

HTML Elements and Forms Cheat Sheet

Document structure, semantics, links, tables, and form controls

From **HTML, CSS & JavaScript the TPRM Way** by John I. Jerkovic

PRACTICAL TECH MADE SIMPLE

A MINIMAL DOCUMENT

<code><!doctype html></code>	The first line of every page. Nothing may come before it.
<code><html lang="en"></code>	The root element. lang names the language of the content.
<code><head></code>	Metadata for the browser. Nothing here is drawn on the page.
<code><meta charset="utf-8"></code>	Declares UTF-8 text, so accented letters appear as typed.
<code><meta name="viewport" ⚡ content="..."></code>	Value: width=device-width, initial-scale=1. Needed on phones.
<code><title>Page title</title></code>	The text in the browser tab and in search results.
<code><link rel="stylesheet" ⚡ href="styles.css"></code>	Loads the stylesheet. The path is relative to this file.
<code><script src="app.js" ⚡ defer></script></code>	Loads the script. defer runs it after the page is built.
<code><body></code>	Everything the reader sees goes here.

Three files sit side by side in one folder: `index.html`, `styles.css`, and `app.js`.

TEXT CONTENT

<code><h1>Page title</h1></code>	The top of the heading outline. Normally one per page.
<code><h2>Section</h2> ... <h6></code>	Lower levels for nested sections. Do not skip levels.
<code><p>A block of text.</p></code>	A paragraph.
<code>save your work</code>	Strong importance. Bold is only the default appearance.
<code>really</code>	Stress emphasis. Italic is only the default appearance.
<code>label and <i>term</i></code>	Visual difference with no added meaning. Use them rarely.
<code><blockquote><p>Quoted.</p></blockq ⚡ uote></code>	A quotation taken from another source.
<code>
 and <hr></code>	A line break inside text; a thematic break between sections.
<code>inline</code>	A neutral inline wrapper with no meaning of its own.
<code><div>block</div></code>	A neutral block wrapper. Correct when the grouping has no theme.

Headings describe hierarchy. Use CSS, not a heading level, to make text large.

SEMANTIC LANDMARKS

<code><header></code>	Introductory content for a page or a section.
<code><nav aria-label="Primary"></code>	Major navigation links. The label names this one.
<code><main></code>	The primary unique content. Normally one per document.
<code><section></code>	A thematic grouping, usually carrying a heading.
<code><article></code>	A self-contained item that could stand on its own.
<code><aside></code>	Related but secondary content.
<code><footer></code>	Closing information for a page or a section.
<code><figure> <figcaption></code>	An illustration with its caption kept together.

Use a landmark when the content genuinely has that role. Otherwise `div` is still correct.

LINKS AND PATHS

<code>Projects</ ⚡ a></code>	A file next to this one.
<code>Notes</ ⚡ a></code>	Go down into the pages folder.
<code>Back</code>	Two dots mean go up one folder.
<code>Jump to gear</code>	A fragment. Scrolls to the element with <code>id="gear"</code> .
<code>Ref ⚡ </code>	An absolute URL. Works from anywhere.
<code>Ma ⚡ il</code>	Opens the reader's mail program.
<code>id="gear"</code>	The target a fragment link scrolls to. Unique in the document.

Link text should name the destination. “Read the installation guide” beats “click here”.

IMAGES

<code></code>	<code>src</code> is where the file is; <code>alt</code> is what the picture means.
<code></code>	An empty <code>alt</code> marks a decorative image, so it is skipped.
<code>width="800" height="600"</code>	The natural pixel size. Reserves space while the file loads.
<code></code>	A linked image: the <code>alt</code> text describes where the link goes.

Stay away from images of text. Real text is selectable, searchable, and adaptable.

