएँ सीखेंगे। मुझे आशा है कि इस ई-पुस्तक को पढ़ने के बाद आपके ज्ञान में वृद्धि होगी और आपको कंप्यूटर विज्ञान के बारे में और अधिक जानने में रुचि होगी।

“In this E-Book of TWKSAA HTML you will learn all the basic concepts related to HTML. I hope after reading this E-Book your knowledge will be improve and you will get more interest to know more thing about computer Science”.

Online & Offline Class:

Python, Web Development, Java, Full Stack Course, Data Science, UI/UX Training, Internship & Research

करने के लिए Message/Call करें. 9202707903 E-Mail_id: ridorg.in@gmail.com

Website: www.ridtech.in

RID हमें क्यों करना चाहिए ?

<u>(Research)</u>	<u>(Innovation)</u>	<u>(Discovery)</u>
अनुसंधान हमें क्यों करना चाहिए ? Why should we do research? 1. नई ज्ञान की प्राप्ति (Acquisition of new knowledge) 2. समस्याओं का समाधान (To Solving problems) 3. सामाजिक प्रगति (To Social progress) 4. विकास को बढ़ावा देने (To promote development) 5. तकनीकी और व्यापार में उन्नति (To advances in technology & business) 6. देश विज्ञान और प्रौद्योगिकी के विकास (To develop the country's science & technology)	नवीनीकरण हमें क्यों करना चाहिए ? Why should we do Innovation? 1. प्रगति के लिए (To progress) 2. परिवर्तन के लिए (For change) 3. उत्पादन में सुधार (To Improvement in production) 4. समाज को लाभ (To Benefit to society) 5. प्रतिस्पर्धा में अग्रणी (To be ahead of competition) 6. देश विज्ञान और प्रौद्योगिकी के विकास (To develop the country's science & technology)	खोज हमें क्यों करना चाहिए ? Why should we do Discovery? 1. नए ज्ञान की प्राप्ति (Acquisition of new knowledge) 2. अविष्कारों की खोज (To Discovery of inventions) 3. समस्याओं का समाधान (To Solving problems) 4. ज्ञान के विकास में योगदान (Contribution to development of knowledge) 5. समाज के उन्नति के लिए (for progress of society) 6. देश विज्ञान और तकनीक के विकास (To develop the country's science & technology)

Definition: - www is a global collection of documents and other resources linked by hyperlink and URLs. It is known as web, it is an information system technology enabling.

History: - computer scientist "Tim Berners Lee" at CERN {(European Organization for nuclear Research) it is Intergovernmental org. established in 1954} invented in 1989. 1st proposal was written & working system implemented by end of 1990 including www Browser & http server.

Function: - 1). HTML 2). Linking 3). www prefix 4). Scheme specifiers 5). Web Page 6). Website 7). Browser 8). Search Engine

HTML: - Hypertext Markup Language it used for Creating Web page & Web Application.

Linking: - it is interconnecting the web page via Hyperlinks.

www prefix: - it is like .com, .org, .net etc. **Scheme specifiers:** - http:// or https://

Browser: - it is a software responsible for open the website

Web Page: - A webpage is an HTML document on the WWW. **Website:** - it is a collection of web page.

Search Engine: - it is a software program/system Software Design to carry out the web search.

use: - it is used to search www in a systematic way for particular



Sir Tim Berners Lee

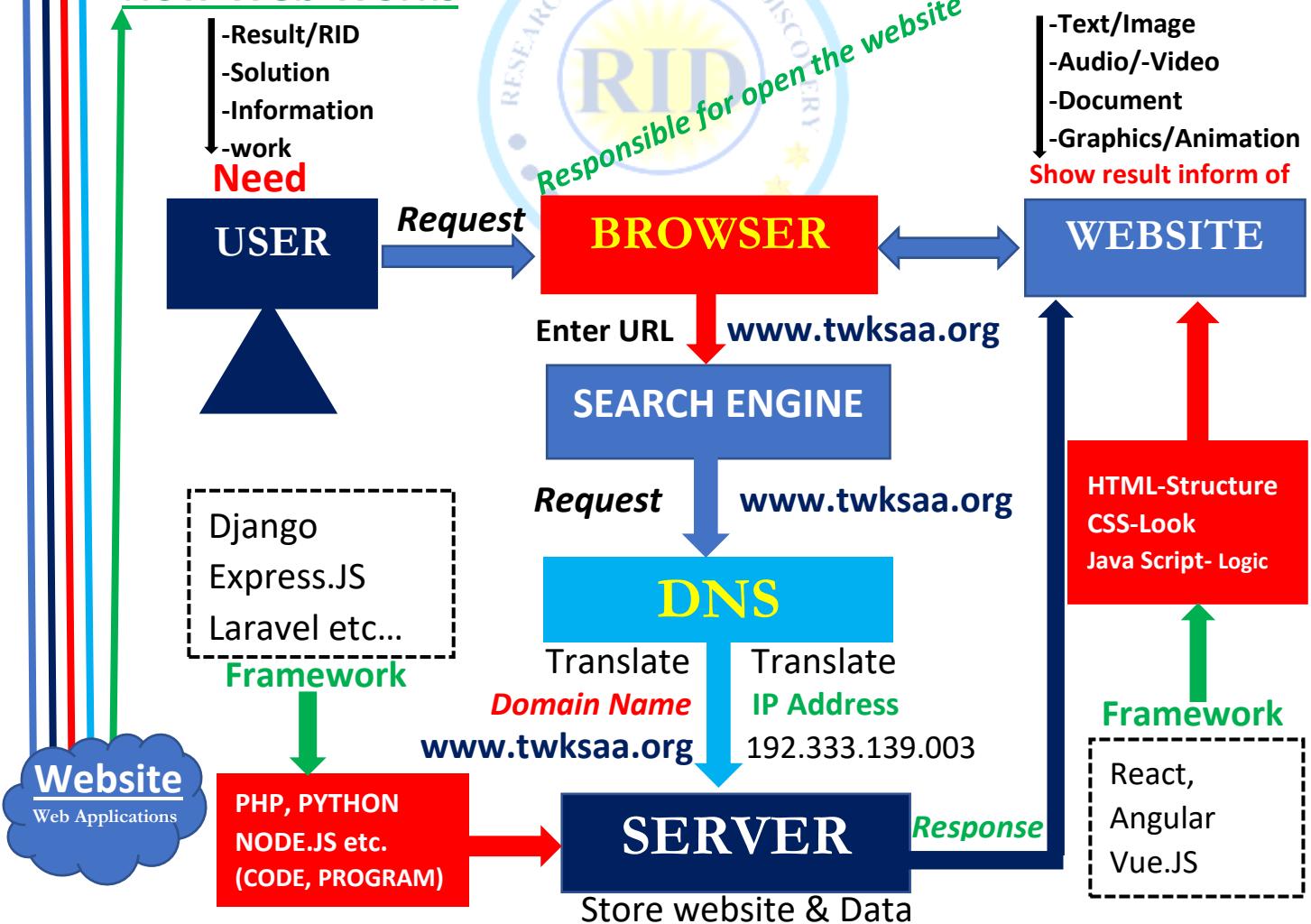
Definition: - website is collection of web page and related content that is identified by a common "Domain name"

Types: - 1). Static Website 2). Dynamic Website

Static Website: - consists of a series of HTML files, each one representing a physical page of a website.

