

Learn • Code • Build

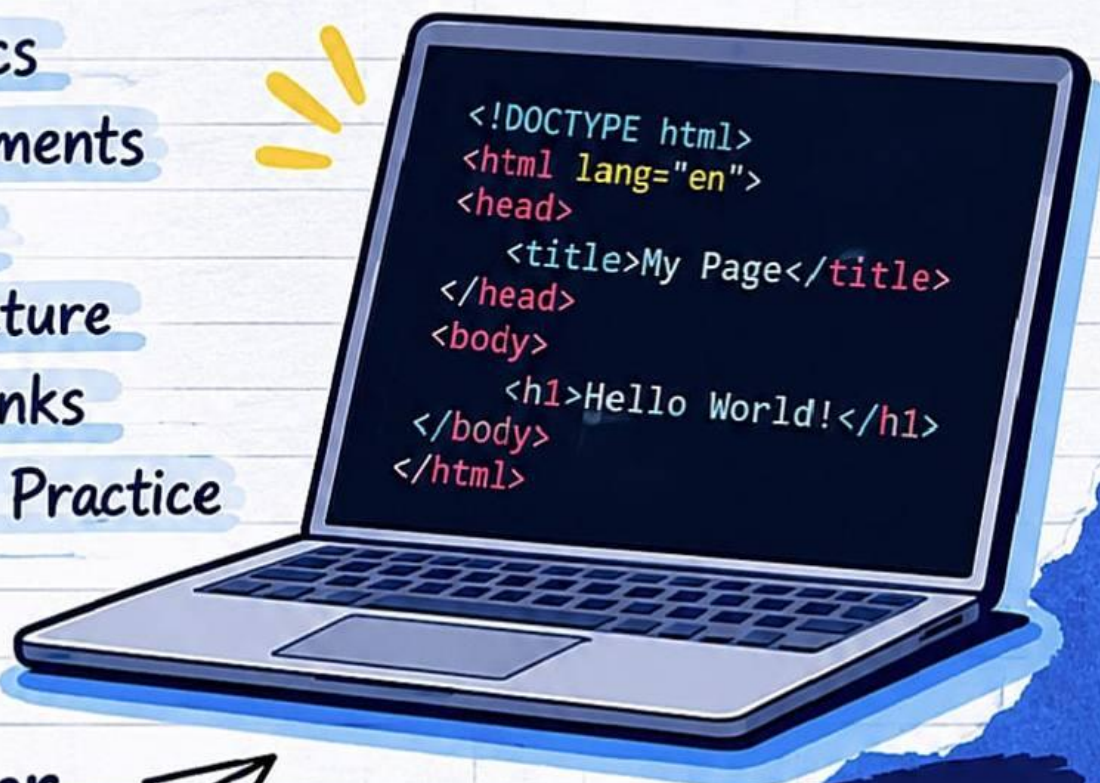
Web Development



HTML Notes

Complete Guide to HTML

- ✓ HTML Basics
- ✓ Tags & Elements
- ✓ Attributes
- ✓ Page Structure
- ✓ Forms & Links
- ✓ Examples & Practice



Dream Big
Code Better



ZainSpire

01/20 — HTML: THE STARTING POINT

HTML means HyperText Markup Language.

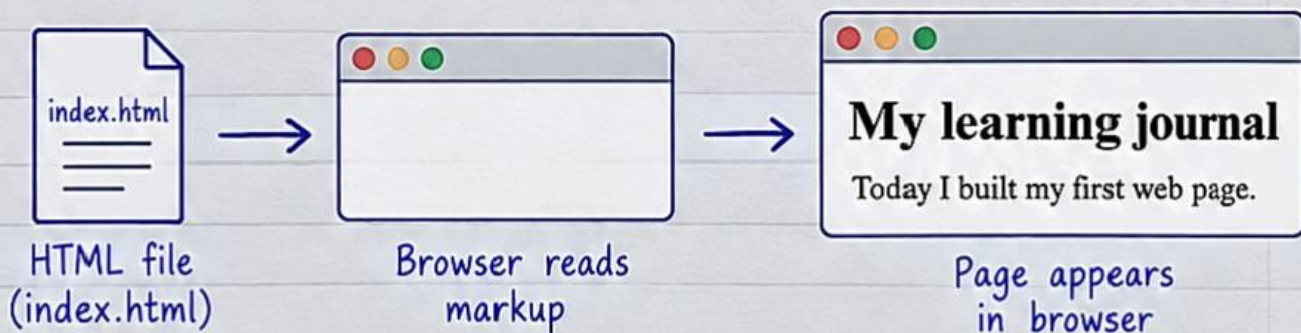
It describes the meaning and structure of web content: headings, paragraphs, links, images and forms.

