

The Encyclopedia of the HTML Living Standard: A Comprehensive Technical Reference of Elements and Attributes (2026 Edition)

1. Introduction to the Living Standard Architecture

The architecture of the World Wide Web rests fundamentally upon the Hypertext Markup Language (HTML). Unlike its predecessors, which were defined by static version numbers such as HTML 4.01 or XHTML 1.1, the modern web is governed by the **HTML Living Standard**. Maintained by the Web Hypertext Application Technology Working Group (WHATWG), this standard represents a continuously evolving specification that adapts to the needs of modern web applications, browser capabilities, and user accessibility requirements.¹

For the professional software architect and front-end engineer, a superficial knowledge of tags is insufficient. To construct robust, accessible, and performant interfaces, one must understand the comprehensive inventory of elements, their Document Object Model (DOM) interfaces, their content models, and their semantic implications. This report provides an exhaustive analysis of the approximately 145 valid HTML elements currently recognized by major user agents.² It dissects their attributes, usage patterns, and the complex relationships that bind them into a coherent document structure.

The analysis is categorized by the official functional groupings defined in the specification: the root, metadata, sectioning, text content, inline semantics, multimedia, embedding, scripting, table data, forms, and interactive elements. Furthermore, it examines the "Global Attributes" that provide a unified interface for styling and interaction across the entire spectrum of elements.

1.1 The Quantifiable Scope of HTML

While the exact count of "valid" tags fluctuates as browsers implement experimental features (such as `<fencedframe>` or `<portal>`) and deprecate obsolete ones (like `<marquee>`), the current consensus among browser vendors establishes a core set of approximately **142 to 145 standard elements**.² This count excludes the Scalable Vector Graphics (SVG) and MathML namespaces, which, while embeddable in HTML, constitute separate specifications. Understanding this totality allows developers to select the most semantically appropriate element for any given data structure, a practice that directly influences Search Engine Optimization (SEO), accessibility (A11y), and code maintainability.³

2. Global Attributes and The DOM Interface

Before examining specific elements, it is essential to analyze the **Global Attributes**. These are attributes that can be specified on any HTML element, even those not defined in the standard. They provide the fundamental hooks for the Document Object Model (DOM), CSS styling, and accessibility trees.⁴

The global attributes represent the baseline functionality of the HTML`Element` interface, from which all specific element interfaces (e.g., `HTMLDivElement`, `HTMLAnchorElement`) inherit.

2.1 Core Identity and Styling Attributes

The most frequently utilized global attributes are those that establish identity and classification for styling and scripting.

Attribute	Description & Technical Context	Best Practice Usage
class	A space-separated list of case-sensitive classes. It maps to the <code>classList</code> property in the DOM.	Used primarily for CSS styling and targeting groups of elements in JavaScript. Avoid semantic-less names (e.g., <code>.red-text</code>); prefer utility or BEM-style naming (e.g., <code>.alert--error</code>).
id	A unique identifier for the element. It must be unique within the entire document tree. Maps to the <code>id</code> property.	Used for fragment identifiers (anchor links like <code>#section</code>), specific CSS targeting, and high-performance JS selection via <code>getElementById</code> .
style	Contains CSS styling declarations to be applied directly to the element.	Usage Warning: Inline styles generally violate Content Security Policy (CSP) best practices and hinder maintainability. Use only for dynamic, calculated values (e.g., coordinates).
title	Represents advisory information. Browsers typically render this as a tooltip on hover.	Accessibility Warning: title text is often inaccessible to touch users and inconsistently read by screen readers. Do not rely on it for critical information.

2.2 Interaction and Editing Attributes

Modern HTML applications often function as rich text editors or complex interfaces, requiring attributes that control user input and focus.

- **contenteditable**: An enumerated attribute indicating if the element should be editable by the user. If true, the browser modifies the widget to allow rich text editing.
 - *Values*: true, false, plaintext-only.
 - *Insight*: This is the foundation of web-based WYSIWYG editors. The plaintext-only value is critical for building code editors where rich formatting is undesirable.⁵
- **tabindex**: Controls whether an element is focusable and its participation in sequential keyboard navigation.
 - *Values*:
 - 0: Inserts the element into the natural tab order.
 - -1: Makes the element programmatically focusable (via `.focus()`) but removes it from the tab sequence.
 - *Positive integers*: Define a specific order (Anti-pattern: avoids messing with the natural flow).
- **accesskey**: Provides a hint for generating a keyboard shortcut for the current element.
 - *Insight*: Developers rarely use this due to conflicts with browser and screen reader shortcuts.
- **draggable**: An enumerated attribute indicating whether the element can be dragged, using the Drag and Drop API.
- **spellcheck**: Defines whether the element may be checked for spelling errors. Critical for `<textarea>` and `contenteditable` elements.