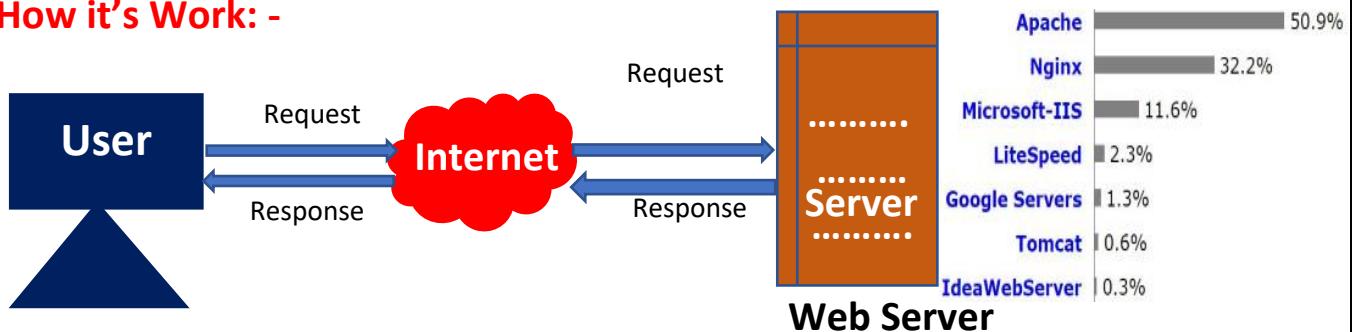
Dynamic Website: - changes or customizes itself frequently and automatically.

How Web Works



- 1) **Web:** it is a part of the Internet that lets people access and share information through websites.
- 2) **Site:** means location or a collection of web pages hosted on a web server and accessible through a specific domain.
- 3) **Page:** A page refers to a single, individual document or resource on the web.
- 4) **Web Page:** A web page is a single HTML document on the web that can be viewed in a web browser.
- 5) **Hyper Text:** It means text that has links (connections) to other text or information, going beyond normal text.
- 6) **Website:** web site is a collection of related web pages.
- 7) **Web Server:** A web server is a computer or program that stores and delivers web pages to users on the Internet.

How it's Work: -



WEB PAGE

- A web page is a single HTML document shown in a web browser.
- It can have text, images, videos, or links.
- Web pages are made using HTML, which gives the structure and content of the page.

❖ TYPES OF WEBS PAGES

There are two types of web pages:

1. **Static Page:**
 - Shows the same information every time.
 - Content does not change for different users.
 - File extensions: .htm, .html
2. **Dynamic Page:**
 - Content changes based on user request or input.
 - Different users may see different information.

Note: Dynamic Web Page Extensions: .php (PHP pages), .asp (Active Server Pages), .aspx (ASP.NET pages), .jsp (Java Server Pages), .ejs (Embedded JavaScript templates in Node.js), .jsx (React JavaScript XML pages).

HTML (Hyper Text Markup Language)

What is HTML?

- HTML stands for Hyper Text Markup Language.
 - It is the standard language for creating web pages.
 - HTML defines the structure of a web page.
 - HTML elements tell the browser how to display content (heading, paragraph, link, etc.).
 - It is a markup language, not a programming language.
 - HTML is not case sensitive.
- 1) **Hyper Text:** Text that contains links to other text.
 - 2) **Markup Language:** A way to format and structure text for the web.
 - 3) **Developer:** Created by Sir Tim Berners-Lee in 1990 at CERN.
 - 4) **Maintained by** W3C (World Wide Web Consortium).
 - 5) **Latest version: HTML5.**
 - 6) **HTML Editors** (examples): Notepad, Notepad++, **VS Code**, Sublime Text, Atom, Brackets, Dreamweaver, Pinegrow, etc.

❖ Features of HTML

1. **Markup Language:** Uses tags to define structure and content.
2. **Document Structure:** Provides basic layout with `<html>`, `<head>`, `<body>`.
3. **Hyperlinks:** Connects pages using `<a>` tag.
4. **Text Formatting:** Supports headings, paragraphs, bold, italic, etc.
5. **Lists:** Supports ordered ``, unordered ``, and definition lists `<dl>`.
6. **Images & Multimedia:** Allows images, audio, and video embedding.
7. **Forms:** Provides input fields, buttons, and form elements.
8. **Semantic Elements:** HTML5 adds `<header>`, `<section>`, `<article>`, etc.
9. **Cross-Browser:** Works across different browsers.
10. **Embedded Scripting:** Allows JavaScript for interactivity.

Tag in HTML

- **Tags in HTML:** Tags are **special keywords** written inside angle brackets `<>` that tell the browser how to display content.
Example: `<p>` for paragraph, `<h1>` for heading, `<a>` for link.
- **Note:** - Tags are **instructions** for the browser to format and show web page content.

HTML Page Structure

```
<!DOCTYPE html>           ← Tells version of HTML
<html>                  ← HTML Root Element
<head>                  ← Used to contain page HTML metadata
  <title>Page Title</title> ← Title of HTML page
</head>

<body>                  ← Hold content of HTML
  <h2>Heading Content</h2> ← HTML heading tag
  <p>Paragraph Content</p> ← HTML paragraph tag
</body>

</html>
```

Structure of an HTML Page

Every HTML page has two main parts:

1. Document Declaration

- Tells browser which version of HTML is used.
- For HTML5, we write at the top:
- `<!DOCTYPE html>`

2. Document Scope

- Defined with `<html> ... </html>` tags.
- lang attribute defines language of page.
Example: `<html lang="en">`

1. Body Section `<body> ... </body>`

Contains actual content of the page (text, images, links, etc.) that is displayed in browser.

Inside Document Scope

1. **Head Section `<head> ... </head>`**
Contains information *about* the page (not shown on the page itself).
 - `<meta>` → Metadata (like charset, viewport, SEO info).
 - `<title>` → Title shown in browser tab.
 - `<link>` → Link external files (like CSS).
 - `<style>` → Add internal CSS styles.
 - `<script>` → Add JavaScript code.

```
index.html X
index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |  <meta charset="UTF-8">
5  |  <title>My Page</title>
6  |  <link rel="stylesheet" href="style.css">
7  </head>
8  <body>
9  |  <h1>Hello World</h1>
10 |  <p>This is my first web page.</p>
11 </body>
12 </html>
```

<meta charset="UTF-8"> ensures your web page can display **any language or symbol** properly.

1. ASCII (American Standard Code for Information Interchange):

- Represents characters using **7 or 8 bits**.
- Only supports **English letters, digits, and basic symbols**.
- Example: A → 65, a → 97.
- Cannot handle characters from other languages like Hindi, Chinese, or Arabic.

2. UTF-8 (Unicode Transformation Format - 8 bit):

- Can represent **all characters from all languages**.
- Uses **1 to 4 bytes** per character depending on the symbol.
- Backward compatible with ASCII (first 128 characters are same).
- Example: A → 65, अ → E0 A4 85.
- **ASCII** → **limited to English**.
- **UTF-8** → **universal, supports almost all languages and symbols**.

Building blocks of HTML

- 1. Tags:** it **keywords** written inside angle brackets `< >` that tell the browser how to display content.