HTML is a markup language, not a programming language.

HOW A PAGE OPENS

HTML file → browser reads markup → page appears.

Save a plain-text file as index.html, then open it in a browser. After editing, save and refresh.



YOUR FIRST EXAMPLE

```
<h1>My learning journal</h1>  
<p>Today I built my first web page.</p>
```

VISUAL RESULT

The visual result shows a browser window with the text "My learning journal" in a large, bold font, followed by "Today I built my first web page." in a smaller font. The browser window has a title bar with three colored dots (red, yellow, green).

The heading introduces the page. The paragraph contains its text.

THINK IN CONTENT

Ask: Is this a heading, a list, a link, or a form?

Choose an element because of its meaning, not just its appearance.

PRACTICE

Create index.html. Add one main heading and two paragraphs about a skill you want to learn.

Save it as .html, not .html.txt.

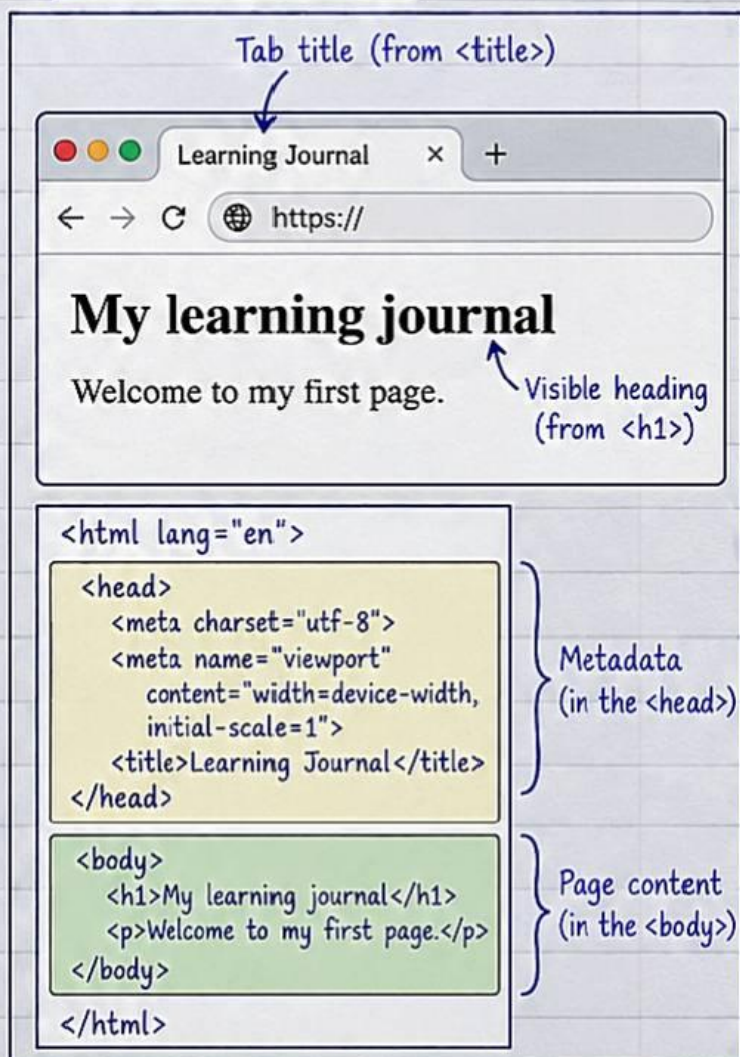
02 / 20 — A COMPLETE HTML DOCUMENT

STARTER FILE

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport"
      content="width=device-width,
        initial-scale=1">
    <title>Learning Journal</title>
  </head>

