

HTML

MASTER COMPLETE HANDWRITTEN-STYLE NOTES

HyperText Markup Language — Full Study & Reference Notebook

This expanded edition is designed around the user's requirement that important HTML language features should not be skipped. It covers the HTML document model, all major/current standard HTML elements, attributes, forms, media, tables, semantics, accessibility, metadata, obsolete elements commonly seen in exams, examples, practical pages and revision questions.

Section	Coverage
Foundation	Web, HTML, DOM, syntax, parsing, conformance
Document	doctype, html, head, body, metadata, language
Text	headings, paragraphs, quotations, semantics, code
Navigation	a, URLs, fragments, targets, link relations
Media	img, picture, source, audio, video, track, iframe
Data	lists, tables, colgroup, spanning and captions
Forms	controls, all common input types, validation, submission
Semantics	header, nav, main, section, article, aside, footer
Advanced	details, dialog, template, canvas, SVG, data/time
Reference	elements, attributes, obsolete tags, cheat sheets
Exam	MCQs, short answers, long answers, practical questions

Important scope: This is a comprehensive HTML-language study/reference guide. It does not attempt to reproduce every browser-specific feature or every separate Web API.

Contents

1. HTML foundations and terminology
2. HTML syntax, parsing and document structure
3. Global attributes and common attribute patterns
4. Text content and semantic inline elements
5. Links, URLs and navigation
6. Images, responsive images and image maps
7. Lists and description lists
8. Tables and accessible tabular data
9. Forms – complete reference
10. Input types – complete reference
11. Form validation and submission
12. Audio, video, captions and embedded content
13. Semantic document structure
14. Head, metadata, link relations and resource hints
15. Special/advanced HTML elements
16. Accessibility and ARIA basics
17. HTML entities, character encoding and direction
18. Browser rendering, DOM and developer tools
19. Complete practical examples
20. Obsolete/deprecated HTML elements for exams
21. Element master reference
22. Attribute master reference
23. HTML exam questions – MCQs
24. Short and long-answer revision
25. Final one-page-style cheat sheet

1. HTML Foundations and Terminology

HTML stands for HyperText Markup Language. It is a markup language for structuring content and describing its meaning. A browser parses HTML and creates a document representation that scripts and browser features can interact with.

HTML / CSS / JavaScript

Technology	Main responsibility
HTML	Structure, content and semantics
CSS	Presentation, layout, visual styling
JavaScript	Behaviour, application logic and DOM/Web API interaction

Important terms

Term	Meaning
Element	A complete HTML construct, usually consisting of a start tag, content and end tag.
Tag	Markup syntax such as <p> or </p>.
Attribute	Additional information on an element, e.g. href, id or class.
DOM	Document Object Model representation of the document.
User agent	Software acting for a user, commonly a browser.
URL	Uniform Resource Locator used to identify/address a resource.
HTTP	Protocol commonly used to transfer Web resources.
HTTPS	HTTP protected by TLS at the transport/security layer.
Semantic HTML	Elements whose names communicate content meaning/role.
Void element	Element that cannot have child content and has no normal end tag.

Basic workflow

```

HTML source
↓
Browser parser
↓
Document / DOM
↓
Referenced resources
↓
Layout + rendering
↓
User interaction

```

Exam point: HTML is a markup language. Do not describe HTML itself as a programming language.

2. HTML Syntax, Parsing and Document Structure

Minimal valid document

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>My Page</title>
</head>
<body>
  <h1>Hello</h1>
  <p>Welcome.</p>
</body>
</html>

```

DOCTYPE

`<!doctype html>` is the HTML doctype declaration. In modern browsers it is used to trigger standards mode rather than quirks mode.

Elements

```
<p>Hello <strong>student</strong>.</p>
```

↑
↑
↑

start tag
nested element
end tag

Void elements

Element	Typical use
<code>area</code>	Hotspot in an image map
<code>base</code>	Base URL/target
<code>br</code>	Line break
<code>col</code>	Table column definition
<code>embed</code>	Embedded external content
<code>hr</code>	Thematic break
<code>img</code>	Image
<code>input</code>	Form control
<code>link</code>	External resource/link relationship
<code>meta</code>	Metadata
<code>param</code>	Legacy object parameter; obsolete in modern HTML
<code>source</code>	Media/source candidate
<code>track</code>	Text track
<code>wbr</code>	Optional word-break opportunity

The list above includes `param` only because it is historically relevant; it is obsolete in current HTML.

Nesting

```
<p><em>Correctly nested</em> text.</p>
```

- Close elements in the reverse order in which they were opened.
- Do not invent end tags for void elements.
- Attribute values should normally be quoted.
- Use lowercase HTML syntax consistently for readability, even though HTML is generally ASCII-case-insensitive for element/attribute names.

Whitespace

HTML source whitespace is generally collapsed in ordinary phrasing content during rendering. The `pre` element preserves whitespace and line breaks for preformatted text.

3. Global Attributes and Common Attribute Patterns

Attribute	Purpose / notes
<code>id</code>	Unique identifier in a document.
<code>class</code>	Space-separated reusable class names.
<code>title</code>	Advisory information.
<code>lang</code>	Language of an element's content.
<code>dir</code>	Text direction: <code>ltr</code> , <code>rtl</code> or <code>auto</code> .
<code>hidden</code>	Marks content as not currently relevant/visible.
<code>tabindex</code>	Controls focusability/order; avoid positive <code>tabindex</code> values in normal accessible design.
<code>contenteditable</code>	Makes content editable by the user.
<code>spellcheck</code>	Hint to enable/disable spelling/grammar checking.
<code>draggable</code>	Drag-and-drop hint.

Attribute	Purpose / notes
<code>translate</code>	Hint whether content should be translated.
<code>data-*</code>	Custom data attributes for application use.
<code>slot</code>	Assigns content to a named slot in a web component.
<code>part</code>	Exposes a shadow-tree part for styling by external CSS.
<code>style</code>	Inline CSS declaration.
<code>accesskey</code>	Keyboard shortcut hint; use carefully because shortcuts can conflict.
<code>autocapitalize</code>	Hint for capitalization behaviour on supported controls.
<code>autofocus</code>	Requests initial focus; use sparingly.
<code>inert</code>	Makes a subtree non-interactive for user interaction/focus.
<code>popover</code>	Associates an element with the popover mechanism.

Boolean attributes