Example: `<h1>.....</h1>`, `<div>.....</div>`, `<p>.....</p>`

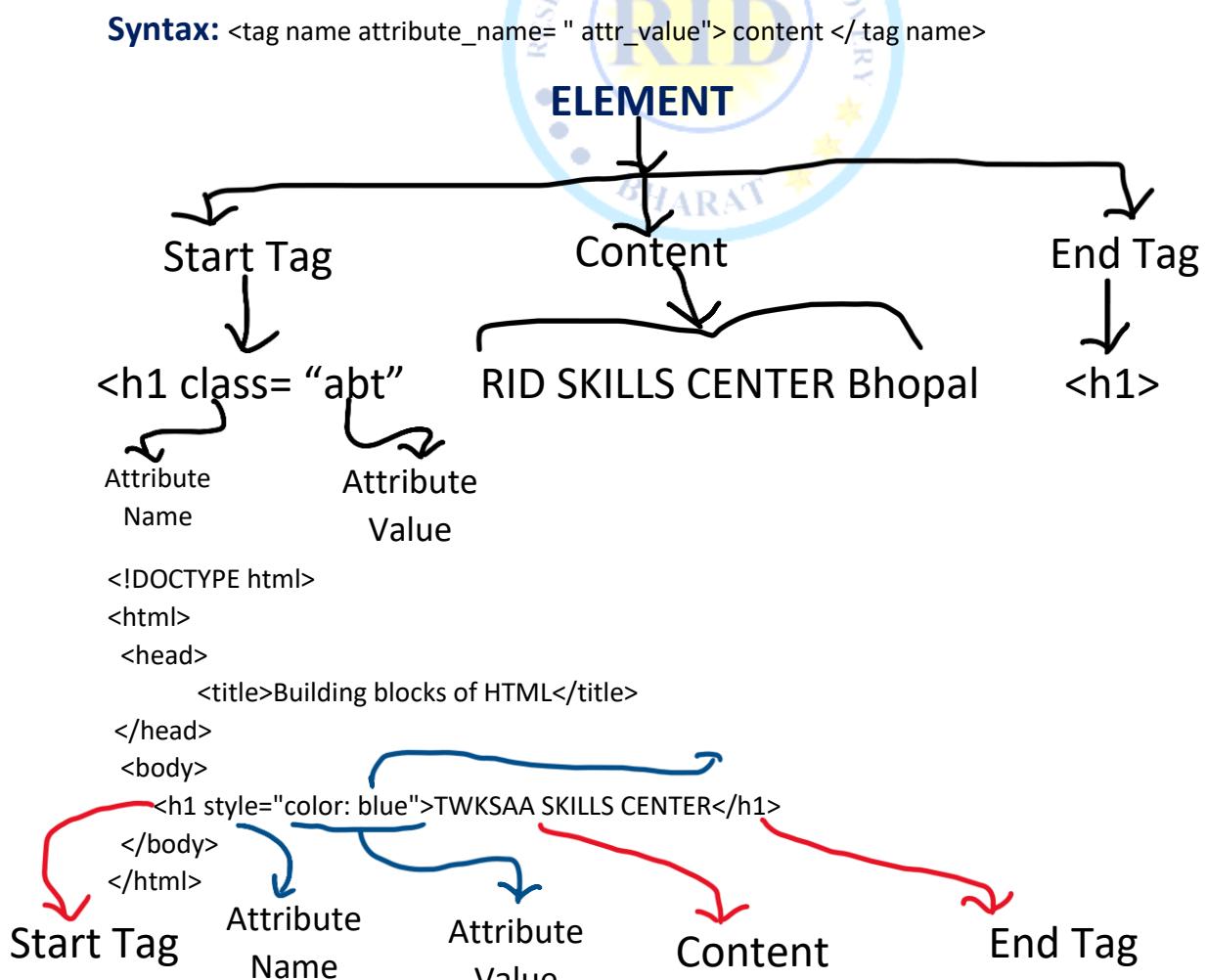
- 2. Attribute:** it is extra information about an HTML element and is written inside the start tag. It has two parts: name and value.

Example: `` src is the attribute name, "image.jpg" is its value.

- 3. Element:** it is a single part of an HTML page, including the start tag, content, and end tag.

Example: `<p>Hello World</p>` → The `<p>` tag, text, and `</p>` together form a paragraph element.

Syntax: `<tag name attribute_name= " attr_value"> content </ tag name>`



HTML Code editor

You can write HTML using **text editors, code editors, or IDEs**. Common ones:

- **Visual Studio Code (VS Code)**: Free, extensible, supports HTML, CSS, JavaScript.
- **Sublime Text**: Fast, lightweight, supports plugins.
- **Atom**: Free, customizable, good for web development.
- **Notepad++**: Lightweight editor with syntax highlighting.
- **Brackets**: Live preview for HTML, CSS, JS.
- **Dreamweaver**: Visual design + code editor (paid).
- **Vim & Emacs**: Powerful text-based editors for advanced users.
- **Online Editors**: CodePen, JSFiddle, Repl.it for quick testing.

➤ Writing HTML in an Editor (Simple Steps)

1. **Open the Editor**: VS Code, Notepad, Sublime Text, etc.

2. **Create New File**: File → New

3. **Write HTML Code**:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>My First Web Page</title>
</head>
<body>
  <h1>Hello, World!</h1>
  <p>This is a simple HTML page.</p>
</body>
</html>
```

4. **Save File**: Use .html extension, e.g., index.html.

5. **Preview**: Open the file in a web browser to see your page.

6. **Edit & Update**: Make changes in editor → save → refresh browser.

Note:

- Editors like **VS Code** and **Sublime Text** offer **auto-completion** and **formatting**.
- Use **Live Server** in VS Code to see changes instantly.
- Notepad is simple, no extra features, but works for basic HTML.

HTML Tags

- HTML tags are **keywords** in < > that tell the browser how to display content.
- **Syntax**: <tag> content </tag>

Types of HTML Tags

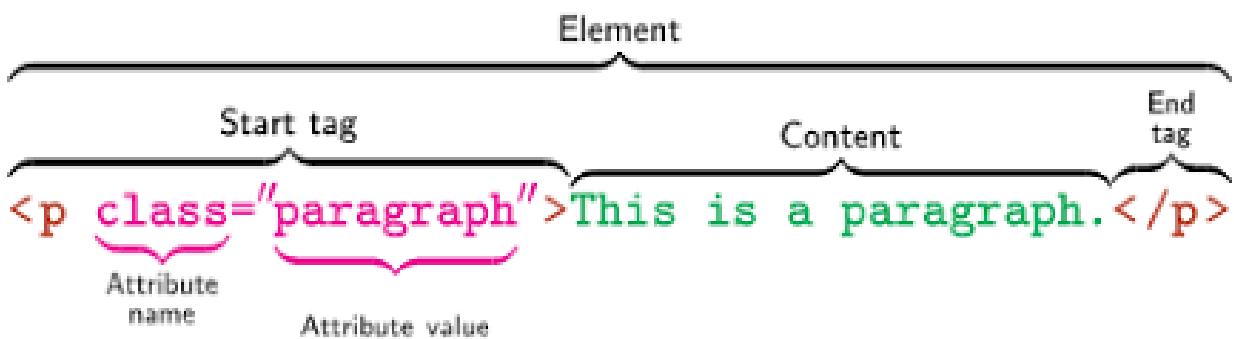
1. **Meta Tags**: <!DOCTYPE>, <title>, <link>, <meta>, <style>
2. **Text Tags**: <p>, <h1>–<h6>, , , <abbr>, <code>,
, etc.
3. **Link Tags**: <a>, <base>
4. **Image & Object Tags**: , <area>, <map>, <param>, <object>
5. **List Tags**: , , , <dl>, <dt>, <dd>
6. **Table Tags**: <table>, <tr>, <td>, <th>, <thead>, <tbody>, <tfoot>, <col>, <caption>
7. **Form Tags**: <form>, <input>, <textarea>, <select>, <option>, <button>, <label>, <fieldset>, <legend>
8. **Scripting Tags**: <script>, <noscript>

HTML Attributes: Provide **extra information** about an element. Written inside **start tag** in **name="value"** format.

Examples: Image path:

Image size:

HTML Elements



- **HTML element** is everything from the **start tag** to the **end tag**.
- **Syntax:** <tagname>Content</tagname>
- **Example:**