  <body>
    <h1>My learning journal</h1>
    <p>Welcome to my first page.</p>
  </body>
</html>
```

VISUAL RESULT



WHAT EACH PART DOES

<!DOCTYPE html>	asks for standards mode.
<html>	is the root element.
lang="en"	identifies the document language.
<head>	holds document metadata.
<title>	supplies the browser-tab title.
<body>	contains the page content.
UTF-8	supports a wide range of characters.
The viewport setting	helps mobile browsers size the page.

REMEMBER

<title> and <h1> have different jobs.
Use a useful tab title and a meaningful visible heading.

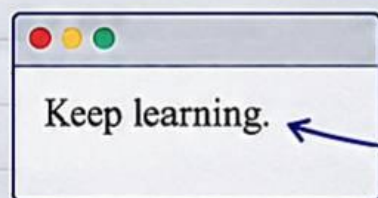
03 / 20 — TAGS, ELEMENTS & NESTING

ANATOMY

`<p>Keep learning.</p>`

↑
`<p>` = opening tag
Keep learning. = content
`</p>` = closing tag
The whole structure is an element.

VISUAL RESULT



← Renders as plain text (without tags).

NEST ELEMENTS CORRECTLY

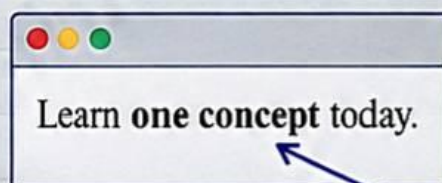
`<p>Learn <strong>one concept</strong> today.</p>`

Close the inner element before its parent.

Wrong: `<p><strong>Hello</p> </strong>`

Right: `<p><strong>Hello</strong></p>`

VISUAL RESULT



← Text inside `<strong>` renders bold.

VOID ELEMENTS

Some elements cannot contain children and have no end tag.

Examples: `<img>`, `<input>`, `<br>`, `<hr>`, `<meta>`, `<link>`.

Do not write `</img>` or `</input>`.

A trailing slash does not make ordinary HTML elements self-closing.

VISUAL RESULT

``



← Renders an image.

`<input type="text" placeholder="Name">`

Name

← Renders a text input.

`<br>` Line 1
Line 2

← Renders a line break.

`<hr>`

← Renders a horizontal rule.

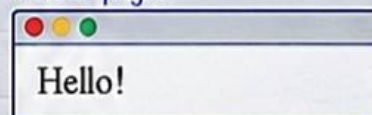
COMMENTS

`<!-- Explain a section here -->`

Comments are not displayed as page text. They remain visible in the page source: never put secrets in them.

VISUAL RESULT

Visible page:



← Comment does not appear in the rendered page.

Page source (view source):

```
<html>
<!-- Explain a section here -->
<body>
  <p>Hello!</p>
</body>
</html>
```

← Comment is visible in the page source.

04 / 20 — HTML: ATTRIBUTES, IDs & CLASSES

ATTRIBUTES ADD INFORMATION

```
<a href="about.html">About me</a>
```

href is the attribute name.

"about.html" is its value.

Write attributes inside the opening tag;
quote their values.

VISUAL RESULT



This renders
as a link.

COMMON GLOBAL ATTRIBUTES

id: a unique identifier in the document.

class: a reusable list of class names.

lang: the language of an element's content.

dir: text direction, such as rtl.

hidden: hides content from normal presentation.

EXAMPLES

```
<section id="projects" class="featured work">
```

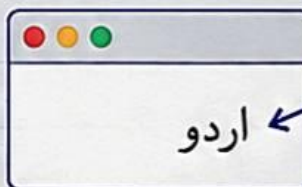
```
  <h2>Projects</h2>
```