```
<input required>
<button disabled>Save</button>
<details open>...</details>
```

For a boolean attribute, presence normally represents the true state. Writing `disabled="false"` does not make a boolean attribute false.

`data-*`

```
<article data-subject="html" data-semester="5">
  <h2>HTML Notes</h2>
</article>
```

Custom data attributes are intended for private application data. They are accessible to scripts through DOM APIs such as `dataset`.

4. Text Content and Semantic Inline Elements

Element	Meaning / typical use
<code>h1-h6</code>	Six levels of headings.
<code>p</code>	Paragraph.
<code>br</code>	Line break.
<code>hr</code>	Thematic break.
<code>strong</code>	Strong importance.
<code>em</code>	Stress emphasis.
<code>b</code>	Attention without adding strong importance.
<code>i</code>	Text in an alternate voice/mood/technical term, depending on context.
<code>u</code>	Non-textual annotation.
<code>mark</code>	Highlighted/relevant text.
<code>small</code>	Side comments/small print.
<code>s</code>	No-longer-accurate/relevant text.
<code>del</code>	Deleted content.
<code>ins</code>	Inserted content.
<code>sub</code>	Subscript.
<code>sup</code>	Superscript.
<code>abbr</code>	Abbreviation/acronym.
<code>cite</code>	Title of a creative work.
<code>q</code>	Short inline quotation.
<code>blockquote</code>	Extended quotation.
<code>dfn</code>	Term being defined.
<code>time</code>	Machine-readable date/time.

Element	Meaning / typical use
<code>data</code>	Human-readable content with machine-readable value.
<code>code</code>	Code fragment.
<code>pre</code>	Preformatted block.
<code>kbd</code>	User keyboard/input.
<code>samp</code>	Sample program output.
<code>var</code>	Variable.
<code>span</code>	Generic phrasing container.
<code>bdi</code>	Isolates bidirectional text.
<code>bdo</code>	Overrides text direction.
<code>wbr</code>	Word-break opportunity.
<code>ruby</code>	Ruby annotation container.
<code>rt</code>	Ruby annotation text.
<code>rp</code>	Fallback punctuation for ruby.

Examples

```
<p><strong>Important:</strong> submit before <time datetime="2026-10-01">1 October</time>.</p>
<p>Water: H<sub>2</sub>O. Area: 10m<sup>2</sup>.</p>
<p><abbr title="HyperText Markup Language">HTML</abbr></p>
<blockquote cite="https://example.com/source">
  A quoted passage.
</blockquote>
<pre><code>print("Hello")</code></pre>
```

Semantic vs purely visual

Prefer `strong` or `em` when the meaning is importance/emphasis. CSS should normally handle purely visual requirements such as font size, weight or spacing.

5. Links, URLs and Navigation

Anchor

```
<a href="https://example.com">Visit site</a>
<a href="notes.html">Relative page</a>
<a href="#faq">Jump to FAQ</a>
```

URL/path	Example	Meaning
Absolute	<code>https://example.com/notes.html</code>	Complete URL.
Relative	<code>notes.html</code>	Resolved relative to the current document URL.
Root-relative	<code>/images/logo.png</code>	Resolved from the origin's root path.
Fragment	<code>#faq</code>	Identifies a location in the current document/resource.
Mail	<code>mailto:student@example.com</code>	Email URI.
Telephone	<code>tel:+911234567890</code>	Telephone URI.

Important anchor attributes

Attribute	Use
<code>href</code>	Destination URL.
<code>target</code>	Browsing context target.
<code>rel</code>	Relationship/security/referrer hints.
<code>download</code>	Requests download for applicable same-origin/blob/data resources depending on browser rules.
<code>hreflang</code>	Language of linked resource.
<code>type</code>	Advisory MIME type.
<code>ping</code>	URLs to notify when link is activated; privacy considerations apply.

Attribute	Use
<code>referrerpolicy</code>	Controls referrer information for the navigation.

target values

`_self`, `_blank`, `_parent`, `_top`, plus named browsing contexts.

```
<a href="https://example.com"
  target="_blank"
  rel="noopener">Open in a new context</a>
```

Base URL

```
<head>
  <base href="https://example.com/docs/">
</head>
<body>
  <a href="page.html">Resolves under the base URL</a>
</body>
```

Use carefully: it changes how relative URLs in the document are resolved.

6. Images, Responsive Images and Image Maps

Image

```

```

Feature	Purpose
<code>src</code>	Image resource URL.
<code>alt</code>	Alternative text; essential for meaningful images.
<code>width/height</code>	Intrinsic/display dimension hints.
<code>loading</code>	Eager/lazy loading hint.
<code>decoding</code>	Image decoding hint.
<code>ismap</code>	Server-side image map hint when used with a suitable link.
<code>usemap</code>	Connects image to a client-side image map.

Figure

```
<figure>
  
  <figcaption>HTML document structure</figcaption>
</figure>
```

Picture / source

```
<picture>
  <source media="(max-width: 600px)" srcset="small.jpg">
  <source type="image/avif" srcset="photo.avif">
  
</picture>
```

Responsive srcset

```

```

Image map

```


<map name="campus">
  <area shape="rect" coords="20,20,180,120"
    href="library.html" alt="Library">
  <area shape="circle" coords="300,100,50"
    href="lab.html" alt="Computer lab">
</map>
```

map defines an image map and area defines clickable regions.

7. Lists

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

<ol start="1">
  <li>Open editor</li>
  <li>Write HTML</li>
</ol>

<dl>
  <dt>HTML</dt>
  <dd>Markup language for Web structure.</dd>
</dl>
```

Element	Important attributes / role
<code>ul</code>	Unordered list.
<code>ol</code>	Ordered list; start, reversed, type can affect numbering.
<code>li</code>	List item; value can set an item's ordinal value in ordered lists.
<code>dl</code>	Description list.
<code>dt</code>	Term/name.
<code>dd</code>	Description/value.

Ordered list attributes

start sets starting number, reversed reverses numbering, and type selects a numbering style. CSS is normally preferred for presentation.