```
<h1>RID SKILLS CENTER</h1>
<p>TWKSAA WIT CENTER</p>
<h3>TWKSAA RID CENTER</h3>
```

Empty HTML Elements

- Elements with **no content** are called **empty or void elements**.
- Examples:
 (line break), <hr> (horizontal line)

Nested HTML Elements: - HTML elements can **contain other elements**.

❖ Block-level Elements

- Start on a **new line** and take **full width** of the page.
- Can contain **block-level and inline elements**.
- **Ex:** <div>, <p>, <h1>–<h6>, <section>, <article>, , , <table>, <footer>, <header>

Example:

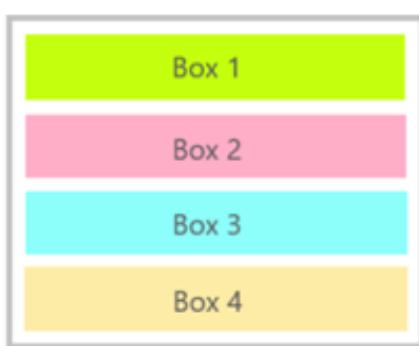
```
<div style="background-color: green"> WIT CENTER</div>
<p style="background-color: red"> RID CENTER</p>
```

❖ Inline Elements

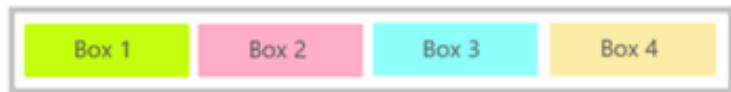
- Do **not start a new line**, width depends on content.
- Usually used **inside block elements**.
- **Examples:** <a>, , , <i>, , <input>,

Example:

```
<a href="https://www.twksaa.org">Click on link</a>
<span style="background-color: lightblue">This is inline element</span>
<p style="color: brown;">This takes width of text only</p>
```



Block-Level Elements



Inline Elements

HTML Headings, Paragraphs, and related tags

❖ HTML Heading Tags

- Used for **titles or subtitles** on a webpage.
- Defined with `<h1>` to `<h6>`:
 - `<h1>` → most important
 - `<h6>` → least important

➤ Example:

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
.....
<h6>Heading 6</h6>
```

❖ HTML Paragraph Tag `<p>`

- Defines a **paragraph**.
- Starts on a **new line**.
- Extra spaces/lines are ignored by the browser.

➤ Example: `<p>Research, Innovation and Discovery. </p>`

❖ HTML Preformatted Text `<pre>`

- Preserves **spaces and line breaks**.
- Uses **fixed-width font**.

➤ Example: `<pre>Research, Innovation and Discovery. </pre>`

❖ Line Break `
` and Horizontal Line `<hr>`

- `
` → adds a **line break** inside text or paragraph.
- `<hr>` → adds a **horizontal line** to separate content.

➤ Example:

```
<p> SKILLS CENTER <br> WIT CENTER <br> RID CENTER</p> <hr>
<pre>Research, Innovation and
Discovery.</pre>
<p>This is LED Based Skill Center</p> <hr>
<p>NEW RID <br> NEW PMS <br> NEW TLR</p> <hr>
```

HTML TEXT FORMATTING

HTML Text Formatting: - Make text look better without using CSS.

Two types of tags:

- **Physical tags:** Change the appearance of text.
- **Logical tags:** Add meaning or importance to text.

Example:

Common Text Formatting Tags

Tag	Type	Use / Meaning
1) <code></code>	Physical	Makes text bold
2) <code></code>	Logical	Marks text as important
3) <code><i></code>	Physical	Makes text italic
4) <code></code>	Logical	Marks text as emphasized / important (italic)
5) <code><mark></code>	Physical	Highlights text
6) <code><big></code>	Physical	Makes text slightly bigger
7) <code><small></code>	Physical	Makes text slightly smaller
8) <code><u></code>	Physical	Underlines text
9) <code></code>	Logical	Shows text as deleted
10) <code><ins></code>	Logical	Shows text as added
11) <code><sub></code>	Physical	Displays text below the line
12) <code><sup></code>	Physical	Displays text above the line

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>RID</title>
</head>
<body>
  <p>1. RID SKILLS CENTER </p>
  <p><b>2. RID SKILLS CENTER</b> </p>
  <p><strong>3. RID SKILLS CENTER</strong></p>
  <p><i>4. RID SKILLS CENTER</i></p>
  <p><em>5. RID SKILLS CENTER</em></p>
  <p><mark>6. RID SKILLS CENTER</mark></p>
  <p><big>7. RID SKILLS CENTER</big></p>
  <p><small>8. RID SKILLS CENTER</small></p>
  <p><u>9. RID SKILLS CENTER</u></p>
  <p><del>10. RID SKILLS CENTER</del></p>
  <p><ins>11. RID SKILLS CENTER</ins></p>
  <p><sub>12. X</sub>2 + <sub>Y</sub>2</p>
  <p><sup>13. X</sup>2 + <sup>Y</sup>2</p>
  <!-- Math formula -->
  <p>(<sup>X</sup> + <sup>Y</sup>)<sup>2</sup> = <sup>X</sup>2 + <sup>Y</sup>2 + 2XY</p>
</body></html>
```

1. Physical Tags

- **Definition:** These tags **only change how the text looks** on the webpage.
- **Purpose:** They are about **visual appearance**, not meaning.

Real-time example:

- On a website, you want to **bold the name of a course** just to make it stand out.

<p>Enroll in Python Programming today!</p>

- Here just makes the text bold. It doesn't say that the text is important semantically.

Other examples: <i> (italic), <u> (underline), <big>, <small>

2. Logical Tags

- **Definition:** These tags **give meaning or importance** to the text, not just style.
- **Purpose:** They help **browsers, search engines, or screen readers** understand the importance of the text.

Real-time example:

- You want to **highlight an important announcement** on a website for search engines and accessibility:

<p>Admission closes on 10th September!</p>

- Here makes the text bold and also tells the browser this text is important.

Other examples: (emphasized), <ins> (added content), (deleted content)

HTML COMMENT TAG

- You can add comments to your HTML source by using the following syntax:
- <!-- Write your comments here -->
- Notice that there is an exclamation point (!) in the start tag, but not in the end tag.
- Comments can be used to hide content.

Note: Comments are not displayed by the browser.

- **Example Hide a section of HTML code:**

```
<p>This is a paragraph. </p>
<!--
<p>Look at this cool image:</p>

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

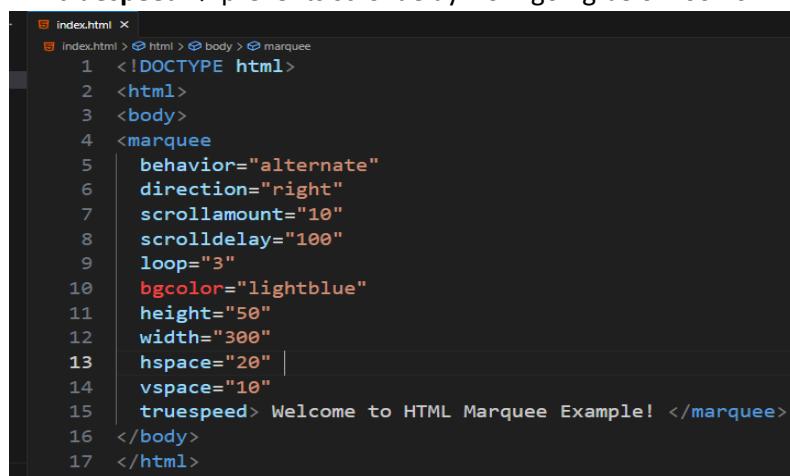


<marquee> Tag in HTML

- Used to create **scrolling text or images** on a webpage.
- Note: <marquee> is **deprecated** in HTML5. Use CSS animations instead.*