2.3 Semantic and Accessibility Attributes

- **lang**: Helps define the language of an element: the language that non-editable elements are written in, or the language that the editable elements should be written in by the user.
 - *Impact*: Crucial for screen readers to select the correct pronunciation engine (e.g., differentiating English "Chat" from French "Chat").
- **dir**: Indicates the directionality of the element's text.
 - *Values*: ltr (Left-to-Right), rtl (Right-to-Left), auto (Let the browser decide based on content).
- **translate**: Specifies whether an element's attribute values and text content are to be translated when the page is localized.
 - *Values*: yes, no.
 - *Usage*: Use no for brand names, technical code samples, or identifiers that must remain literal.

2.4 Modern Application Attributes

As HTML evolves to support complex applications, new global attributes have been introduced to handle data, security, and componentization.

- **data-***: A class of attributes, called custom data attributes, that allow proprietary

information to be exchanged between the HTML and its DOM representation that may be used by scripts.

- **Access:** These are available via the `HTMLElement.dataset` property (e.g., `data-user-id` becomes `dataset.userId`).
- **hidden:** A Boolean attribute indicating that the element is not yet, or is no longer, relevant.
 - **Semantics:** Browser default styles apply `display: none`. Unlike CSS hiding, this semantic signal tells screen readers to ignore the content entirely.
- **inert:** A Boolean attribute that indicates to the browser that the element is to be ignored by the user.
 - **Impact:** The user agent acts as if the element is absent for the purposes of hit testing and focus. This is vital for disabling the background page when a modal dialog is open.⁵
- **slot:** Used in Shadow DOM usage. It assigns a slot in a shadow DOM shadow tree to an element: an element with a slot attribute is assigned to the slot created by the `<slot>` element whose name attribute's value matches that slot attribute's value.⁶
- **part:** Contains a space-separated list of the part names of the element. Part names allow CSS to select and style specific elements in a shadow tree via the `::part` pseudo-element.⁷

3. The Main Root

The root of an HTML document is the foundation upon which the DOM tree is constructed.

<html>

- **DOM Interface:** `HTMLHtmlElement`
- **Category:** Main root.
- **Description:** The `<html>` element represents the root (top-level element) of an HTML document, so it is also referred to as the root element. All other elements must be descendants of this element.⁸
- **Attributes:** Accepts all global attributes. The `xmlns` attribute is required for XHTML serialization but optional for HTML5.
- **Usage Context:** It must be the first element in the document after the `<!DOCTYPE html>` declaration.
- **Critical Attribute - lang:** It is a strict requirement for accessibility compliance (WCAG) to declare the primary language of the document on this tag.

HTML

```
<!DOCTYPE html>
```

```
<html lang="en-US" dir="ltr">
```

```
</html>
```

- **Insight:** While browsers can render a page without this tag (via tag omission rules),

omitting it makes it impossible to define the document language effectively, hurting both SEO and accessibility.

4. Document Metadata

Metadata elements, primarily found within the `<head>`, do not display content directly to the user but define how the document is processed, rendered, and indexed.

`<head>`

- **DOM Interface:** `HTMLHeadElement`
- **Category:** Document metadata.
- **Description:** The `<head>` element contains machine-readable information (metadata) about the document, like its title, scripts, and style sheets.⁸
- **Context:** It is the first child of the `<html>` element.
- **Insight:** The order of elements in `<head>` matters. Character encoding (`<meta charset>`) should be the first child to ensure the browser parses the rest of the document using the correct encoding immediately.

`<title>`

- **DOM Interface:** `HTMLTitleElement`
- **Category:** Document metadata.
- **Description:** Defines the document's title that is shown in a browser's title bar or a page's tab. It only contains text; tags within the element are ignored.⁹
- **Importance:** This is arguably the single most important tag for SEO and user experience. It provides the blue link text in search engine results and the label for browser bookmarks.³

`<base>`

- **DOM Interface:** `HTMLBaseElement`
- **Category:** Document metadata.
- **Description:** Specifies the base URL to use for all relative URLs in a document.⁸
- **Attributes:**
 - `href`: The base URL to be used throughout the document for relative links.
 - `target`: A default target (e.g., `_blank`) for all hyperlinks in the document.
- **Usage Case:** Useful in development environments or when serving content from a CDN where the asset path differs from the document path.
- **Warning:** Improper use can break internal page anchors (`#section`), as they will resolve against the base URL rather than the current document URL.