8. Tables and Accessible Tabular Data

```
<table>
  <caption>Semester Marks</caption>
  <colgroup>
    <col>
    <col span="2">
  </colgroup>
  <thead>
    <tr>
      <th scope="col">Student</th>
      <th scope="col">HTML</th>
      <th scope="col">CSS</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">Jojin</th>
      <td>90</td>
      <td>88</td>
    </tr>
  </tbody>
  <tfoot>
    <tr><td colspan="3">End</td></tr>
  </tfoot>
</table>
```

Element/attribute	Purpose
<code>table</code>	Tabular data container.
<code>caption</code>	Table title/caption.
<code>thead</code>	Header rows.
<code>tbody</code>	Main rows.
<code>tfoot</code>	Footer/summary rows.
<code>tr</code>	Table row.
<code>th</code>	Header cell.
<code>td</code>	Data cell.
<code>scope</code>	Associates simple header with row/column.
<code>colgroup</code>	Groups columns.
<code>col</code>	Column definition.
<code>colspan</code>	Cell spans multiple columns.
<code>rowspan</code>	Cell spans multiple rows.
<code>headers</code>	Associates a cell with one or more header ids; useful for complex tables.

Rules

- Tables are for data, not page layout.
- Use `th` for header cells.
- Use `caption` where a table needs a clear title.
- Use `scope` for straightforward row/column headers.
- Use `headers/id` relationships for complex header structures.

9. Forms — Complete Reference

Basic form

```
<form action="/register" method="post">
  <fieldset>
    <legend>Student</legend>

    <label for="name">Name</label>
    <input id="name" name="name" type="text" required>

    <label for="course">Course</label>
    <select id="course" name="course">
      <option value="bca">BCA</option>
      <option value="mca">MCA</option>
    </select>

    <button type="submit">Register</button>
  </fieldset>
</form>
```

Element	Purpose
<code>form</code>	Form-associated element containing controls.
<code>label</code>	Label for a control.
<code>input</code>	Many types of form controls.
<code>button</code>	Button control.
<code>select</code>	Selection control.
<code>option</code>	Option in select/datalist.
<code>optgroup</code>	Group of options in select.
<code>textarea</code>	Multiline text input.
<code>fieldset</code>	Groups controls.
<code>legend</code>	Caption for fieldset.
<code>datalist</code>	Suggested options for an input.
<code>output</code>	Represents calculation/result.

Form attributes

Attribute	Meaning
<i>action</i>	URL to which submission is sent.
<i>method</i>	get, post or dialog in appropriate contexts.
<i>enctype</i>	Encoding for submitted data: application/x-www-form-urlencoded, multipart/form-data, text/plain.
<i>autocomplete</i>	Hint about automatic completion.
<i>novalidate</i>	Disables built-in constraint validation on submission.
<i>name</i>	Form name.
<i>target</i>	Submission browsing context.
<i>accept-charset</i>	Accepted character encodings; UTF-8 is standard practice.
<i>rel</i>	Relationship hints for the submission/navigation context.

Label association

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

<!-- Or wrap the control -->
<label>
  Subscribe <input type="checkbox" name="subscribe">
</label>
```

Fieldset

Use *fieldset* and *legend* to group related controls, especially radio-button groups and logically related fields.

10. Input Types – Complete Reference

type	Typical use
<i>hidden</i>	Hidden submitted value; not a security mechanism.
<i>text</i>	Single-line text.
<i>search</i>	Search text.
<i>tel</i>	Telephone input.
<i>url</i>	URL input.
<i>email</i>	Email address.
<i>password</i>	Password entry.
<i>date</i>	Date.
<i>month</i>	Month.
<i>week</i>	Week.
<i>time</i>	Time.
<i>datetime-local</i>	Local date/time.
<i>number</i>	Number.
<i>range</i>	Range control.
<i>color</i>	Color chooser.
<i>checkbox</i>	Independent boolean/multiple selection.
<i>radio</i>	One choice in a named group.
<i>file</i>	File selection.
<i>submit</i>	Submit form.
<i>image</i>	Image submit button.
<i>reset</i>	Reset form.
<i>button</i>	Generic button.

Important input attributes

Attribute	Purpose
<code>name</code>	Submitted field name.
<code>value</code>	Initial/submitted value depending on type.
<code>placeholder</code>	Short hint; not a label replacement.
<code>required</code>	Constraint: value required.
<code>readonly</code>	User cannot edit value; value can still be submitted in relevant controls.
<code>disabled</code>	Control disabled; generally not submitted.
<code>checked</code>	Initial checked state for checkbox/radio.
<code>selected</code>	Initial option selection; belongs to option.
<code>multiple</code>	Allows multiple values for supported controls.
<code>min/max</code>	Numeric/date limits.
<code>step</code>	Allowed increment.
<code>minlength/maxlength</code>	Text length limits.
<code>pattern</code>	Regular-expression constraint for applicable text inputs.
<code>accept</code>	File type hint.
<code>autocomplete</code>	Autofill hint.
<code>autofocus</code>	Requests initial focus.
<code>list</code>	Associates input with a datalist.
<code>form</code>	Associates control with a form by id.
<code>formaction</code>	Submit destination override for submit controls.
<code>formenctype</code>	Encoding override.
<code>formmethod</code>	Method override.
<code>formnovalidate</code>	Validation override.
<code>formtarget</code>	Target override.

Radio groups

```
<label><input type="radio" name="semester" value="1"> 1</label>
<label><input type="radio" name="semester" value="2"> 2</label>
```

Radio buttons with the same name form a group; normally only one can be selected.

11. Form Validation and Submission

Built-in constraints

```
<input type="email" required>
<input type="number" min="1" max="100" step="1">
<input type="text" minlength="3" maxlength="50">
<input type="text" pattern="[A-Za-z ]+>
```

Constraint	Applies to
<code>required</code>	Many form controls
<code>min/max</code>	Numeric and applicable date/time types
<code>step</code>	Numeric and applicable date/time types
<code>minlength/maxlength</code>	Text-like controls
<code>pattern</code>	Applicable text-like inputs
<code>type</code>	Built-in syntax constraints for types such as email/url
<code>accept</code>	File input hint, not complete server validation

Output