LISTS

<code>Item</code>	Unordered. Use it when the order carries no meaning.
<code>Step</code>	Ordered. Use it when sequence or rank matters.
<code>BuildHTML</code> ↳ <code>i></code>	A nested list belongs inside the li it belongs to.
<code><dl><dt>Sun</dt><dd>Six hours.</dd></dl></code>	A description list: each term paired with its description.

A navigation bar is a list of links inside nav. The list survives when CSS does not.

TABLES

<code><table></code>	For genuinely tabular data, as opposed to page layout.
<code><caption>Weekly practice</caption></code> ↳ <code>time</caption></code>	Names the table. It comes first, inside table.
<code><thead> <tbody> <tfoot></code>	The heading rows, the body rows, and any summary rows.
<code><tr></code>	One row.
<code><th scope="col">Day</th></code>	A header cell for the column below it.
<code><th scope="row">Monday</th></code>	A header cell for the row beside it.
<code><td>25</td></code>	An ordinary data cell.
<code>colspan="2" rowspan="2"</code>	One cell spanning two columns, or two rows.

Real header cells and scope are what let a screen reader read a value with its labels.

FORMS: STRUCTURE

<code><form></code>	Groups the controls that are submitted together.
<code><label for="email">Email</label></code>	The for value must equal the control's id.
<code><input id="email" name="email"></code>	id identifies it in the document; name labels the submitted data.
<code><label><input type="checkbox"> I agree</label></code> ↳ <code>agree</label></code>	A wrapping label. The usual pattern for a checkbox.
<code><fieldset><legend>Contact</legend></code>	Groups related controls under a visible caption.
<code><button type="submit">Send</button></code> ↳ <code>></code>	Submits the form. A real button, keyboard included.
<code><button type="button">Toggle</button></code> ↳ <code>on></code>	Does nothing on its own. The type for a scripted action.

Placeholder text is not a label. It disappears the moment the user types.

FORM CONTROLS

<code><input type="text"></code>	A short single-line value.
<code><input type="email"></code>	An email address. The browser checks the shape.
<code><input type="number"></code>	A number. Pairs with min, max, and step.
<code><input type="date"></code>	A calendar control. Its value reads 2026-05-04.
<code><input type="password"></code>	Text hidden as it is typed.
<code><input type="radio" name="contact"></code> ↳ <code>name="contact"></code>	One choice among several. One name for the whole group.
<code><input type="checkbox" name="agree"></code> ↳ <code>name="agree"></code>	An independent yes or no.
<code><textarea id="message" rows="5"></code>	Long text. It has a closing tag and no value attribute.
<code><select id="topic"><option value=""></code> ↳ <code>value=""></code>	A menu. An empty first value forces a real choice.

Choose the control by the question you are asking. The right type improves phone keyboards too.

VALIDATION ATTRIBUTES

<code>required</code>	The form will not submit while this control is empty.
<code>min="1" max="120"</code>	The numeric or date range the browser accepts.
<code>minlength="2" maxlength="40"</code>	The number of characters allowed.
<code>step="0.01"</code>	The allowed increment for a number control.
<code>pattern="[0-9]{5}"</code>	A regular expression the value must match.
<code>readonly</code> and <code>disabled</code>	Readable but not editable; disabled values are not submitted.
<code>autocomplete="email"</code>	Tells the browser what this field is for.

Browser validation improves the experience. It is never a security boundary.

ATTRIBUTES AND COMMENTS

<code>class="card featured"</code>	One or more styling hooks, separated by spaces.
<code>id="count-output"</code>	A unique identifier. The usual JavaScript target.
<code>aria-expanded="false"</code>	States whether the control's region is open. Keep it honest.
<code>aria-label="Primary"</code>	An accessible name where no visible label exists.
<code>hidden</code>	Removes the element from the page and from assistive technology.
<code><!-- a note --></code>	An HTML comment. The browser ignores it.
<code>&amp; &lt; &gt; &copy;</code>	Character references for &, <, >, and the copyright sign.

An attribute value is always text. Even a number arrives in JavaScript as a string.