`<link>`

- **DOM Interface:** `HTMLLinkElement`

- **Category:** Document metadata.
- **Description:** Specifies relationships between the current document and an external resource. This element is most commonly used to link to stylesheets, but is also used to establish site icons, preloading directives, and canonical URLs.⁸
- **Attributes:**
 - rel: The relationship (e.g., stylesheet, icon, canonical, preload, dns-prefetch).
 - href: The URL of the resource.
 - media: Specifies the media the resource applies to (e.g., print, (min-width: 800px)).
 - crossorigin: Handling of CORS requests for the resource.
 - as: Used with rel="preload" to tell the browser what type of content is being loaded (e.g., font, image, script).¹⁰
- **Example:**
HTML

```
<link rel="stylesheet" href="style.css">
<link rel="icon" href="favicon.ico" sizes="any">
<link rel="canonical" href="https://example.com/page">
<link rel="preload" href="font.woff2" as="font" type="font/woff2" crossorigin>
```

<meta>

- **DOM Interface:** HTMLMetaElement
- **Category:** Document metadata.
- **Description:** Represents metadata that cannot be represented by other HTML meta-related elements.
- **Attributes:**
 - name: The metadata name (e.g., viewport, description, author).
 - content: The value of the metadata.
 - charset: Declares the document's character encoding (standard is UTF-8).
 - http-equiv: Simulates an HTTP response header (e.g., default-style, refresh, Content-Security-Policy).¹⁰
- **Insight - The Viewport:** The viewport meta tag is non-negotiable for responsive web design.
HTML

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

<style>

- **DOM Interface:** HTMLStyleElement
- **Category:** Document metadata.
- **Description:** Contains style information for a document or part of a document.
- **Use Case:** Critical CSS (styles needed for the first paint) is often inlined here to improve

loading performance, while the rest is loaded via `<link>`.

5. Sectioning Root and Content Sectioning

These elements define the semantic structure of the document, creating a skeletal outline that assistive technologies navigate.

5.1 The Body

`<body>`

- **DOM Interface:** `HTMLBodyElement`
- **Category:** Sectioning root.
- **Description:** Represents the content of an HTML document. There can be only one `<body>` element in a document.⁸
- **Attributes:** Accepts global attributes. Legacy attributes like `background` or `bgcolor` are obsolete and must be avoided in favor of CSS.¹⁰
- **Event Handlers:** Often used for document-wide events like `onload`, `ononline`, `onhashchange`.

5.2 Content Sectioning Elements

These elements allow you to organize the document content into logical pieces. They are vital for creating a "semantic" outline.

`<article>`

- **DOM Interface:** `HTMLArticleElement`
- **Category:** Content sectioning.
- **Description:** Represents a self-contained composition in a document, page, application, or site, which is intended to be independently distributable or reusable.¹¹
- **Use Case:** Forum posts, magazine articles, blog entries, user comments, or interactive widgets.
- **Heuristic:** If the content would make sense in an RSS feed or syndicated on another site, use `<article>`.

`<section>`

- **DOM Interface:** `HTMLSectionElement`
- **Category:** Content sectioning.
- **Description:** Represents a generic standalone section of a document, which doesn't have a more specific semantic element to represent it.¹²
- **Usage:** A section should typically have a heading. If it is purely for styling purposes, use a `<div>` instead.

<nav>

- **DOM Interface:** HTMLElement
- **Category:** Content sectioning.
- **Description:** Represents a section of a page whose purpose is to provide navigation links, either within the current document or to other documents.¹³
- **Accessibility:** Screen readers provide a shortcut to jump directly to <nav> regions.
- **Example:**

HTML

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

<aside>

- **DOM Interface:** HTMLElement
- **Category:** Content sectioning.
- **Description:** Represents a portion of a document whose content is only indirectly related to the document's main content.¹²
- **Use Case:** Sidebars, call-out boxes, advertising slots, or groups of nav elements separate from the main content.