```
<form oninput="result.value = Number(a.value) + Number(b.value)">
  <input id="a" name="a" type="number" value="10">
  <input id="b" name="b" type="number" value="20">
  <output name="result">30</output>
</form>
```

The example uses JavaScript for the calculation; output is the HTML element used to represent the result.

Submission encoding

Encoding	Typical use
<code>application/x-www-form-urlencoded</code>	Default form encoding for most ordinary submissions.
<code>multipart/form-data</code>	Required for file uploads.
<code>text/plain</code>	Plain-text form submission; rarely appropriate for production protocols.

Security: HTML validation improves user experience but does not replace server-side validation, authorization, CSRF protections where applicable, safe file handling, output encoding and other server-side security controls.

12. Audio, Video, Captions and Embedded Content

Audio

```
<audio controls>
  <source src="lecture.mp3" type="audio/mpeg">
  Your browser does not support audio.
</audio>
```

Video

```
<video controls width="640" poster="preview.jpg">
  <source src="lecture.mp4" type="video/mp4">
  Your browser does not support video.
</video>
```

Track / captions

```
<video controls>
  <source src="lecture.mp4" type="video/mp4">
  <track kind="captions"
        src="captions.vtt"
        srclang="en"
        label="English">
</video>
```

Media feature	Notes
<code>controls</code>	Displays user controls.
<code>autoplay</code>	Requests automatic playback; browser policy may restrict it.
<code>muted</code>	Starts media muted when applicable.
<code>loop</code>	Repeats playback.
<code>poster</code>	Video poster image.
<code>preload</code>	Loading hint: none, metadata or auto.
<code>source</code>	Provides media candidate(s).
<code>track</code>	Timed text such as captions/subtitles/chapters.

iframe

```
<iframe src="https://example.com"
  title="Example"
  width="600" height="400"
  loading="lazy"
  sandbox></iframe>
```

Useful iframe attributes include `src`, `srcdoc`, `name`, `title`, `width`, `height`, `loading`, `allow`, `allowfullscreen`, `referrerpolicy` and `sandbox`.

object / embed

```
<object data="document.pdf"
        type="application/pdf"
        width="600" height="500">
  PDF fallback content
</object>

<embed src="document.pdf"
        type="application/pdf">
```

object and embed can embed external resources. Browser/plugin support varies; modern applications should use appropriate platform-supported embedding approaches.

13. Semantic Document Structure

```
<header>...</header>
<nav aria-label="Main">...</nav>

<main>
  <section>
    <h2>Notes</h2>
    <article>
      <h3>HTML</h3>
      <p>...</p>
    </article>
  </section>

  <aside>Related resources</aside>
</main>

<footer>...</footer>
```

Element	Meaning
<i>header</i>	Introductory/navigation content for a page or section.
<i>nav</i>	Major navigation links.
<i>main</i>	Dominant content of the document.
<i>section</i>	Thematic grouping, normally with a heading.
<i>article</i>	Self-contained composition that could stand independently.
<i>aside</i>	Tangentially related content.
<i>footer</i>	Footer for a page or section.
<i>address</i>	Contact information for relevant page/article.
<i>search</i>	Container for search controls (modern HTML).
<i>div</i>	Generic flow container with no inherent semantics.
<i>span</i>	Generic phrasing container with no inherent semantics.

Other semantic containers

figure/figcaption group self-contained content and its caption. address is for contact information. search can identify a search-related region.

14. Head, Metadata, Link Relations and Resource Hints

Typical head

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
        content="width=device-width, initial-scale=1.0">
  <title>StudyHub HTML Notes</title>
  <meta name="description"
        content="Complete HTML study notes">
  <link rel="icon" href="/favicon.ico">
  <link rel="stylesheet" href="/styles.css">
</head>
```

Element	Important use
<code>title</code>	Document title.
<code>meta</code>	Metadata.
<code>link</code>	External resource/relationship.
<code>style</code>	Embedded CSS.
<code>base</code>	Base URL/target for relative URLs.
<code>script</code>	Script resource/inline script; scripting is not part of HTML's markup syntax itself.
<code>noscript</code>	Fallback content when scripting is unavailable/disabled.

Common meta names

Name	Purpose
<code>description</code>	Description metadata.
<code>viewport</code>	Viewport configuration.
<code>theme-color</code>	Hint for browser UI theming where supported.
<code>robots</code>	Crawler directives where honored; not a security mechanism.
<code>author</code>	Author metadata.
<code>generator</code>	Software used to generate content.
<code>referrer</code>	Referrer policy via meta when appropriate.

Common link rel values

`stylesheet`, `icon`, `canonical`, `alternate`, `author`, `help`, `license`, `manifest`, `next`, `prev`, `preconnect`, `dns-prefetch`, `preload`, `modulepreload`, `prefetch` and others exist with defined meanings or ecosystem support.

```
<link rel="preconnect" href="https://cdn.example.com">
<link rel="preload" href="/font.woff2"
      as="font" type="font/woff2" crossorigin>
<link rel="canonical" href="https://example.com/page">
```

Do not treat metadata such as robots directives or hidden HTML fields as a security boundary.

15. Special and Advanced HTML Elements

Element	Purpose
<code>details</code>	Disclosure widget.
<code>summary</code>	Label/summary for details.
<code>dialog</code>	Dialog/modal-style interactive container.
<code>popover attribute</code>	HTML popover mechanism; associated with elements rather than a separate element.
<code>template</code>	Inert HTML fragment for later cloning/instantiation.
<code>slot</code>	Placeholder in a web component shadow tree.
<code>canvas</code>	Scriptable bitmap drawing surface.
<code>svg</code>	SVG graphics embedded in HTML.
<code>math</code>	MathML root in HTML documents.
<code>progress</code>	Progress toward a goal.
<code>meter</code>	Scalar measurement within known range.
<code>output</code>	Result of a calculation/action.
<code>datalist</code>	Suggested input options.
<code>noscript</code>	Fallback when scripting is unavailable.
<code>picture</code>	Responsive/art-directed image selection.
<code>source</code>	Resource candidate for media/picture.
<code>track</code>	Timed text track.