```
</section>
```

```
<p lang="ur" dir="rtl">اردو</p>
```



id and class
are identifiers.
They are not
visible labels.



The text is in
Urdu and right
aligned.

BOOLEAN ATTRIBUTES

```
<input type="checkbox" checked>
```

Presence means true.

checked="false" still means checked.

Remove checked when you do not want it enabled.



Renders as a
checked checkbox
by default.

PRACTICE

Give one section a unique id.

Reuse one class on two paragraphs.

05 / 20 — HTML: HEADINGS & TEXT MEANING

BUILD A CLEAR OUTLINE

```
<h1>Learning journal</h1>
<h2>Week one</h2>
<h3>What I practised</h3>
```

Heading levels express hierarchy. Do not choose a level just for its size. One clear main heading is a useful default.

VISUAL RESULT

Learning journal ← `<h1>`

Week one ← `<h2>`

What I practised ← `<h3>`

TEXT ELEMENTS

`<p>` begins a paragraph.
`<strong>` marks importance.
`<em>` adds emphasis.
`<mark>` highlights relevant text.
`<small>` represents side comments.
`<sup>` and `<sub>` mark superscript and subscript.

VISUAL RESULT

Remember: save your work.

I really enjoyed this lesson.

Water is H₂O.

EXAMPLE

```
<p><strong>Remember:</strong>
  save your work.</p>
<p>I <em>really</em> enjoyed
  this lesson.</p>
<p>Water is H<sub>2</sub>O.</p>
```

BREAKS & SEPARATORS

`<br>` is a line break, useful in an address.
`<hr>` is a thematic break.
Do not repeat `<br>` to create page spacing. Ordinary whitespace in HTML text usually collapses.

VISUAL RESULT

My Street 10
Riverside
London

← Line breaks
with `<br>`

New section starts here.

← Thematic
break with
`<hr>`

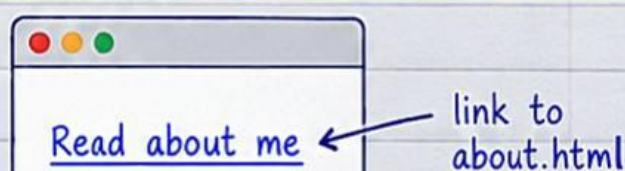
06/20 — HTML: LINKS & FILE PATHS

LINK TO ANOTHER PAGE

```
<a href="about.html">
  Read about me
</a>
```

Use meaningful link text. Avoid vague labels like "click here".

VISUAL RESULT

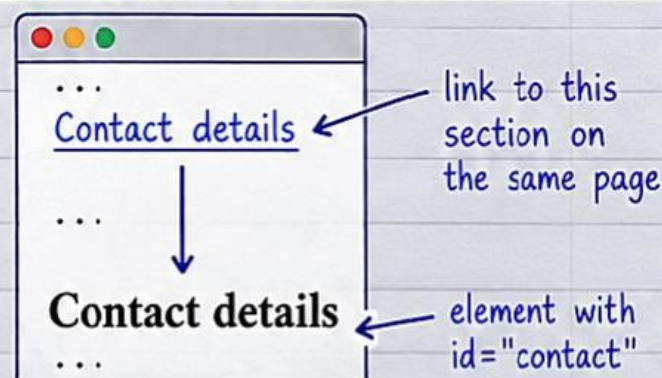


JUMP WITHIN A PAGE