<h1> through <h6>

- **DOM Interface:** HTMLHeadingElement
- **Category:** Content sectioning.
- **Description:** Represent six levels of section headings. <h1> is the highest section level and <h6> is the lowest.¹⁴
- **Semantic Hierarchy:** Users of assistive technology often skip from heading to heading to determine the content of a page. It is critical not to skip heading levels (e.g., jumping from <h1> to <h3>) as this breaks the logical outline of the document.

<hgroup>

- **DOM Interface:** HTMLElement
- **Category:** Content sectioning.
- **Description:** Represents a heading and related content. It allows grouping a heading with a tagline or subtitle without affecting the document outline depth.
- **Status:** This element was removed from the spec for a time but has been reinstated in the Living Standard.

<header>

- **DOM Interface:** HTMLInputElement
- **Category:** Content sectioning.
- **Description:** Represents introductory content, typically a group of introductory or navigational aids.¹³ It may contain some heading elements but also a logo, a search form, an author name, and other elements.
- **Context:** A <header> can be used for the whole page (inside <body>) or for a specific <article>.

<footer>

- **DOM Interface:** HTMLInputElement
- **Category:** Content sectioning.
- **Description:** Represents a footer for its nearest sectioning content or sectioning root element.¹³ A footer typically contains information about the author of the section, copyright data, or links to related documents.

<address>

- **DOM Interface:** HTMLInputElement
- **Category:** Content sectioning.
- **Description:** Indicates that the enclosed HTML provides contact information for a person or people, or for an organization.¹³
- **Nuance:** It should not be used for arbitrary physical addresses (like a postal address in a story), but specifically for the *contact* information of the document owner.

<main>

- **DOM Interface:** HTMLInputElement
- **Category:** Content sectioning.
- **Description:** Represents the dominant content of the <body> of a document. The main content area consists of content that is directly related to or expands upon the central topic of a document.
- **Constraint:** There must be only one visible <main> element in a document. It should not contain content that is repeated across a set of documents (like sidebars, navigation links, copyright information).

6. Text Content

These elements are used to organize blocks of text within the sections defined above.

<p> (Paragraph)

- **Description:** Represents a paragraph. Paragraphs are block-level elements, and

browsers automatically add some white space (a margin) before and after a paragraph.

<hr> (Horizontal Rule)

- **Description:** Represents a thematic break between paragraph-level elements.¹⁰
- **Semantic:** While visually a line, semantically it represents a shift in topic (e.g., a scene change in a novel or a transition between topics in a tutorial).

<pre> (Preformatted Text)

- **Description:** Represents preformatted text which is to be presented exactly as written in the HTML file.¹⁵
- **Behavior:** Text within this element is typically displayed in a non-proportional ("monospace") font, and whitespace (spaces and line breaks) is preserved.
- **Use Case:** Displaying code blocks, poetry, or ASCII art.

<blockquote>

- **Description:** Indicates that the enclosed text is an extended quotation.
- **Attributes:**
 - **cite:** A URL that designates a source document or message for the information quoted.

** (Ordered List)**

- **Description:** Represents an ordered list of items — typically rendered as a numbered list.
- **Attributes:**
 - **reversed:** Numbering is descending.
 - **start:** Specifies the start value of the list.
 - **type:** Kind of marker (1, a, A, i, I).¹⁰

** (Unordered List)**

- **Description:** Represents a list of items where the order does not strictly matter. Typically rendered as a bulleted list.

<menu>

- **Description:** A semantic alternative to , representing a group of commands that a user can perform or activate. Used often for toolbars or context menus.¹⁶

** (List Item)**

- **Description:** Represents an item in a list. It must be contained in a parent element: an ordered list (), an unordered list (), or a menu (<menu>).
- **Attributes:**
 - **value:** Indicates the current ordinal value of the list item (only valid in).

<dl> (Description List)

- **Description:** Represents a description list. The element encloses a list of groups of terms (specified using the <dt> element) and descriptions (provided by <dd> elements). Common uses include glossaries or metadata (key-value pairs).

<dt> (Description Term)

- **Description:** Specifies a term in a description or definition list.

<dd> (Description Details)

- **Description:** Provides the description, definition, or value for the preceding term (<dt>).¹³

<figure>

- **Description:** Represents self-contained content, potentially with an optional caption, which is specified using the <figcaption> element.
- **Semantic:** The content acts as a single unit and is referenced from the main content of the document. Moving it to an appendix would not affect the flow of the document.