Attributes of <marquee>

- behavior** → scrolling style scroll (default, continuous), slide (stops at end), alternate (bounces back and forth)
- direction** → left, right, up, down
- scrollamount** → speed (pixels per step, higher = faster)
- scrolldelay** → delay between moves (in ms, lower = smoother)
- loop** → number of repeats (-1 = infinite)
- bgcolor** → background color
- height** → height of marquee area
- width** → width of marquee area
- hspace** → horizontal space around marquee
- vspace** → vertical space around marquee
- truespeed** → prevents scrolldelay from going below 60ms



```
index.html
index.html > HTML > body > marquee
1  <!DOCTYPE html>
2  <html>
3  <body>
4  <marquee
5    behavior="alternate"
6    direction="right"
7    scrollamount="10"
8    scrolldelay="100"
9    loop="3"
10   bgcolor="lightblue"
11   height="50"
12   width="300"
13   hspace="20" |
14   vspace="10"
15   truespeed> Welcome to HTML Marquee Example! </marquee>
16 </body>
17 </html>
18
```

HTML STYLES

- Style** attribute is used to add **CSS styles** to HTML elements like color, font, size, etc.

Syntax: <tagname style="property: value;">

- property** → CSS property, value → CSS value

1. Background Color:-

Use background-color to change the background color of an element.

```
<body style="background-color: blue;">
<h1>RID SKILLS CENTER</h1>
<p style="color: white;">This is LED Based Skill Center</p>
<h2 style="background-color: yellow;">RID RID CENTER</h2>
<p style="background-color: aqua;">New (RID, PMS, TLR)</p>
<h2 style="background-color: yellow; color: red;">RID RID CENTER</h2>
</body> </html>
```

2. Text Color:-

Use color to change the text color.

3. Font Family :-

Use font-family to change the font style of text.

- <h1 style="font-family: verdana;">This is a heading</h1>
- <p style="font-family: courier;">This is a paragraph.</p>

4. Font Size:-

Use font-size to change the size of text.

- <h1 style="font-size:250%;">Big Heading</h1>

HTML Anchor Tag (<a>)

anchor tag is used to create links (hyperlinks).

With it, we can link:

- One page to another page
- A page to a file, location, or any website (URL)

Note: **href attribute** is the most important part. It tells the browser **where the link should go**.

Syntax: Link Text

Example: Click for Third Page

Using the target Attribute The target attribute defines **where the link will open**.

Examples:

```
<a href="https://www.ridbharat.com/" target="_blank">Open in New Tab</a>
<a href="https://www.ridbharat.com/" target="_self">Open in Same Tab</a>
<a href="https://www.ridbharat.com/" target="_parent">Open in Parent Frame</a>
<a href="https://www.ridbharat.com/" target="_top">Open in Full Window</a>
<a href="https://www.ridbharat.com/">Open Normally</a>
```

Types of targets:

1. **_blank** → Opens link in a **new tab/window**
2. **_self** → Opens link in the **same page (default)**
3. **_parent** → Opens link in the **parent frame**
4. **_top** → Opens link in the **full window**, breaking out of frames

Appearance of Anchor Tag

- **Unvisited link** → Blue & underlined
- **Visited link** → Purple & underlined
- **Active link (clicked)** → Red & underlined

Link, Hyperlink, Hypertext, Hypermedia

1. **Link**: - it is just a **connection** between two resources (like one page to another).

Example: Go to Page 2

2. **Hyperlink**: - A **clickable element** (text or image) that takes you somewhere else.

Example: About Us <!-- or with image -->
a href="gallery.html">

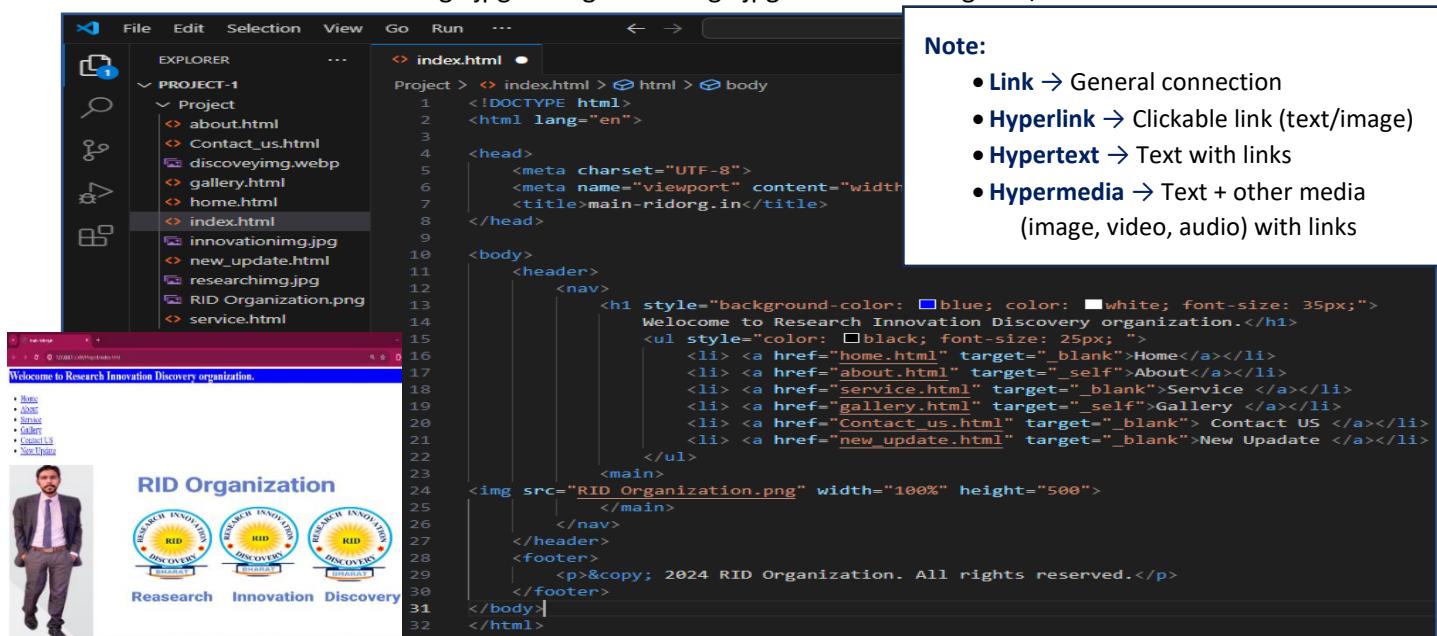
3. **Hypertext**: - Text that contains **links** to other parts of a document or other pages.

Example: <p>Read more in Chapter 2</p>

4. **Hypermedia**: Extends hypertext — includes **links in text, images, videos, audio** etc.