Details

```
<details>
  <summary>What is HTML?</summary>
  <p>HTML structures Web content.</p>
</details>
```

Dialog

```
<dialog id="help">
  <p>Help information</p>
  <form method="dialog">
    <button>Close</button>
  </form>
</dialog>
```

Progress / meter

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

Canvas

```
<canvas id="chart" width="400" height="200">
  Chart fallback text
</canvas>
```

Canvas drawing is normally performed through JavaScript. Canvas itself is an HTML element.

SVG / MathML

```
<svg viewBox="0 0 100 100" aria-label="Circle">
  <circle cx="50" cy="50" r="40"></circle>
</svg>

<math>
  <mi>x</mi><mo>+</mo><mi>y</mi>
</math>
```

16. Accessibility and ARIA Basics

- Use semantic HTML first: button for buttons, a for links, input for inputs.
- Give meaningful images appropriate alt text; decorative images commonly use alt="".
- Associate labels with form controls.
- Use heading hierarchy to communicate structure.
- Make interactive controls keyboard accessible.
- Provide captions/subtitles for appropriate media.
- Do not use color alone to communicate essential information.
- Use lang correctly.
- Use table headers and relationships for tabular data.
- Use ARIA only when native HTML semantics are insufficient or need enhancement.

Accessible naming

```
<button aria-label="Close dialog">X</button>

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

Common ARIA attributes

Attribute	Typical role
aria-label	Accessible name when appropriate.
aria-labelledby	Names an element using another element's id.
aria-describedby	Associates descriptive text.

Attribute	Typical role
<code>aria-hidden</code>	Removes content from accessibility tree when appropriate; must be used carefully.
<code>aria-expanded</code>	Expanded/collapsed state.
<code>aria-controls</code>	Identifies controlled element.
<code>aria-current</code>	Current item in a set/navigation.
<code>aria-live</code>	Live-region announcement behaviour.
<code>role</code>	Explicit ARIA role; prefer native semantics when possible.

ARIA does not automatically add keyboard behaviour, focus management or event handling. Native HTML controls are usually safer and more complete.

17. HTML Entities, Character Encoding and Direction

Character encoding

```
<meta charset="UTF-8">
```

UTF-8 is the standard practical choice for modern HTML documents and supports a very large range of Unicode characters.

Common character references

Character	Reference
<	<
>	>
&	&
"	"
'	'
©	©
®	®
™	™
non-breaking space	

Bidirectional text

```
<p>English <bdi>عالم</bdi> text.</p>
<p><bdo dir="rtl">ABC 123</bdo></p>
```

bdi isolates bidirectional text so its direction does not unintentionally affect surrounding text. *bdo* explicitly overrides direction.

Language

```
<html lang="en">
<p lang="ml">മലയാളം</p>
</html>
```

18. Browser Rendering, DOM and Developer Tools

DOM

The DOM is an object representation of a document. HTML source is parsed into a document tree; the browser may correct malformed markup while parsing, so the inspected DOM can differ from the raw source.

```
Document
├── html
│   ├── head
│   │   └── title
│   └── body
│       ├── h1
│       └── p
```

Developer tools

- *Elements/Inspector: inspect the parsed DOM and attributes.*
- *Console: inspect JavaScript/runtime messages.*
- *Network: inspect requested resources, status codes, timing and headers.*
- *Application/Storage tools: inspect site storage and related resources.*
- *Accessibility tools: inspect roles, names, states and properties.*
- *View Source: inspect the delivered source, which may differ from the live DOM.*

Validation vs browser tolerance

Browsers are intentionally tolerant of many malformed documents, but browser recovery does not mean markup is conforming or maintainable. Validation and testing are useful for finding mistakes.

19. Complete Practical Examples

Example A — Student resource page

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
    content="width=device-width, initial-scale=1.0">
  <title>StudyHub Resources</title>
  <meta name="description"
    content="Student study resources">
</head>
<body>
<header>
  <h1>StudyHub</h1>
  <nav aria-label="Main navigation">
    <a href="#notes">Notes</a>
    <a href="#papers">Question Papers</a>
    <a href="#contact">Contact</a>
  </nav>
</header>

<main>
<section id="notes">
  <h2>Notes</h2>
  <article>
    <h3>HTML</h3>
    <p>Complete HTML notes.</p>
    <figure>
      
      <figcaption>HTML study material</figcaption>
    </figure>
  </article>
</section>

<section id="papers">
  <h2>Question Papers</h2>
  <ul>
    <li><a href="html-2025.pdf" download>HTML 2025</a></li>
    <li><a href="html-2024.pdf" download>HTML 2024</a></li>
  </ul>
</section>
</main>

<footer>
  <p>&copy; 2026 StudyHub</p>
</footer>
</body>
</html>
```

Example B – Complete form

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Student Registration</title>
</head>
<body>
<h1>Registration</h1>

<form action="/register" method="post"
  enctype="multipart/form-data">
  <fieldset>
    <legend>Personal details</legend>

    <label for="name">Name</label>
    <input id="name" name="name"
      type="text" required minlength="2">

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

    <label for="course">Course</label>
    <select id="course" name="course">
      <option value="bca">BCA</option>
      <option value="mca">MCA</option>
    </select>

    <label>
      <input type="checkbox" name="updates">
      Receive updates
    </label>

    <label for="resume">PDF</label>
    <input id="resume" name="resume"
      type="file" accept="application/pdf">