<figcaption>

- **Description:** Represents a caption or legend for the rest of the contents of its parent <figure> element.¹³

<div> (Content Division)

- **Description:** The generic container for flow content. It has no effect on the content or layout until styled using CSS.
- **Best Practice:** As a purely structural element, it should be used only when no other semantic element (like <article> or <nav>) is appropriate.

7. Inline Text Semantics

These elements define the meaning, structure, or style of a specific word or line of text.

<a> (Anchor)

- **Description:** Creates a hyperlink to web pages, files, email addresses, locations in the same page, or anything else a URL can address.
- **Attributes:**
 - href: The URL.
 - target: Context for opening the link (_self, _blank, _parent, _top).
 - download: Prompts the user to save the linked URL instead of navigating to it.
 - rel: Relationship (e.g., noopener, noreferrer, nofollow).¹⁰

- **Security Insight:** When using `target="_blank"`, always include `rel="noopener"` to prevent the new page from accessing the `window.opener` object of the original page, which is a security vulnerability.

** (Emphasis)**

- **Description:** Marks text that has stress emphasis.
- **Semantic:** Changes the meaning of the sentence (e.g., "I *did* do it" vs "I did *do* it").

** (Strong Importance)**

- **Description:** Indicates that its contents have strong importance, seriousness, or urgency.
- **Visual:** Typically rendered in bold.

<small>

- **Description:** Represents side-comments and small print, like copyright and legal text, independent of its styled presentation.

<s> (Strikethrough)

- **Description:** Renders text with a strikethrough, or a line through it. Use the `<s>` element to represent things that are no longer relevant or no longer accurate.

<cite>

- **Description:** Used to describe a reference to a cited creative work, and must include the title of that work.

<q> (Inline Quotation)

- **Description:** Indicates that the enclosed text is a short inline quotation. Most browsers automatically add quotation marks.

<dfn> (Definition)

- **Description:** Represents the defining instance of a term.

<abbr> (Abbreviation)

- **Description:** Represents an abbreviation or acronym.
- **Attributes:** `title` (contains the full description).
- **Example:** `<abbr title="Hypertext Markup Language">HTML</abbr>`

<ruby>, <rt>, <rp>

- **Description:** Used for East Asian typography (Ruby annotations) to show pronunciation above characters.
- **Components:** `<ruby>` is the container, `<rt>` provides the annotation (e.g., Furigana), and `<rp>` provides fallback parentheses for browsers that do not support Ruby.⁹

<data>

- **Description:** Links a given content with a machine-readable translation.
- **Attributes:** value (the machine-readable value).

<time>

- **Description:** Represents a specific period in time.
- **Attributes:** datetime (machine-readable format, e.g., "2026-05-14").
- **Use Case:** Essential for calendar events and search engines parsing event data.

<code>

- **Description:** Displays its contents styled in a fashion intended to indicate that the text is a short fragment of computer code.

<var>

- **Description:** Represents the name of a variable in a mathematical expression or a programming context.

<samp>

- **Description:** Represents sample (or quoted) output from a computer program.

<kbd> (Keyboard Input)

- **Description:** Represents a span of inline text denoting textual user input from a keyboard, voice input, or any other text entry device.

<sub> and <sup>

- **Description:** Subscript and Superscript. Used for typographical conventions like chemical formulas (H_2O) or footnotes.

<i> (Idiomatic Text)

- **Description:** Represents a range of text that is set off from the normal text for some reason, such as idiomatic text, technical terms, taxonomic designations, among others.
- **Usage:** Use this when , , <cite>, or <dfn> do not apply.

** (Bring Attention)**

- **Description:** Brings attention to the element's contents, which are not otherwise granted special importance.
- **Usage:** Keywords in a summary, product names in a review, or other spans of text whose typical typographical presentation is bold.

<u> (Unarticulated Annotation)

- **Description:** Represents a span of inline text which should be rendered in a way that indicates that it has a non-textual annotation.
- **Visual:** Underline.
- **Warning:** Avoid confusing users who may mistake it for a hyperlink.

<mark>

- **Description:** Represents text which is marked or highlighted for reference or notation purposes.
- **Visual:** Typically yellow highlight.

<bdi> (Bidirectional Isolate)

- **Description:** Tells the browser's bidirectional algorithm to treat the text it contains in isolation from its surrounding text.
- **Use Case:** Useful when embedding user-generated content (like a name) where the text direction (LTR or RTL) is unknown.