Example: Watch Video


```
<a href="image.jpg"></a>
```



```

<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>main-ridorg.in</title>
  </head>
  <body>
    <header>
      <nav>
        <ul style="color: black; font-size: 25px; ">
          <li> <a href="home.html" target="_blank">Home</a></li>
          <li> <a href="about.html" target="_self">About</a></li>
          <li> <a href="service.html" target="_blank">Service </a></li>
          <li> <a href="gallery.html" target="_self">Gallery </a></li>
          <li> <a href="Contact_us.html" target="_blank"> Contact US </a></li>
          <li> <a href="new_update.html" target="_blank">New Update </a></li>
        </ul>
      </nav>
    </header>
    <main>
      
    </main>
    </header>
    <footer>
      <p>© 2024 RID Organization. All rights reserved.</p>
    </footer>
  </body>
</html>

```

Note:

- **Link** → General connection
- **Hyperlink** → Clickable link (text/image)
- **Hypertext** → Text with links
- **Hypermedia** → Text + other media (image, video, audio) with links

HTML IMAGE, VIDEO AND AUDIO TAG

- img tag is used to display image on the web page. tag is empty, it contains attributes only, and does not have a closing tag.

Syntax:

❖ Attributes of HTML img tag:

- 1) **src:** -describes the source or path of the image.
- 2) **alt:** -The alt attribute defines an alternate text for the image, if it can't be displayed.
- 3) **width:** - used to specify the width to display the image.
- 4) **height:** - height of the image.
- 5). **Style:** - Apply Style

Ex-1:

Ex-2:
- You can also specify the background image in the <style> element, in the <head> section.

<video> Tag

➤ **Defines video content on a web page**

- **src:** Specifies the URL of the video file.
- **controls:** Specifies that video controls should be displayed (play, pause, volume, etc.).
- **autoplay:** Specifies that the video should start playing automatically.
- **loop:** Specifies that the video should start over again when it reaches the end.
- **width:** Specifies the width of the video player.
- **height:** Specifies the height of the video player.
- **poster:** Specifies an image to be displayed before the video starts playing.
- **Muted:** it is used for mute the sound.

Example: <video src="video.mp4" controls width="300" height="250" loop poster="E.png" autoplay="autoplay" muted></video>

Example: <video controls width="300" height="250" loop poster="E.png" autoplay muted><source src="video.mp4" type="video/mp4"></video>

❖ **Audio Tag: Defines sound content on a web page.**

Attributes:

- **src:** Specifies the URL of the audio file.
- **controls:** Specifies that audio controls should be displayed (play, pause, volume, etc.).
- **autoplay:** Specifies that the audio should start playing automatically.
- **loop:** Specifies that the audio should start over again when it reaches the end.

Example: <audio src="audio.mp4" controls autoplay loop></audio>

HTML TABLE TAG

HTML table tag is used to display data in tabular form (row * column).

- 1) **<table>** : It defines a table.
- 2) **<tr>** : It defines a row in a table.
- 3) **<th>** : It defines a header cell in a table.
- 4) **<td>** : It defines a cell in a table.
- 5) **<caption>** : It defines the table caption.
- 6) **<col group>** : It specifies a group of one or more columns in a table for formatting.
- 7) **<tbody>** : It is used to group the body content in a table.
- 8) **<thead>** : It is used to group the header content in a table.
- 9) **<tfooter>** : It is used to group the footer content in a table.
- 10) **<colspan>** : To make a cell span over multiple columns, use the colspan attribute:
- 11) **<rowspan>** : To make a cell span over multiple rows, use the rowspan attribute:
- 12) **<col>** : it is used with <col group> element to specify column properties for each column.

index.html

```

<!DOCTYPE html><html><head><meta charset="UTF-8"><title>HTML Table Example</title>
<style>
  table { border-collapse: collapse; width: 520px; }
  th, td { border: 1px solid #333; padding: 6px; text-align: left; }
  caption { font-weight: bold; margin-bottom: 6px; }
</style></head><body>
<table border="1">
  <caption>Student Marks Summary</caption>
  <colgroup> <!-- 6) colgroup + 12) col -->
    <col span="1" style="background: #red; width:120px;">
    <col style="width:160px; background-color: #green;">
  </colgroup>
  <thead> <!-- 8) thead with 3) th --><tr>
    <th rowspan="2">Student</th> <!-- 11) rowspan -->
    <th colspan="3">Marks</th> <!-- 10) colspan -->
  </tr><tr>
    <th>Math</th>
    <th>Science</th>
    <th>Total</th>
  </tr>
  <thead>
  <tbody> <!-- 7) tbody with 2) tr -->
    <tr>
      <td>Sangam</td>
      <td>78</td>
      <td>82</td>
      <td>160</td>
    </tr>
  </tbody>
  <tfoot><!-- 9) tfoot -->
    <tr>
      <th>Total/Avg</th>
      <td>77.3</td>
      <td>81.3</td>
      <td>158.7</td>
    </tr>
</tfoot></table></body></html>

```

<colgroup> Attributes

Span Specifies how many columns the <colgroup> applies to.

Example: <colgroup span="2"> → applies to 2 columns.

width (deprecated in HTML5, use CSS)

Sets the width of the grouped columns.

Example: <colgroup width="150">

class / id

Assigns CSS classes or IDs to the column group for styling.

Example: <colgroup class="highlight">

Style Inline CSS to style the column group (background, text alignment, etc.).

Example: <colgroup style="background:#f5f5f5;">

Student Marks Summary

Student	Marks		
	Math	Science	Total
Sangam	78	82	160
Total/Avg	77.3	81.3	158.7

HTML Lists

HTML lists are used to display information in list format. There are 3 types:

1) Ordered List (Numbered List)

- Displays items with **numbers**.
- Each item uses tag.

Attributes:

- type → "1", "A", "a", "i", "l" (numbering style)
- start → sets **starting number**

Example:

```
<ol type="A" start="3">
  <li>Apple</li>
  <li>Banana</li>
</ol>
```

```
<ol type="A" start="3" reversed>
  <li>Apple</li>
  <li>Banana</li>
  <li>Mango</li>
</ol>
```

Output

- Starts at **C** (3rd letter)
 - Goes backward: **C, B, A**
- you can **combine start + reversed** to start at a custom value and count down

2) Unordered List (Bulleted List)

- Displays items with **bullets**.

Types:

- disc → default filled circle
- circle → empty circle
- square → square
- none → no bullet

Example:

```
<ul type="square">
  <li>Milk</li>
  <li>Bread</li>
</ul>
```



3) Description List (Definition List) <dl>

- Displays items like **dictionary definitions**.

Tags:

- <dl> → defines the list
- <dt> → data term
- <dd> → data definition

Example:

```
<dl>
  <dt>HTML</dt>
  <dd>HyperText Markup Language</dd>
  <dt>CSS</dt>
  <dd>Cascading Style Sheets</dd>
</dl>
```

Nested list: - list inside list

HTML FORM TAG

- An **HTML form** collects user input and sends it to the server.
- **<form>** tag is used.
- Important attributes:
 - **action** → where to send data.
 - **method** → how to send (GET / POST).