    <button type="submit">Submit</button>
    <button type="reset">Reset</button>
  </fieldset>
</form>
</body>
</html>

```

20. Obsolete / Deprecated HTML Elements — Exam Reference

These elements are included because older textbooks, question papers and MCQs may mention them. They should generally not be used in new HTML.

Obsolete element	Historical purpose / replacement
<i>acronym</i>	Acronym markup; use <i>abbr</i> where appropriate.
<i>applet</i>	Java applet embedding; obsolete.
<i>basefont</i>	Default font styling; use CSS.
<i>big</i>	Larger text; use CSS.
<i>center</i>	Centering; use CSS.
<i>content</i>	Old shadow DOM distribution element; obsolete.
<i>dir</i>	Directory list; use <i>ul</i> for ordinary lists.
<i>font</i>	Font styling; use CSS.
<i>frame</i>	Frameset document frame; obsolete.
<i>frameset</i>	Frame layout container; obsolete.
<i>image</i>	Old non-standard synonym historically seen for <i>img</i> ; use <i>img</i> .
<i>keygen</i>	Key-pair generation form control; obsolete.
<i>marquee</i>	Scrolling text; non-standard/obsolete.
<i>menuitem</i>	Historical context-menu command element; obsolete.

Obsolete element	Historical purpose / replacement
<code>multicol</code>	Non-standard experimental multi-column element.
<code>nobr</code>	Prevent line wrapping; use CSS white-space.
<code>noembed</code>	Old fallback for <code>embed</code> ; modern alternatives depend on resource.
<code>param</code>	Object parameter; obsolete in modern HTML.
<code>plaintext</code>	Historical raw-text behaviour; obsolete.
<code>rb</code>	Ruby base text grouping; obsolete in current HTML ruby model.
<code>rtc</code>	Ruby semantic annotation container; obsolete.
<code>shadow</code>	Old shadow DOM insertion point; obsolete.
<code>spacer</code>	Non-standard spacing element; use CSS.
<code>strike</code>	Strikethrough; use <code>del/s</code> where meaning fits.
<code>tt</code>	Teletype text; use <code>code/kbd/samp</code> or CSS based on meaning.
<code>xmp</code>	Raw preformatted HTML; obsolete; use <code>pre/code</code> and escape markup.

Exam strategy: if a question asks for an old HTML tag, identify its historical purpose, then state that modern HTML uses the appropriate semantic element/CSS alternative.

21. Element Master Reference

Element	Meaning
<code>html</code>	Root element
<code>head</code>	Document metadata container
<code>body</code>	Document body
<code>base</code>	Base URL/target
<code>link</code>	External resource/relationship
<code>meta</code>	Metadata
<code>style</code>	Embedded CSS
<code>title</code>	Document title
<code>address</code>	Contact information
<code>article</code>	Self-contained composition
<code>aside</code>	Tangential content
<code>footer</code>	Footer
<code>header</code>	Introductory content
<code>h1-h6</code>	Headings
<code>hgroup</code>	Heading + related content grouping
<code>main</code>	Main content
<code>nav</code>	Navigation
<code>section</code>	Thematic section
<code>search</code>	Search-related region
<code>blockquote</code>	Extended quotation
<code>dd</code>	Description/value
<code>div</code>	Generic flow container
<code>dl</code>	Description list
<code>dt</code>	Term/name
<code>figcaption</code>	Figure caption
<code>figure</code>	Figure
<code>hr</code>	Thematic break
<code>li</code>	List item
<code>menu</code>	Menu/list of commands/items

Element	Meaning
<code>ol</code>	Ordered list
<code>p</code>	Paragraph
<code>pre</code>	Preformatted text
<code>ul</code>	Unordered list
<code>a</code>	Hyperlink
<code>abbr</code>	Abbreviation
<code>b</code>	Attention text
<code>bdi</code>	Bidirectional isolation
<code>bdo</code>	Direction override
<code>br</code>	Line break
<code>cite</code>	Creative work title
<code>code</code>	Code fragment
<code>data</code>	Machine-readable value
<code>dfn</code>	Defined term
<code>em</code>	Stress emphasis
<code>i</code>	Alternate voice/technical term
<code>kbd</code>	User input
<code>mark</code>	Highlighted text
<code>q</code>	Inline quote
<code>rp</code>	Ruby fallback punctuation
<code>rt</code>	Ruby annotation
<code>ruby</code>	Ruby annotation container
<code>s</code>	No-longer-accurate text
<code>samp</code>	Sample output
<code>small</code>	Small print
<code>span</code>	Generic phrasing container
<code>strong</code>	Strong importance
<code>sub</code>	Subscript
<code>sup</code>	Superscript
<code>time</code>	Date/time
<code>u</code>	Non-textual annotation
<code>var</code>	Variable
<code>wbr</code>	Break opportunity
<code>area</code>	Image-map area
<code>audio</code>	Audio
<code>img</code>	Image
<code>map</code>	Image map
<code>track</code>	Timed text
<code>video</code>	Video
<code>embed</code>	Embedded external content
<code>iframe</code>	Nested browsing context
<code>object</code>	External resource/object
<code>picture</code>	Responsive/art-directed images
<code>source</code>	Media/image source
<code>svg</code>	SVG graphics
<code>canvas</code>	Scriptable drawing surface
<code>caption</code>	Table caption

Element	Meaning
<code>col</code>	Table column
<code>colgroup</code>	Column group
<code>table</code>	Tabular data
<code>tbody</code>	Table body
<code>td</code>	Table data cell
<code>tfoot</code>	Table footer
<code>th</code>	Table header cell
<code>thead</code>	Table head
<code>tr</code>	Table row
<code>button</code>	Button
<code>datalist</code>	Suggested input values
<code>fieldset</code>	Form control group
<code>form</code>	Form
<code>input</code>	Form control
<code>label</code>	Control label
<code>legend</code>	Fieldset caption
<code>meter</code>	Scalar measurement
<code>optgroup</code>	Option group
<code>option</code>	Option
<code>output</code>	Calculation/result
<code>progress</code>	Progress indicator
<code>select</code>	Selection control
<code>textarea</code>	Multiline input
<code>details</code>	Disclosure widget
<code>dialog</code>	Dialog
<code>summary</code>	Details summary
<code>template</code>	Inert template
<code>slot</code>	Web component slot
<code>math</code>	MathML
<code>noscript</code>	Fallback without scripting

22. Attribute Master Reference

Attribute	Purpose
<code>id</code>	Unique identifier
<code>class</code>	Reusable class names
<code>title</code>	Advisory information
<code>lang</code>	Language
<code>dir</code>	Text direction
<code>hidden</code>	Not currently relevant/visible
<code>tabindex</code>	Focus navigation
<code>contenteditable</code>	Editable content
<code>spellcheck</code>	Spelling/grammar hint
<code>translate</code>	Translation hint
<code>data-*</code>	Custom application data
<code>slot</code>	Assign shadow-tree slot
<code>part</code>	Shadow-tree styling hook
<code>style</code>	Inline CSS

Attribute	Purpose
<code>accesskey</code>	Keyboard shortcut hint
<code>draggable</code>	Drag hint
<code>inert</code>	Non-interactive subtree
<code>popover</code>	Popover mechanism attribute
<code>href</code>	Link destination
<code>target</code>	Browsing context target
<code>rel</code>	Relationship
<code>download</code>	Download request
<code>hreflang</code>	Linked resource language
<code>referrerpolicy</code>	Referrer control
<code>ping</code>	Link activation notification URLs
<code>src</code>	Resource URL
<code>alt</code>	Image alternative text
<code>width</code>	Width
<code>height</code>	Height
<code>loading</code>	Loading hint
<code>decoding</code>	Image decoding hint
<code>srcset</code>	Image candidates
<code>sizes</code>	Image candidate sizing
<code>usemap</code>	Image-map reference
<code>ismap</code>	Server-side image map hint
<code>action</code>	Form destination
<code>method</code>	Form method
<code>enctype</code>	Form encoding
<code>autocomplete</code>	Autofill hint
<code>novalidate</code>	Disable validation
<code>accept-charset</code>	Submission encoding
<code>name</code>	Name/value identifier
<code>value</code>	Control value
<code>placeholder</code>	Input hint
<code>required</code>	Required
<code>readonly</code>	Not user-editable
<code>disabled</code>	Disabled control
<code>checked</code>	Checked state
<code>multiple</code>	Multiple values
<code>min</code>	Minimum
<code>max</code>	Maximum
<code>step</code>	Increment
<code>minlength</code>	Minimum text length
<code>maxlength</code>	Maximum text length
<code>pattern</code>	Pattern constraint
<code>accept</code>	Accepted file types hint
<code>list</code>	Datalist reference
<code>form</code>	Form association
<code>formaction</code>	Submit destination override
<code>formenctype</code>	Encoding override
<code>formmethod</code>	Method override

Attribute	Purpose
<code>formnovalidate</code>	Validation override
<code>formtarget</code>	Target override
<code>selected</code>	Initial option selection
<code>rows</code>	Textarea height hint
<code>cols</code>	Textarea width hint
<code>wrap</code>	Textarea wrapping mode
<code>open</code>	Open state for details/dialog
<code>controls</code>	Media controls
