



HTML Cheat Sheet

This HTML Cheat Sheet provides a quick reference for HTML document structure, common elements, attributes, forms, semantic HTML, media, accessibility, URLs, and other frequently used HTML features.

Use this cheat sheet as a fast reminder while creating webpages or working through the HTML tutorials and reference pages.

1. Basic HTML Document

A modern HTML page begins with the HTML5 doctype, followed by the root `<html>` element. Metadata and document resources belong in the `<head>`, while visible page content belongs in the `<body>`.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Page Title</title>
  </head>
  <body>
    <h1>Page Heading</h1>
    <p>Page content goes here.</p>
  </body>
</html>
```

Element	Purpose
<code><!doctype html></code>	Declares an HTML5 document and enables standards mode.
<code><html></code>	The root element of the HTML document.
<code><head></code>	Contains metadata and document resources.
<code><meta></code>	Provides metadata such as character encoding and viewport information.
<code><title></code>	Defines the document title.

<body>	Contains the rendered content of the webpage.
--------	---

2. Text & Content

Element	Use
<h1>-<h6>	Heading levels from highest to lowest.
<p>	Defines a paragraph.
	Indicates strong importance.
	Indicates stress emphasis.
 	Creates a line break.
<hr>	Represents a thematic break.
<blockquote>	Defines a block quotation.
<q>	Defines an inline quotation.
<code>	Represents computer code.
<pre>	Displays preformatted text while preserving whitespace.
<mark>	Highlights or marks text.
<sub> / <sup>	Creates subscript and superscript text.

3. Links & Images

Contact Us

Attribute	Common Use
href	Specifies the destination of a link.
target="_blank"	Opens a link in a new browsing context.

download	Suggests downloading the linked resource.
src	Specifies an image or media resource.
alt	Provides a text alternative for an image.
width / height	Specifies image dimensions.
loading="lazy"	Allows off-screen images or iframes to be loaded later.

Accessibility: Write useful alt text for informative images. Decorative images normally use alt="".

4. Lists

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

<ol>
  <li>Create the page.</li>
  <li>Test the page.</li>
</ol>
```

Element	Purpose
	Defines an unordered list.
	Defines an ordered list.
	Defines a list item.
<dl>	Defines a description list.
<dt>	Defines a term or name.
<dd>	Provides the description or value for a term.

5. Tables

```

<table>
  <caption>Plant Schedule</caption>
  <thead>
    <tr>
      <th scope="col">Plant</th>
      <th scope="col">Water</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Tomato</td>
      <td>Regularly</td>
    </tr>
  </tbody>
</table>

```

Element	Purpose
<table>	Defines a table.
<caption>	Provides a title or caption for the table.
<thead> / <tbody> / <tfoot>	Groups table rows by purpose.
<tr>	Defines a table row.
<th>	Defines a header cell.
<td>	Defines a data cell.
colspan / rowspan	Makes a cell span multiple columns or rows.
scope	Associates a header cell with a row, column, row group, or column group.

6. Forms

```

<form action="process.php" method="post">
  <label for="name">Name:</label>
  <input type="text" id="name" name="name" required>

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

```

Element / Attribute	Purpose
<form>	Contains form controls.
action	Specifies the URL that receives the submitted form data.
method	Specifies the submission method, commonly GET or POST.
<label>	Provides an accessible label for a form control.
<input>	Creates an input control.
<textarea>	Creates a multi-line text input.
<select> / <option>	Creates a selection list and its choices.
<button>	Creates a button control.
name	Provides the name used when a control's value is submitted.
required	Requires a value before the form can be submitted.
autocomplete	Provides browser autofill guidance.

Important: Connect a <label> to its form control with matching for and id values whenever the control can be explicitly labeled.

7. Semantic Page Structure

```
<header>...</header>
<nav>...</nav>
<main>
  <article>
    <section>...</section>
  </article>

  <aside>...</aside>
</main>
<footer>...</footer>
```

Element	Typical Purpose

<header>	Introductory content for a page or section.
<nav>	Major navigation links.
<main>	The dominant content of the document.
<article>	A self-contained composition.
<section>	A thematic grouping of content.
<aside>	Content indirectly related to the surrounding content.
<footer>	Footer information for a page or section.
<search>	Contains search or filtering functionality.

8. Audio, Video & Embedded Content

```
<video controls>
  <source src="video/example.mp4" type="video/mp4">
</video>

<audio controls>
  <source src="audio/example.mp3" type="audio/mpeg">
</audio>

<iframe src="/example.html" title="Example webpage"></iframe>
```

Element	Purpose
<audio>	Embeds audio.
<video>	Embeds video.
<source>	Provides an alternate media or image source.
<track>	Provides timed text such as captions or subtitles.
<iframe>	Embeds another HTML page or resource.
<picture>	Provides responsive or art-directed image sources.

