

HTML Tags — Complete List with Examples & Explanations

This reference lists **all standard HTML elements** in practical groups. For each element you get: **What it does** → **Minimal example** → **Notes**. Obsolete elements are listed at the end (avoid using them).

How to use: Copy examples into a page with a valid skeleton. Unless shown otherwise, examples assume a normal HTML5 document.

0) Document & Metadata

`<!DOCTYPE html>` (doctype)

Purpose: Tells the browser to use modern HTML parsing. Not a tag, but must be the first line.

```
<!DOCTYPE html>
```

`<html>` (root)

Purpose: Root element of the page; carries the document language.

```
<!DOCTYPE html>
<html lang="en">
  ...
</html>
```

Notes: Use a correct `lang` (e.g., `en`, `hi`, `fr`).

`<head>`

Purpose: Metadata, links, scripts that don't render content.

```
<head>
  <meta charset="UTF-8">
  <title>My Page</title>
  <link rel="stylesheet" href="styles.css">
</head>
```

`<title>`

Purpose: Browser tab title, bookmarking.

```
<title>Welcome to My Site</title>
```

```
<base>
```

Purpose: Sets the base URL for relative links.

```
<base href="https://example.com/docs/">
```

Notes: Only one per document; affects all relative URLs below it.

```
<link>
```

Purpose: Links external resources (CSS, icons, preconnect, etc.).

```
<link rel="stylesheet" href="styles.css">  
<link rel="icon" href="/favicon.ico">
```

```
<meta>
```

Purpose: Metadata (charset, viewport, SEO, social).

```
<meta charset="UTF-8">  
<meta name="viewport" content="width=device-width, initial-scale=1">  
<meta name="description" content="Short description of this page.">
```

```
<style>
```

Purpose: Inline CSS in the document head or body.

```
<style>  
  body { font-family: system-ui, Arial, sans-serif; }  
</style>
```

```
<body>
```

Purpose: Visible content of the document.

```
<body>  
  <!-- page content -->  
</body>
```

`<script>` & `<noscript>`

Purpose: Run JavaScript; show fallback content when JS is disabled.

```
<script src="app.js" defer></script>
<noscript>Please enable JavaScript for the best experience.</noscript>
```

`<template>`

Purpose: Stores inert HTML fragments for later cloning via JS.

```
<template id="card-tpl">
  <article class="card"><h3></h3><p></p></article>
</template>
```

`<slot>`

Purpose: Placeholder for light-DOM content inside Web Components.

```
<my-card>
  <span slot="title">Hello</span>
  <p>Body content</p>
</my-card>
```

Notes: Used inside custom elements' Shadow DOM.

1) Content Sectioning

`<header>` / `<footer>`

Purpose: Intro/outro content for a page or section.

```
<header><h1>Site Title</h1></header>
<footer>© 2025 My Company</footer>
```

`<main>`

Purpose: Unique main content of the page (one per page).

```
<main>
  <h1>Article Title</h1>
```

```
<p>Primary content...</p>
</main>
```

```
<nav>
```

Purpose: Major navigation links.

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

```
<section>
```

Purpose: Thematically grouped content with a heading.

```
<section>
  <h2>Features</h2>
  <p>Why choose us...</p>
</section>
```

```
<article>
```

Purpose: Self-contained item (blog post, card, comment).

```
<article>
  <h2>News</h2>
  <p>Today we launched...</p>
</article>
```

```
<aside>
```

Purpose: Tangential content (sidebar, callouts).

```
<aside>Related links</aside>
```

```
<h1> ... <h6>
```

Purpose: Headings; outline the structure.

```
<h1>Title</h1>
<h2>Subsection</h2>
<h3>Sub-subsection</h3>
```

Notes: Use one `<h1>` per page or section root.

```
<address>
```

Purpose: Contact information for a person or org.

```
<address>
  Contact: <a href="mailto:info@example.com">info@example.com</a>
</address>
```

2) Grouping Content

```
<p>
```

Purpose: Paragraph of text.

```
<p>Hello world.</p>
```

```
<hr>
```

Purpose: Thematic break (section change).

```
<hr>
```

```
<pre>
```

Purpose: Preformatted text; preserves whitespace.

```
<pre>Line 1
  Line 2 (indented)
</pre>
```

```
<blockquote>
```

Purpose: Quoted section from another source.

```
<blockquote cite="https://example.com/article">
  The best way to predict the future is to invent it.
</blockquote>
```

Lists: ``, ``, ``

Purpose: Ordered/unordered lists.

```
<ol>
  <li>First</li>
  <li>Second</li>
</ol>
<ul>
  <li>Tea</li>
  <li>Coffee</li>
</ul>
```