<bdo> (Bidirectional Override)

- **Description:** Overrides the current directionality of text, so that the text is rendered in a different direction.

- **Description:** A generic inline container for phrasing content, which does not inherently represent anything. It can be used to group elements for styling purposes (using class or id).

**
 (Line Break)**

- **Description:** Produces a line break in text (carriage-return). It is a void element.
- **Usage:** Use for poems or addresses, not for vertical spacing (margins).

<wbr> (Word Break Opportunity)

- **Description:** Specifies where in a text it would be ok to add a line-break.¹²

8. Image and Multimedia

HTML5 introduced native support for audio and video, significantly reducing the reliance on third-party plugins like Flash.

- **DOM Interface:** HTMLImageElement
- **Category:** Embedded content.
- **Description:** Embeds an image into the document.

- **Attributes:**
 - src: The path to the image.
 - alt: Alternative text description (Mandatory for accessibility).
 - srcset: Defines a set of images to allow the browser to choose the best one for the screen resolution/density.
 - sizes: Defines the slot width for the image to help the browser pick from srcset.
 - loading: lazy (default for performance) or eager.
 - fetchpriority: high, low, auto.¹⁰
 - decoding: sync, async, auto.
- **Example:**
HTML

```

```

<audio>

- **DOM Interface:** HTMLAudioElement
- **Category:** Embedded content.
- **Description:** Used to embed sound content in documents.
- **Attributes:** src, controls, autoplay, loop, muted, preload.

<video>

- **DOM Interface:** HTMLVideoElement
- **Category:** Embedded content.
- **Description:** Embeds a media player which supports video playback into the document.
- **Attributes:** poster (an image shown while downloading), controls, playsinline (for mobile), width, height.

<track>

- **Category:** Embedded content.
- **Description:** Used as a child of the media elements, <audio> and <video>. It lets you specify timed text tracks (or time-based data), for example to automatically handle subtitles.
- **Attributes:** kind (subtitles, captions, descriptions, chapters, metadata), src, srclang, label, default.⁸

<map> and <area>

- **Category:** Embedded content.
- **Description:** Together, these define an image map (a clickable link area). <map> defines

the map, and <area> defines the clickable regions (rectangles, circles, polygons).¹⁷

9. Embedded Content

Beyond simple images, these elements allow embedding of complex external resources.

<iframe>

- **Description:** Represents a nested browsing context (embedding another HTML page).
- **Attributes:** src, srcdoc, sandbox (enables a set of extra restrictions for the content), allow (Feature Policy), loading, width, height.¹⁰

<embed>

- **Description:** Embeds external content at the specified point in the document. Historically used for plug-ins (like Flash).

<object>

- **Description:** Represents an external resource, which can be treated as an image, a nested browsing context, or a resource to be handled by a plugin.
- **Attributes:** data, type.

<param>

- **Description:** Defines parameters for an <object> element.

<picture>

- **Description:** Contains zero or more <source> elements and one element to offer alternative image versions for different display/device scenarios (Art Direction).
- **Example:**
HTML

```
<picture>  
  <source media="(min-width: 800px)" srcset="headshot.jpg">  
  <source media="(min-width: 450px)" srcset="headshot-small.jpg">  
    
</picture>
```

<source>

- **Description:** Specifies multiple media resources for the <picture>, <audio>, or <video> elements.¹⁹

<fencedframe> (Experimental)

- **Description:** Similar to an iframe, but restricts communication with its embedding

context to allow the frame access to cross-site data without sharing it with the embedding page. Part of the Privacy Sandbox.²⁰

<portal> (Experimental)

- **Description:** Allows for the embedding of another page that the user can seamlessly navigate into.

10. Scripting and Graphics

<script>

- **Description:** Used to embed executable code or data (typically JavaScript).
- **Attributes:**
 - **src:** URI of the script.
 - **type:** Mime type. `type="module"` enables ES6 modules.
 - **async:** Execute script asynchronously as soon as it is available.
 - **defer:** Execute script when the page has finished parsing.
 - **integrity:** Subresource Integrity (SRI) hash.
 - **crossorigin:** CORS settings.

<noscript>

- **Description:** Defines a section of HTML to be inserted if a script type on the page is unsupported or if scripting is currently turned off in the browser.

<canvas>

- **Description:** Container element used to draw graphics via scripting (usually JavaScript, either Canvas 2D or WebGL context).
- **Attributes:** width, height.