9. Common Global Attributes

Attribute	Purpose
id	Provides a unique identifier for an element.
class	Provides one or more class names commonly used by CSS and scripts.
style	Provides inline CSS declarations.
title	Provides advisory information about an element.
lang	Specifies the language of an element and its contents.
dir	Specifies text direction.
hidden	Indicates content is not currently relevant.
tabindex	Controls keyboard focus participation and ordering.
data-*	Stores custom data for scripts and applications.
contenteditable	Makes element content editable when enabled.
inert	Makes an element subtree non-interactive.

10. URLs & Paths

Example	Meaning
page.html	A file in the current directory.
images/photo.jpg	A file in a child directory.
../index.html	Moves up one directory.
/index.html	A root-relative URL.
#section	A fragment link to an element with <code>id="section"</code> .
https://example.com/	An absolute URL.

11. Accessibility Essentials

Practice	Quick Check
Document Language	Use <code><html lang="en"></code> or another appropriate language tag.
Headings	Use meaningful heading levels to organize content.
Images	Provide appropriate alternative text.
Links	Use link text that describes the destination or purpose.
Forms	Provide programmatic labels and clear instructions.
Keyboard	Ensure interactive controls can be reached and operated with a keyboard.
Tables	Use appropriate table headers for data tables.
Color	Do not rely on color alone to communicate important information.
Native HTML	Prefer native semantic HTML before adding ARIA.
Media	Provide captions or other alternatives when required for equivalent access.

ARIA Rule of Thumb: Use native HTML elements and semantics whenever they already provide the behavior and accessibility information you need.

12. Useful <head> Markup

```
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Descriptive Page Title</title>
<meta name="description" content="A concise description of the page.">
<link rel="canonical" href="https://www.example.com/page.html">
<link rel="stylesheet" href="/css/style.css">
```

13. Common HTML Mistakes

Avoid	Better Practice
Duplicate <code>id</code> values	Keep each <code>id</code> unique within the document.
Using headings only for visual size	Choose headings for document structure and control their appearance with CSS.
Using <code>
</code> for page spacing	Use CSS margins and padding for layout spacing.
Missing form labels	Associate form controls with meaningful labels.
Missing image <code>alt</code> attributes	Provide an appropriate <code>alt</code> value, including an empty value for decorative images.
Obsolete presentational HTML	Use modern HTML semantics and CSS.
Invalid element nesting	Place elements only where the HTML content model permits them.

14. Before Publishing

Check	Why
Validate HTML	Find markup errors and conformance problems.
Test Chrome, Edge & Firefox	Catch browser rendering or behavior differences.
Test Keyboard Navigation	Find inaccessible interactions and focus problems.
Check Links & Paths	Prevent broken navigation and missing resources.
Check Responsive Layouts	Make sure content works at different viewport widths.
Check Forms	Confirm validation, submission, labels, and returned results.
Check Images & Media	Confirm resources load and appropriate alternatives are provided.
Check HTTP Responses	Confirm redirects, missing pages, and server errors return appropriate status codes.

15. Quick Element Index

Frequently used modern HTML elements grouped by purpose:

Category	Elements
Document	html, head, body, title, meta, link, style, base
Structure	header, nav, main, article, section, aside, footer, search
Text	h1-h6, p, div, span, strong, em, small, mark, br, hr
Quotations / Code	blockquote, q, cite, code, pre, kbd, samp, var
Links / Images	a, img, picture, source, map, area, figure, figcaption
Lists	ul, ol, li, dl, dt, dd, menu
Tables	table, caption, colgroup, col, thead, tbody, tfoot, tr, th, td
Forms	form, label, input, textarea, select, option, optgroup, button, fieldset, legend, datalist, output, progress, meter
Media	audio, video, source, track
Embedded	iframe, embed, object, canvas, svg, math
Interactive	details, summary, dialog
Scripting / Templates	script, noscript, template, slot

Keep Learning

This cheat sheet is designed as a quick reminder, not a replacement for the complete tutorials and HTML element and attribute references. For detailed explanations, examples, browser considerations, and accessibility guidance, use the HTML Tutorials and HTML References at [kdwebhostingdesign.com](https://www.kdwebhostingdesign.com).

HTML is maintained as a living web standard, so features and browser support can change over time. Check current documentation when working with newer or less familiar features.