HTML5 Form Tags (Simple with Examples)

1. **<form>** → Creates form `<form action="demo.js" method="post"></form>`
2. **<input>** → Single-line input `<input type="text" name="username" placeholder="Enter name">`
3. **<textarea>** → Multi-line input `<textarea name="message" rows="4" cols="30"></textarea>`
4. **<select> + <option>** → Dropdown

```
<select name="fruit">
  <option>Apple</option>
  <option>Banana</option>
</select>
```
5. **<label>** → Label for input

```
<label for="email">Email:</label>
<input id="email" type="email" name="email">
```
6. **<fieldset> + <legend>** → Group fields

```
<fieldset>
  <legend>Login Info</legend>
  <input type="text" placeholder="Username">
  <input type="password" placeholder="Password">
</fieldset>
```
7. **<button>** → Clickable button → `<button type="submit">Submit</button>`
8. **<datalist>** → Suggestions for input

```
<input list="browsers">
<datalist id="browsers">
  <option value="Chrome">
  <option value="Firefox">
</datalist>
```
9. **<output>** → Show result → `<output>Result: 50</output>`
10. **<progress>** → Progress bar → `<progress value="40" max="100"></progress>`
11. **<meter>** → Measurement → `<meter min="0" max="5" value="4">4/5</meter>`

Note:

- Use **<form>** to create form.
- Inside, use inputs like **<input>**, **<textarea>**, **<select>**, **<button>**.
- Extra tags like **<fieldset>**, **<legend>**, **<datalist>**, **<progress>**, **<meter>** enhance forms.

<input> tag in HTML supports many type values for collecting different types of data.

<input> types

- **Textual input** → text, password, email, number, tel, url, search
- **Date/Time input** → date, time, datetime-local, month, week
- **Choice input** → radio, checkbox, color
- **File/Button input** → file, hidden, submit, reset, button, image

All Types of <input> in HTML

- 1) **Text input** → Single-line text
- 2) **Password input** → Hidden characters
- 3) **Email input** → Validates email format
- 4) **Number input** → Only numbers
- 5) **Tel input** → For phone numbers
- 6) **URL input** → For web addresses
- 7) **Search input** → Search box
- 8) **Date input** → Date picker
- 9) **Time input** → Time picker
- 10) **Datetime-local input** → Date + time (local)
- 11) **Month input** → Month + year
- 12) **Week input** → Week of the year

Selection Inputs

- 13). **Radio button** → Select one from many

- 14). **Checkbox** → Select multiple options

- 15). **Color picker** → Choose a color

File & Button Inputs

- 16). **File input** → Upload files

- 17). **Hidden input** → Hidden data (not visible)

- 18). **Submit button** → Submit form

- 19). **Reset button** → Reset form fields

- 20). **Button input** → Simple button

- 21). **Image button** → Submit using image

```

<input type="text" name="username" placeholder="Enter name">
<input type="password" name="pass" placeholder="password">
<input type="email" name="email" placeholder="Enter email">
<input type="number" name="age" min="1" max="100">
<input type="tel" name="phone" placeholder="123-456-7890">
<input type="url" name="website" placeholder="https://ridbharat.com">
<input type="search" name="search" placeholder="Search here">
<input type="date" name="dob">
<input type="time" name="meeting">
<input type="datetime-local" name="appointment">
<input type="month" name="birthmonth">
<input type="week" name="week">

```

```

<input type="radio" name="gender" value="male"> Male
<input type="radio" name="gender" value="female"> Female
<input type="checkbox" name="hobby" value="reading"> Reading
<input type="checkbox" name="hobby" value="sports"> Sports
<input type="color" name="favcolor">

```

```

<input type="file" name="resume">
<input type="hidden" name="userid" value="12345">
<input type="submit" value="Submit">
<input type="reset" value="Reset">
<input type="button" value="Click Me">
<input type="image" src="submit.png" alt="Submit" width="80">

```

```

<form id="myForm">
  <label for="uname">Username:</label>
  <input
    type="text"          <!-- input type -->
    name="username"      <!-- name for backend -->
    id="uname"           <!-- unique id -->
    class="input-box"     <!-- CSS class -->
    placeholder="Enter name" <!-- hint text -->
    value="Sangam Kumar" <!-- default value -->
    maxlength="20"        <!-- max characters -->
    required              <!-- must fill -->
    autofocus             <!-- cursor focus on load -->
    readonly              <!-- not editable but sent -->
    size="25"              <!-- width in characters -->
    autocomplete="on"      <!-- auto suggestions -->
    pattern="[A-Za-z]{3,}" <!-- regex validation -->
    title="Only letters, min 3" <!-- tooltip on error -->
    spellcheck="true"       <!-- enable spell check -->
    list="names"           <!-- datalist support -->
    inputmode="text"        <!-- mobile keyboard mode -->
  > <!-- datalist for suggestions -->
  <datalist id="names">
    <option value="Sangam">
    <option value="Amit">
    <option value="Ravi">
    <option value="Neha">
  </datalist>
</form>

```

Note: <input> → list="names" <datalist> → id="names" They must be same

```

<form id="myForm">
  <label for="uname">Username:</label>
  <input
    type="text"
    id="uname"
    name="username"
    list="names"  <!-- connects input with datalist -->
    placeholder="Enter name">

  <datalist id="names"> <!-- must match list="names" -->
    <option value="Sangam">
    <option value="Amit">
    <option value="Ravi">
    <option value="Neha">
  </datalist>
</form>

```

HTML <iframe> Tag

<iframe> (inline frame) is used to **embed another webpage/document** inside your HTML page.

Syntax: <iframe src="URL" width="400" height="300"></iframe>

Important Attributes:

1. **src** → URL of the page/document.
2. **srcdoc** → Directly write small HTML inside iframe.
3. **name** → Assign name, used with target.
4. **width / height** → Set size.
5. **frameborder** → (Old) border around iframe (0 or 1).
6. **allow** → Allow features like fullscreen, camera, etc.
7. **allowfullscreen** → Enable full-screen mode.
8. **sandbox** → Apply restrictions (allow-scripts, allow-forms, etc.).
9. **scrolling** → (Old) scrollbar control (yes, no, auto).
10. **title** → Accessibility title for screen readers.
11. **referrerpolicy** → Control referrer info sent.

Example:

```
<iframe src="https://www.example.com"
        width="400"
        height="300"
        name="myFrame"
        allowfullscreen
        title="Example site">
</iframe>
<a href="page2.html" target="myFrame">Open Page 2 in Iframe</a>
```

HTML File Paths File paths are used to **link images, pages, CSS, JS** etc.

Types:

1. **Absolute Path** → Full URL or system path.
2. **Relative Path** → Based on current HTML file location.
 - **Same folder:**
 - **Subfolder:**
 - **Parent folder:**
 - **From root (/):**

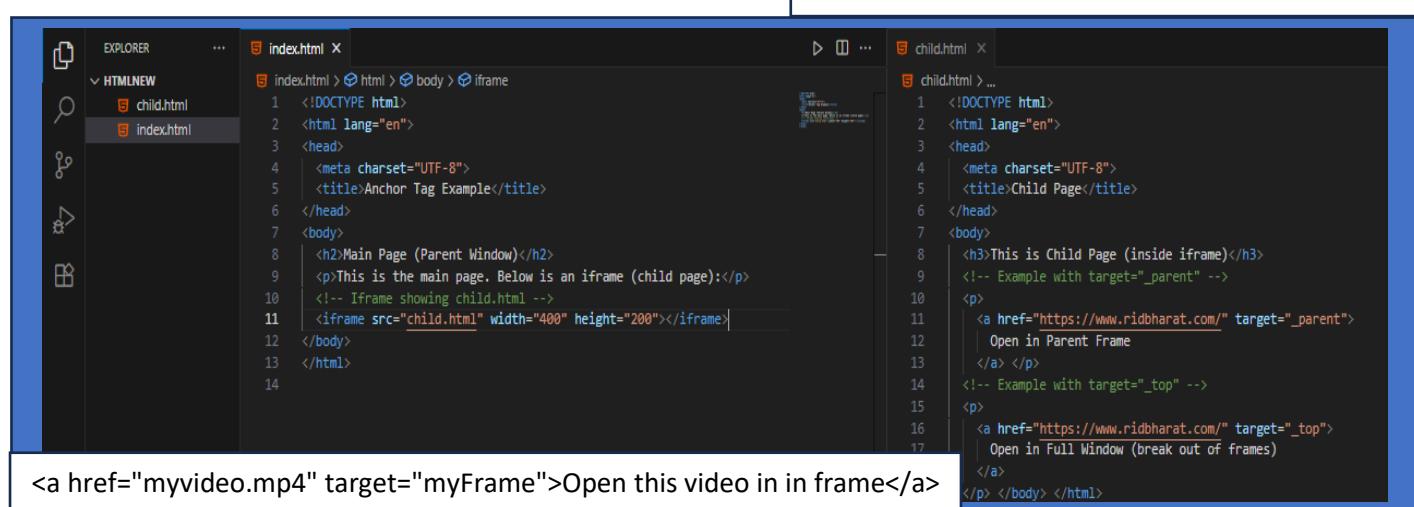
In index.html:

```

<a href="pages/about.html">About Us</a>
```