11. Table Content

Tables are for tabular data *only*, not for layout.

- **<table>:** The wrapper.
- **<caption>:** The title of the table.
- **<colgroup>:** A container for `<col>` elements.
- **<col>:** Defines attribute values for one or more columns.
- **<thead>:** The header row(s).
- **<tbody>:** The body row(s).
- **<tfoot>:** The footer row(s).
- **<tr>:** A table row.

- **<th>**: A header cell. Bold by default. Important attribute: scope (row/col) for accessibility.
 - **<td>**: A standard data cell. Important attributes: colspan, rowspan.
-

12. Forms and Input

Forms interact with users to collect data.

<form>

- **Description:** Represents a document section containing interactive controls for submitting information.
- **Attributes:** action (URL), method (GET/POST), enctype (for file uploads use multipart/form-data), novalidate.¹⁰

<label>

- **Description:** Represents a caption for an item in a user interface.
- **Attributes:** for. The value of for must match the id of the input. This is critical for accessibility.

<input>

- **Description:** The most complex element in HTML, used to create interactive controls.
- **Key Attributes:** type, name, value, placeholder, required, pattern, min, max, step, disabled, readonly.
- **Input Types:**
 - text, password, email (validates email), number (numeric keypad), search, tel (phone keypad), url.
 - checkbox, radio.
 - date, time, datetime-local, month, week.
 - file (uploads), color (picker), range (slider).
 - hidden (invisible data), submit, reset, button, image.

<button>

- **Description:** A clickable button.
- **Attributes:** type (submit, reset, button).
- **Advantage:** Unlike <input type="button">, <button> can contain content (like icons or bold text).

<select>, <optgroup>, <option>

- **Description:** Create a dropdown menu. <optgroup> groups options; <option> defines the individual choices.

<datalist>

- **Description:** Contains a set of <option> elements that represent the permissible or recommended options available to choose from within other controls (like an autocomplete).

<textarea>

- **Description:** Represents a multi-line plain-text editing control.

<output>

- **Description:** Container element into which a site or app can inject the results of a calculation.

<progress>

- **Description:** Displays an indicator showing the completion progress of a task (e.g., download).

<meter>

- **Description:** Represents either a scalar value within a known range or a fractional value (e.g., disk usage, battery level).

<fieldset> and <legend>

- **Description:** <fieldset> groups related elements in a form; <legend> provides a caption for that group.

13. Interactive Elements

<details>

- **Description:** A disclosure widget from which the user can retrieve additional information or controls.
- **Attributes:** open.

<summary>

- **Description:** Specifies a summary, caption, or legend for a <details> element's disclosure box.

<dialog>

- **Description:** Represents a dialog box or other interactive component, such as a dismissible alert or modal.
- **API:** Controlled via .showModal() in JavaScript.

<search>

- **Description:** Represents a part of a document or application that contains a set of form controls or other content related to performing a search or filtering operation.¹⁵

14. Web Components (Shadow DOM)

<template>

- **Description:** A mechanism for holding HTML that is not to be rendered immediately when a page is loaded but may be instantiated subsequently during runtime using JavaScript.

<slot>

- **Description:** A placeholder inside a web component that you can fill with your own markup, which lets you create separate DOM trees and present them together.⁶

15. Obsolete and Deprecated Elements

These elements are part of the history of HTML but are obsolete in the Living Standard. They should **not** be used in modern development.

- <acronym> (Use <abbr>)
- <applet> (Use <embed> or <object>)
- <basefont>, , <center>, <big>, <strike>, <tt> (Use CSS for styling)
- <bgsound> (Use <audio>)
- <blink>, <marquee> (Accessibility violations; use CSS animations)
- <dir> (Use)
- <frame>, <frameset>, <noframes> (Use <iframe> and CSS Grid/Flexbox)
- <isindex> (Use <input>)
- <listing>, <xmp>, <plaintext> (Use <pre> and <code>)
- <hgroup> (Was deprecated, now reinstated, but check browser support)

16. Conclusion

The HTML Living Standard is a sophisticated, semantic language. By utilizing the approximately 145 valid elements correctly, developers ensure their applications are accessible to all users (including those on assistive devices), understandable by search engine algorithms, and performant on all devices. The shift from <div> soup to semantic structures (like <article>, <nav>, <aside>) is the hallmark of professional front-end engineering in 2026.