```
<a href="#contact">
  Contact details
</a>
```

```
<section id="contact">
  <h2>Contact details</h2>
</section>
```

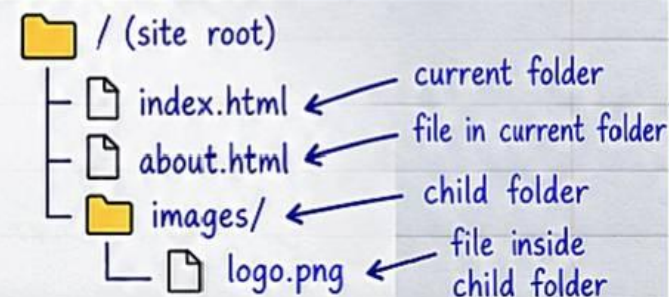
VISUAL RESULT



PATH CHEAT SHEET

about.html → file in the current folder
images/logo.png → child folder
../index.html → parent folder
/help/ → path from the site root
https://example.com/ → full web address

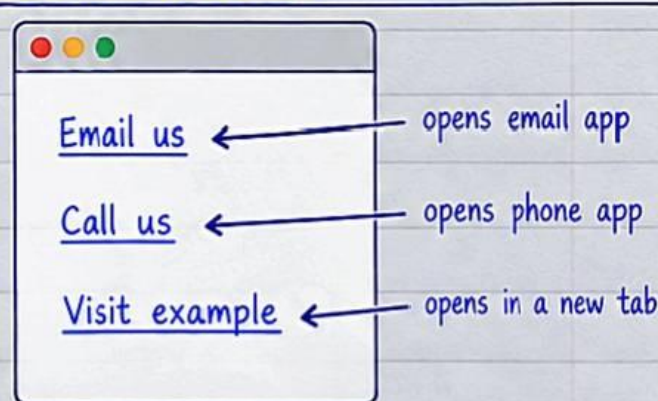
VISUAL RESULT



OTHER LINKS

```
<a href="mailto:hello@example.com">
  Email us
</a>
<a href="tel:+1234567890">
  Call us
</a>
<a href="https://example.com"
  target="_blank" rel="noopener">
  Visit example
</a>
```

VISUAL RESULT



REMEMBER

- Opening a new tab can surprise users; tell them when useful.
- Keep filenames consistent. Many web servers treat uppercase and lowercase as different.

BASIC IMAGE

```

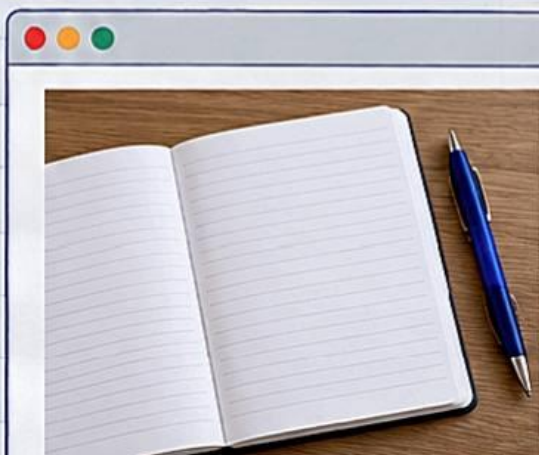
```

src locates the file.

alt communicates the image's purpose.

width and height reserve space and reduce layout movement.

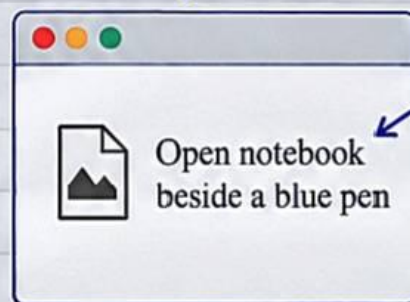
VISUAL RESULT



CHOOSE USEFUL ALT TEXT

- Informative image: describe its relevant information.
- Decorative image: use alt="".
- Linked image: describe the link's purpose.
- Do not rely on a filename as the description.

If the image fails to load



alt is a text alternative, not a visible caption when image loads.

CAPTIONED IMAGE