Description list: `<dl>`, `<dt>`, `<dd>`

Purpose: Term/definition pairs.

```
<dl>
  <dt>HTML</dt><dd>Markup language for the web.</dd>
  <dt>CSS</dt><dd>Styling language.</dd>
</dl>
```

`<figure>` & `<figcaption>`

Purpose: Self-contained media with a caption.

```
<figure>
  
  <figcaption>Figure 1: Q1 Sales</figcaption>
</figure>
```

`<div>`

Purpose: Generic block container when no semantic tag fits.

```
<div class="card">Content</div>
```

3) Text-Level Semantics

Links: `<a>`

Purpose: Hyperlinks.

```
<a href="/pricing">See pricing</a>
```

Notes: Add `rel="noopener"` when using `target="_blank"`.

Emphasis/Importance: ``, ``

```
<p><em>Note:</em> Read the <strong>instructions</strong> carefully.</p>
```

`<small>` / `<s>`

```
<small>Fine print text</small>
<s>Old price ₹999</s>
```

Citations & Quotes: `<cite>`, `<q>`

```
<p>As <cite>Knuth</cite> wrote, <q>Premature optimization...</q></p>
```

Definitions & Abbreviations: `<dfn>`, `<abbr>`

```
<p><dfn id="http">HTTP</dfn> is the protocol...</p>
<p><abbr title="Application Programming Interface">API</abbr></p>
```

Data & Time: `<data>`, `<time>`

```
<p>Price: <data value="499">₹499</data></p>
<time datetime="2025-10-16">Oct 16, 2025</time>
```

Code & Variables: `<code>`, `<var>`, `<samp>`, `<kbd>`

```
<p>Use <code>npm install</code>.</p>
<p>Area = <var>π</var>r<sup>2</sup>. Output: <samp>OK</samp>. Press
<kbd>Ctrl</kbd>+<kbd>S</kbd>.</p>
```

Sub/Superscript: `<sub>`, `<sup>`

```
H<sub>2</sub>O, 10<sup>6</sup>
```

Styling keywords: `<i>`, ``, `<u>`, `<mark>`

```
<i>Foreign word</i>
<b>Key term</b>
<u>Annotated text</u>
<mark>Highlighted</mark>
```

Ruby Annotation: `<ruby>`, `<rb>`, `<rt>`, `<rtc>`, `<rp>`

```
<ruby>
  漢<rt>kan</rt>
  字<rt>ji</rt>
</ruby>
```

Notes: `<rp>` provides parentheses for browsers that don't support ruby.

Bidirectional Text: `<bdi>`, `<bdo>`

```
<p>User: <bdi>أحمد</bdi></p>
<p><bdo dir="rtl">abcd</bdo></p>
```

Generic inline: ``

```
<p><span class="accent">Hello</span> world</p>
```

Line breaks & soft wrap: `
`, `<wbr>`

```
Address line 1<br>Address line 2
supercalifragilistic<wbr>expialidocious
```

4) Edits

`<ins>` / ``

Purpose: Mark inserted/deleted text (diffs, policies).


```
<p><del>Old clause.</del> <ins>New clause applies.</ins></p>
```

5) Images, Media & Embedding

Images: ``

```

```

Responsive images: `<picture>` & `<source>`

```
<picture>
  <source srcset="hero.avif" type="image/avif">
  <source srcset="hero.webp" type="image/webp">
  
</picture>
```

Audio/Video: `<audio>`, `<video>`, `<source>`, `<track>`

```
<video controls width="480" poster="cover.jpg">
  <source src="movie.mp4" type="video/mp4">
  <track kind="captions" src="movie.en.vtt" srclang="en" label="English">
</video>
```

`<iframe>`

Purpose: Embed another HTML page.

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

`<embed>` / `<object>` / `<param>`

Purpose: Embed external content (PDFs, plugins). Use sparingly.

```
<embed src="file.pdf" type="application/pdf" width="100%" height="400">
<object data="file.pdf" type="application/pdf" width="100%" height="400">
  <param name="view" value="FitH">
</object>
```

Image maps: `<map>` & `<area>`

```

<map name="roommap">
  <area shape="rect" coords="34,44,270,350" href="/kitchen" alt="Kitchen">
</map>
```

Graphics: `<canvas>`

Purpose: 2D/3D drawing via JS.

```
<canvas id="c" width="300" height="150"></canvas>
<script>
  const ctx = document.getElementById('c').getContext('2d');
  ctx.fillRect(20,20,80,40);
</script>
```