Quick Example (with folder structure):

```
mywebsite/
  |-- index.html
  |-- images/
  |   |-- logo.png
  |-- pages/
  |   |-- about.html
```



```
index.html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Anchor Tag Example</title>
</head>
<body>
<h2>Main Page (Parent Window)</h2>
<p>This is the main page. Below is an iframe (child page):</p>
<!-- Iframe showing child.html -->
<iframe src="child.html" width="400" height="200"></iframe>
</body>
</html>
```

```
child.html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Child Page</title>
</head>
<body>
<h3>This is Child Page (inside iframe)</h3>
<!-- Example with target="_parent" -->
<p>
<a href="https://www.ridbharat.com/" target="_parent">
  Open in Parent Frame
</a> </p>
<!-- Example with target="_top" -->
<p>
<a href="https://www.ridbharat.com/" target="_top">
  Open in Full Window (break out of frames)
</a>
</p>
</body>
</html>
```

Open this video in in frame

HTML Semantic & None Semantic Elements

Semantic elements have *meaning*.

They tell the browser and developer **what type of content** is inside.

- Improve **SEO** (search engine ranking)
- Improve **Accessibility**
- Make code more **readable**

Examples:

1. <article> → Independent content (blog/news).
2. <aside> → Sidebar or extra info.
3. <header> → Top part (logo, nav, heading).
4. <footer> → Bottom part (contact, copyright).
5. <main> → Main content of the page.
6. <nav> → Navigation links.
7. <section> → Grouped content (like chapters).
8. <figure> → Image/video with <figcaption>.
9. <details> + <summary> → Expand/Collapse info.
10. <time> → Date or time.
11. <mark> → Highlighted text.

Semantic Example:

```
<!DOCTYPE html>
<html>
<head>
<title>Semantic Example</title>
</head>
<body>
<header>
<h1>My Website</h1>
<nav>
<a href="#">Home</a> | <a href="#">About</a>
</nav>
</header>
<main>
<article>
<h2>Article Title</h2>
<p>This is a blog post.</p>
</article>
<aside>
<p>Related links go here.</p>
</aside>
</main>
<footer>
<p>© 2025 My Website</p>
</footer>
</body>
</html>
```

Non-semantic tags **do not describe content**.

They are mostly used for **styling/layout**.

Examples:

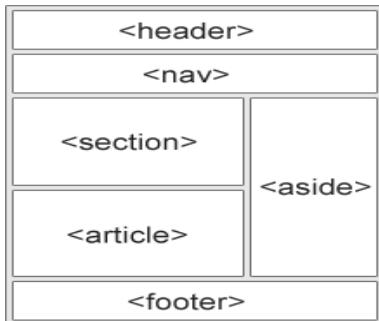
1. <div> → Block container (no meaning).
2. → Inline container (no meaning).
3.
 → Line break.
4. / <i> → Bold & Italic (style only).
5. → Old tag for fonts (deprecated).
6. <center> → Old tag for centering (deprecated).

Non-Semantic Example:

```
<!DOCTYPE html>
<html>
<head>
<title>Non-Semantic Example</title>
</head>
<body>
<div>
<h1>Welcome</h1>
<p>This is inside a div.</p>
</div>
<p>This is a <span style="color:blue;">blue</span> word.</p>
<p>This is text.<br>New line here.</p>
<p><b>Bold</b> and <i>Italic</i></p>
<center>Centered content</center>
</body>
</html>
```

HTML LAYOUT ELEMENTS

HTML layout elements are used to structure the content of a web page, defining the overall organization and arrangement of elements.



◆ HTML Entities (Short Explanation)

Definition: HTML Entities are special codes used to display reserved characters (like <, >), symbols (€, ©), or spaces in HTML.

◆ Common Entities

Result Description	Entity Name	Entity Number
non-breaking space	 	
<	<	<
>	>	>
&	&	&
"	"	"
'	'	'
¢	¢	¢
£	£	£
¥	¥	¥
€	€	€
©	©	©
®	®	®

Symbol Entity	Number	Meaning
+	+	+ Addition
-	−	− Subtraction
×	×	× Multiplication
÷	÷	÷ Division
=	=	= Equals
√	√	√ Square root
≤	≤	≤ Less or equal
≥	≥	≥ Greater or equal
≠	≠	≠ Not equal
Σ	∑	∑ Summation

Example 1: Simple Usage

```
<!DOCTYPE html>
<html>
<body>
<p>&lt;b&gt;Hello&lt;/b&gt; shows
<b>Hello</b></p>
<p>Price: 50&cent; or &pound;100 or &euro;50</p>
<p>&copy; 2025 My Company &reg;</p>
</body>
</html>
```

Symbol Entity Number Meaning

ʃ	∫	∫ Integral
∞	∞	∞ Infinity

◆ Greek Letters

Symbol Entity Number Name

α	α	α Alpha
β	β	β Beta
θ	θ	θ Theta
Σ	Σ	Σ Sigma
π	π	π Pi

◆ Set & Logic Symbols

Symbol Entity Number Meaning

∈	∈	∈ Element of
∉	∉	∉ Not an element
∪	∪	∪ Union
∩	∩	∩ Intersection
∧	∧	∧ AND
∨	∨	∨ OR
¬	¬	¬ NOT

◆ Other Useful Symbols

Symbol Entity Number Description

©	©	© Copyright
®	®	® Registered
™	™	™ Trademark
←	←	← Left arrow
↑	↑	↑ Up arrow
→	→	→ Right arrow
↓	↓	↓ Down arrow
♥	♥	♥ Heart
♦	♦	♦ Diamond
♣	♣	♣ Club
♠	♠	♠ Spade

◆ Example (Symbols + Emojis)

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>HTML Symbols & Emojis</title>
</head>
<body>
<p>5 + 3 = 8</p>
<p>x &lt; y and x &ne; z</p>
<p>&alpha; + &beta; = &gamma;</p>
<p>&copy; 2025 My Website &reg;</p>
<p>Emojis: 😊 😃 😄 😍 🌺 🌄 🌄</p>
</body>
</html>
```



◆ HTML EMOJIS

Emojis are Unicode characters (UTF-8). Add `<meta charset="UTF-8" />` in HTML to display them.

Emoji Number Meaning

😊	😀	Grinning face
😁	😁	Grinning with eyes
😂	😂	Tears of joy
😍	😍	Heart eyes
😅	😅	Smile with sweat
❤	💗	Heart
🗻	🗻	Mount Fuji
🗽	🗽	Statue of Liberty