<code>autoplay</code>	Autoplay request
<code>muted</code>	Muted state
<code>loop</code>	Repeat playback
<code>poster</code>	Video poster
<code>preload</code>	Media loading hint
<code>kind</code>	Track kind
<code>srclang</code>	Track language
<code>label</code>	Human-readable label
<code>sandbox</code>	Iframe restrictions
<code>srcdoc</code>	Inline iframe document
<code>allow</code>	Iframe permissions policy
<code>allowfullscreen</code>	Fullscreen permission
<code>cite</code>	Citation URL for quote/change
<code>datetime</code>	Machine-readable date/time
<code>scope</code>	Table header scope
<code>colspan</code>	Column span
<code>rowspan</code>	Row span
<code>headers</code>	Table header id references
<code>shape</code>	Image-map shape
<code>coords</code>	Image-map coordinates
<code>media</code>	Media query for source/link
<code>type</code>	Type/MIME hint
<code>charset</code>	Legacy/element-specific encoding attribute where applicable; prefer modern mechanisms

23. HTML Exam Questions – MCQs

Question	Options	Answer
Which language is used to structure Web documents?	A. HTML B. CSS C. SQL D. Python	A
Which element is the root of an HTML document?	A. body B. html C. head D. main	B
Which declaration is the modern HTML doctype?	A. B. C. D.	C
Which element contains metadata?	A. body B. head C. main D. footer	B
Which element defines a hyperlink?	A. link B. a C. href D. nav	B
Which attribute specifies an image URL?	A. href B. link C. src D. path	C
Which attribute provides alternative image text?	A. title B. alt C. text D. description	B
Which element creates an ordered list?	A. ul B. ol C. dl D. li	B
Which element defines a list item?	A. item B. li C. list D. dt	B
Which element defines a table row?	A. td B. th C. tr D. row	C
Which element defines a table header cell?	A. head B. th C. td D. theadcell	B
Which element creates a form?	A. input B. form C. fieldset D. submit	B
Which element associates visible text with a form control?	A. label B. legend C. caption D. title	A

Question	Options	Answer
Which input type is intended for email addresses?	A. mail B. email C. text-email D. address	B
Which attribute makes a control mandatory?	A. mandatory B. must C. required D. validate	C
Which input type allows selecting one option from a named group?	A. checkbox B. radio C. selectone D. option	B
Which element represents a video?	A. movie B. media C. video D. film	C
Which element supplies a media candidate?	A. source B. media-src C. track D. file	A
Which element supplies timed text?	A. captions B. track C. subtitle D. texttrack	B
Which element represents the main content?	A. content B. main C. primary D. center	B
Which element represents navigation links?	A. navigation B. navigate C. nav D. menu-link	C
Which element is a generic block/flow container?	A. span B. div C. container D. box	B
Which element is a generic phrasing container?	A. span B. inline C. text D. phrase	A
Which element displays a disclosure widget?	A. details B. disclose C. summary D. toggle	A
Which element labels a details disclosure?	A. label B. legend C. summary D. caption	C
Which element represents progress toward a goal?	A. meter B. progress C. status D. range	B
Which element represents a scalar measurement?	A. meter B. progress C. measure D. scale	A
Which element is used for an image map container?	A. imagemap B. map C. area D. coords	B
Which element defines a clickable image-map region?	A. area B. region C. hotspot D. zone	A
Which attribute connects an image to an image map?	A. map B. usemap C. hrefmap D. imagemap	B
Which attribute provides a unique document identifier?	A. class B. id C. key D. nameid	B
Which attribute can contain multiple reusable class names?	A. class B. classes C. group D. style	A
Which attribute stores custom application data?	A. custom-* B. app-* C. data-* D. meta-*	C
Which element represents a self-contained composition?	A. section B. article C. content D. item	B
Which element is for tangentially related content?	A. aside B. side C. related D. extra	A
Which element groups a figure caption with content?	A. caption B. figure + figcaption C. fig D. media-caption	B
Which attribute controls a link's browsing context?	A. window B. target C. context D. open	B
Which form encoding is normally used for file uploads?	A. text/plain B. multipart/form-data C. application/file D. upload/form	B
Which element groups related form controls?	A. group B. fieldset C. controls D. formgroup	B
Which element names a fieldset?	A. caption B. legend C. label D. title	B

More MCQs — rapid fire

Prompt	Answer
What is DOM?	Document Object Model
Which tag creates a paragraph?	p
Which tag creates a line break?	br
Which tag creates a horizontal/thematic break?	hr
Which tag is used for strong importance?	strong
Which tag is used for stress emphasis?	em
Which tag contains page metadata?	head
Which tag contains visible document content?	body
Which attribute identifies the link destination?	href
Which attribute identifies a form field for submission?	name
Which tag defines a form option?	option
Which tag defines a multiline text input?	textarea
Which element embeds another browsing context?	iframe
Which tag defines a table caption?	caption
Which tag defines a table body?	tbody
Which tag defines a table footer?	tfoot
Which element provides fallback content without scripting?	noscript

Prompt	Answer
Which element is used for a reusable inert HTML fragment?	<code>template</code>
Which element is used for a dialog?	<code>dialog</code>
Which element represents a code fragment?	<code>code</code>

24. Short and Long-Answer Revision

Short-answer questions

Question	Model answer
What is HTML?	HyperText Markup Language, a markup language used to structure Web content.
What is an element?	A complete HTML construct, usually with start tag, content and end tag, except void elements.
What is an attribute?	Additional information/configuration attached to an element's start tag.
What is the purpose of DOCTYPE?	It declares the document type and, in modern browsers, triggers standards mode.
What is semantic HTML?	HTML that communicates content meaning/role through appropriate elements.
What is the DOM?	An object/tree representation of a document.
What is a void element?	An element that cannot contain child content and has no normal end tag.
Difference between id and class?	id identifies one element uniquely in a document; class groups elements for reusable classification.
Difference between ol and ul?	ol represents ordered items; ul represents unordered items.
Difference between th and td?	th is a header cell; td is a data cell.
Difference between GET and POST?	GET commonly places form data in the URL; POST sends it in the request body.
What is alt?	Alternative text for an image when appropriate.
What is iframe?	An embedded browsing context.
What is an HTML form?	A group of controls used to collect/submit user data.
What is accessibility?	Designing content and controls so people with different abilities can use them.

Long-answer questions

- Explain the complete structure of an HTML document with an example.
- Explain HTML elements, tags, attributes and nesting with examples.
- Explain different types of HTML lists.
- Explain HTML tables, their elements and cell spanning.
- Explain HTML forms and all major form controls.
- Explain HTML input types and important input attributes.
- Explain form validation and submission methods.
- Explain semantic HTML5 elements with a suitable page layout.
- Explain hyperlinks, absolute/relative URLs and fragments.
- Explain images, image maps and responsive images.
- Explain audio, video, source and track.
- Explain metadata and the contents of the head element.
- Explain accessibility practices in HTML.
- Explain obsolete HTML elements and their modern alternatives.
- Create a complete student registration HTML page.

Practical programming questions

Practise writing these without looking at the notes:

- Create a six-heading-level demonstration page.
- Create an internal navigation menu using fragment identifiers.
- Create a table with caption, thead, tbody, tfoot, th and td.
- Create a registration form with text, email, password, radio, checkbox, select, textarea and file controls.
- Create a responsive image using picture/source.
- Create audio/video with multiple source elements and captions.

- Create a semantic page using header/nav/main/section/article/aside/footer.
- Create a page containing details/summary and dialog.
- Create an image map with map and area.
- Create a complete StudyHub-style resource page.

25. Final HTML Cheat Sheet

Core skeleton

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
    content="width=device-width, initial-scale=1.0">
  <title>Page title</title>
</head>
<body>
  ...
</body>
</html>
```