Namespaced: `<svg>` / `<math>`

Purpose: Inline SVG graphics; MathML formulas.

```
<svg viewBox="0 0 100 20" width="200"><text x="0" y="15">Hello</text></svg>
<math><msup><mi>a</mi><mn>2</mn></msup></math>
```

6) Tabular Data

`<table>` family

```
<table>
  <caption>Monthly Sales</caption>
  <colgroup>
    <col span="1" style="background:#f8fafc">
    <col span="2">
  </colgroup>
  <thead>
    <tr><th>Month</th><th>Units</th><th>Revenue</th></tr>
  </thead>
  <tbody>
    <tr><th scope="row">Jan</th><td>120</td><td>₹24k</td></tr>
    <tr><th scope="row">Feb</th><td>140</td><td>₹28k</td></tr>
  </tbody>
  <tfoot>
    <tr><td colspan="3">All values are estimates.</td></tr>
```

```
</tfoot>
</table>
```

Elements covered: `<table>`, `<caption>`, `<colgroup>`, `<col>`, `<thead>`, `<tbody>`, `<tfoot>`, `<tr>`, `<th>`, `<td>`.

7) Forms

`<form>` (container)

```
<form action="/signup" method="post">
  ...controls...
</form>
```

`<label>`

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

`<input>` (many types)

```
<input type="text" name="user" minlength="3" required>
<input type="password" name="pwd" minlength="6" required>
<input type="email" name="email">
<input type="number" name="age" min="1" max="120">
<input type="date" name="dob">
<input type="color" name="fav">
<input type="checkbox" name="agree" required>
<input type="radio" name="plan" value="pro">
<input type="file" name="resume" accept="application/pdf">
```

`<textarea>`

```
<textarea name="bio" rows="4" minlength="10"></textarea>
```

`<button>`

```
<button type="submit">Submit</button>
<button type="button" onclick="alert('Hi')">Click</button>
```

`<select>`, `<optgroup>`, `<option>`

```
<label for="country">Country</label>
<select id="country" name="country">
  <optgroup label="Asia">
    <option>India</option>
    <option>Japan</option>
  </optgroup>
  <optgroup label="Europe">
    <option>Germany</option>
  </optgroup>
</select>
```

`<datalist>` (suggestions for `<input list>`)

```
<input list="langs" name="lang">
<datalist id="langs">
  <option value="HTML"><option value="CSS"><option value="JavaScript">
</datalist>
```

`<fieldset>` & `<legend>`

```
<fieldset>
  <legend>Account</legend>
  <label>Username <input name="user"></label>
</fieldset>
```

`<output>`

```
<form oninput="sum.value = +a.value + +b.value">
  <input type="number" id="a"> + <input type="number" id="b"> =
  <output name="sum" for="a b">0</output>
</form>
```

`<progress>` & `<meter>`

```
<progress max="100" value="70">70%</progress>
<meter min="0" max="100" low="30" high="80" value="65">65</meter>
```

8) Interactive Elements

`<details>` & `<summary>` (disclosure widget)

```
<details>
  <summary>Show more</summary>
  <p>Hidden content.</p>
</details>
```

`<dialog>` (modal / popup)

```
<dialog id="dlg">
  <p>Hello!</p>
  <form method="dialog"><button>Close</button></form>
</dialog>
<button onclick="document.getElementById('dlg').showModal()">Open</button>
```

`<menu>`

Purpose: Toolbar or list of commands (limited modern use).

```
<menu type="toolbar">
  <button>Save</button>
  <button>Share</button>
</menu>
```

Notes: Use sparingly; not widely needed.

9) Obsolete/Deprecated (Avoid Using)

These elements are obsolete or non-standard—**do not use** in modern HTML. They remain listed for legacy understanding.

- **Presentational/legacy:** `acronym`, `applet`, `big`, `center`, `dir`, `font`, `strike`, `tt`, `u` (as purely presentational), `basefont`.
- **Frames:** `frame`, `frameset`, `noframes`.
- **Others:** `isindex`, `keygen`, `listing`, `menuitem`, `nobr`, `plaintext`, `xmp`, `blink`, `marquee` (non-standard).

Prefer semantic elements plus CSS for presentation and JS for behavior.

10) Minimal Boilerplate (Copy-Paste Starter)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Starter</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <header><h1>Hello</h1></header>
  <main>
    <p>Content goes here.</p>
  </main>
  <footer>© 2025</footer>
  <script src="app.js" defer></script>
</body>
</html>
```

Tips

- Give every form control a `<label>`.
- Always set `alt` on `` (describe the image's purpose).
- Use headings in order, don't skip levels for styling.
- Prefer CSS for layout (`flex` , `grid`), not tables.
- Use `loading="lazy"` for off-screen images.

End of Reference