HTML

Structure and meaning for the web.

01 Document scaffold

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport"
    content="width=device-width,
             initial-scale=1">
  <title>My notes</title>
</head>
<body><h1>My notes</h1></body>
</html>
```

The doctype enables standards mode. Set lang to the page language; title names the browser tab.

02 Headings & paragraphs

```
<h1>Travel journal</h1>
<h2>Day one</h2>
<p>A walk beside the river.</p>
<p><strong>Bring water.</strong></p>
```

h1-h6 form a heading hierarchy. strong adds importance; em adds stress emphasis.

03 Semantic layout

```
<header>Site introduction</header>
<nav aria-label="Main">Links</nav>
<main>
  <article>Journal entry</article>
</main>
<footer>Contact details</footer>
```

Use one visible main. article stands alone; section groups a theme with a heading. div and span add no meaning.

04 Links & page anchors

```
<a href="/about">About us</a>
<a href="#contact">Contact</a>
<h2 id="contact">Contact</h2>
<a href="https://example.com"
   target="_blank" rel="noopener">
  Example (new tab)
</a>
```

href gives the destination. / starts at the site root; # links to an id. Use descriptive text and warn when opening a new tab.

05 Images & alternatives

```
<img src="trail.jpg"
     alt="A pine trail beside a lake"
     width="640" height="420"
     loading="lazy">
<img src="divider.svg" alt="">
```

Use meaningful alt, or alt="" for decoration. Dimensions reserve space. Lazy-load below-the-fold images. img has no closing tag.

06 Lists & descriptions

```
<ul>
  <li>Tea</li>
  <li>Coffee</li>
</ul>
<dl>
  <dt>HTML</dt>
  <dd>Web page structure</dd>
</dl>
```

ul is unordered; ol is ordered. Both use li items. dl pairs terms (dt) with descriptions (dd).

07 Tables for data

```
<table>
  <caption>Basket</caption>
  <tr>
    <th scope="col">Item</th>
    <th scope="col">Qty</th>
  </tr>
  <tr><td>Pens</td><td>2</td></tr>
</table>
```

tr is a row; th is a header; td is a data cell. Use scope="row" for row headers. thead and tbody group rows. Do not use tables for layout.

08 Inline text & code

```
<em>Read this carefully.</em>
<code>&lt;main&gt;</code>
<pre><code>let n = 1;</code></pre>
<small>Terms apply.</small>
```

pre preserves whitespace; code marks code. Write < for <, > for >, and & for &. Use CSS for visual styling.

09 Common attributes

id="intro"	Unique document identifier
class="card"	Reusable class names
data-id="42"	Custom data for scripts
hidden	Hide irrelevant content
lang="fr"	Language of this content
title="Hint"	Advisory text, not a label

Quote attribute values. Boolean attributes are enabled by their presence: disabled="false" still disables a control.

HTML

Forms and interfaces people can use.

10 Forms & submission

```
<form action="/subscribe" method="post">
  <label for="email">Email</label>
  <input id="email" name="email"
         type="email" required>
  <button type="submit">Join</button>
</form>
```

name identifies submitted data; id connects a label. GET puts data in the URL; POST sends a body. The action needs a server handler.

11 Choose an input type

text / search	Single-line text / search
email / url	Basic format checking
password	Masked text, not encryption
number / range	Numeric entry / slider
date / time	Date / time values
checkbox / radio	Toggle / one in a group
file	File selection

inputmode hints at a keyboard; autocomplete at autofill. File uploads need POST and enctype="multipart/form-data".

12 Labels & grouped controls

```
<fieldset><legend>Updates</legend>
  <label><input type="checkbox"
    name="news" value="yes"> News</label>
</fieldset>
```

Wrap a control in label, or match label for to its id. fieldset and legend name a group. Radio buttons sharing a name allow one choice.

13 Validation & control state

```
<label for="qty">Quantity</label>
<input id="qty" name="qty" type="number"
       min="1" max="8" step="1" required>
<label for="handle">Handle</label>
<input id="handle" name="handle"
       minlength="3" maxlength="20">
```

required checks presence; min/max/step constrain numbers. minlength/maxlength constrain text. Always validate on the server too.

disabled controls are not submitted. readonly keeps supported controls uneditable and submitted. A placeholder is not a label.

14 Selects & multiline text

```
<label for="size">Size</label>
<select id="size" name="size">
  <option value="s">Small</option>
  <option value="m" selected>Medium</option>
</select>
<label for="note">Note</label>
<textarea id="note" name="note"
          rows="3">Hello!</textarea>
```

option value is submitted; selected sets the initial choice. textarea content sets its initial value, not a value attribute.

15 Buttons & native behavior

```
<button type="submit">Save</button>
<button type="button">Preview</button>
```

Set type explicitly. submit sends a form; button needs behavior such as a JavaScript handler. Use links to navigate and buttons for actions.

16 Audio & video

```
<video controls width="320">
  <source src="intro.mp4" type="video/mp4">
  <track kind="captions" src="en.vtt"
         srclang="en" label="English"
         default>
</video>
<audio controls src="intro.mp3"></audio>
```

Provide captions for spoken audio in video and transcripts for audio-only content. Avoid autoplay. controls exposes the native player.

17 Accessibility essentials

```
<a href="#main">Skip to content</a>
<main id="main">
  <h1>Account settings</h1>
</main>
<button type="button"
        aria-label="Close">X</button>
```

Start with native HTML and visible labels. An icon-only control needs an accessible name. Keep keyboard focus visible; avoid positive tabindex.

18 Styles, scripts & comments

```
<!-- Put the stylesheet in head -->
<link rel="stylesheet" href="styles.css">
<script src="app.js" defer></script>
<script type="module" src="main.js">
</script>
```

defer runs an external classic script after parsing, in document order. Modules defer by default. Comments are visible in page source.