Remember the element families

Family	Key elements
Document	html, head, body, title, meta, link, base, style, script, noscript
Structure	header, nav, main, section, article, aside, footer, address, search, div
Text	h1–h6, p, strong, em, b, i, mark, small, s, del, ins, sub, sup, abbr, cite, q, blockquote
Code/text	code, pre, kbd, samp, var, dfn, data, time, span, bdi, bdo, ruby, rt, rp, wbr
Lists	ul, ol, li, dl, dt, dd, menu
Links	a, base, link
Images	img, picture, source, figure, figcaption, map, area
Tables	table, caption, colgroup, col, thead, tbody, tfoot, tr, th, td
Forms	form, label, input, button, select, option, optgroup, textarea, fieldset, legend, datalist, output
Media	audio, video, source, track
Embedding	iframe, object, embed
Interactive	details, summary, dialog
Advanced	template, slot, canvas, svg, math, progress, meter

Must-know attributes

Global: id, class, title, lang, dir, hidden, tabindex, data-*, contenteditable, spellcheck, inert

Links: href, target, rel, download, hreflang, referrerpolicy

Images: src, alt, width, height, srcset, sizes, loading, decoding, usemap

Forms: action, method, enctype, autocomplete, name, value, required, disabled, readonly, checked, multiple, min, max, step, minlength, maxlength, pattern, accept, list, form

Tables: scope, headers, colspan, rowspan

Media: controls, autoplay, muted, loop, poster, preload, kind, srclang

Iframe: src, srcdoc, title, sandbox, allow, allowfullscreen, loading, referrerpolicy

Final exam checklist

- DOCTYPE
- html + lang
- head + charset + title
- body
- Elements vs attributes
- Void elements

- Headings + paragraphs
- Links + URL types
- Images + alt
- Lists
- Tables
- Forms + input types
- Validation
- Audio/video/track
- Semantic HTML5
- Metadata
- Global attributes
- Accessibility
- Entities/UTF-8
- Obsolete tags for MCQs
- Practical HTML page

Master memory sequence: DOCUMENT → TEXT → LINKS → IMAGES → LISTS → TABLES → FORMS → MEDIA → SEMANTICS → METADATA → ACCESSIBILITY → PRACTICE.

StudyHub

HTML MASTER COMPLETE NOTES

End of notebook • Comprehensive HTML language study/reference edition.