```
<figure>
  
  <figcaption>Our first week of growth.</figcaption>
</figure>
```

VISUAL RESULT



LOADING

loading="lazy" can defer images below the initial view.

Avoid lazy-loading the main image people see first.

Use your own images or images you have permission to use.

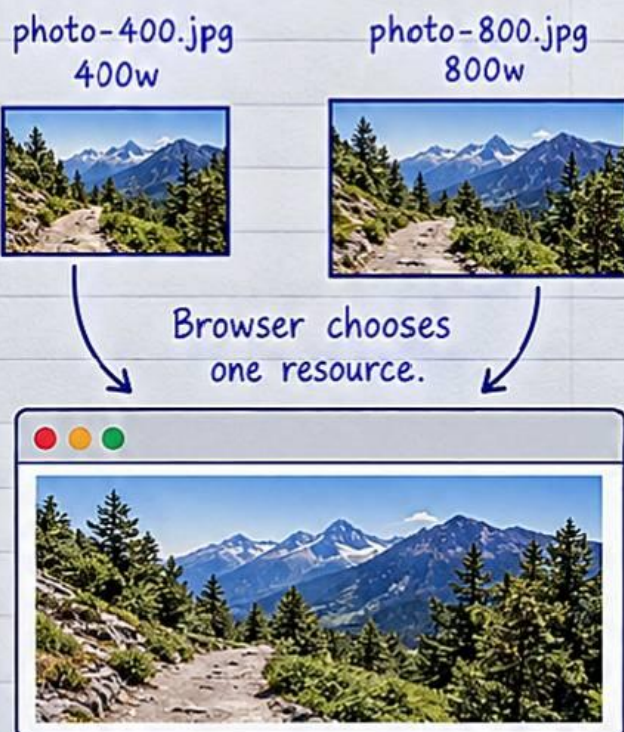
RESOLUTION CHOICES

```

```

400w and 800w describe the files' actual pixel widths.
sizes describes the expected display width.
The browser chooses a suitable resource.
Here, 100vw describes a full-viewport-width image.

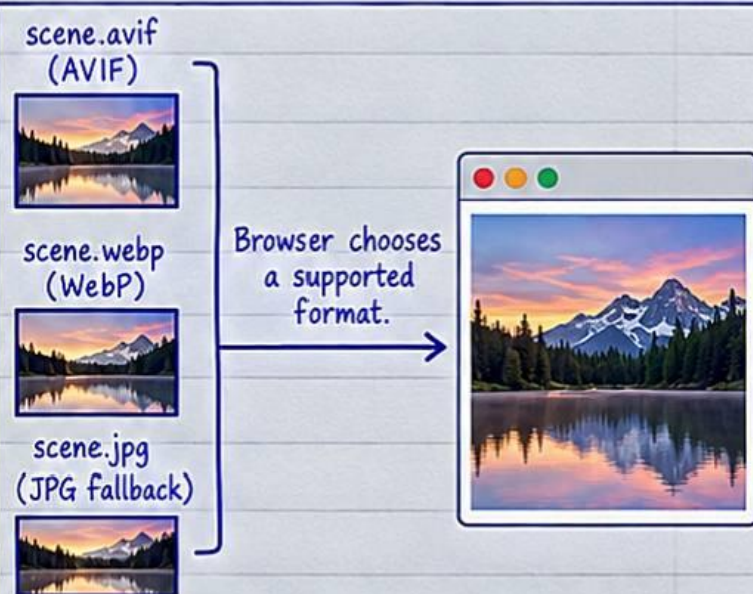
VISUAL RESULT



FORMAT CHOICES

```
<picture>
  <source srcset="scene.avif"
           type="image/avif">
  <source srcset="scene.webp"
           type="image/webp">
  
</picture>
```

VISUAL RESULT



IMPORTANT

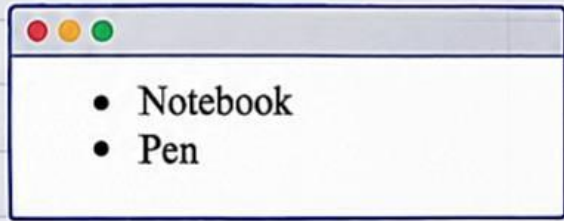
- `<picture>` still needs its `<img>` fallback.
- Put alt on `<img>`.
- Responsive image selection does not, by itself, define the page layout.
- Test that every listed image file exists.

09/20 — LISTS THAT MATCH THE CONTENT

UNORDERED: ORDER DOES NOT MATTER

```
<ul>
  <li>Notebook</li>
  <li>Pen</li>
</ul>
```

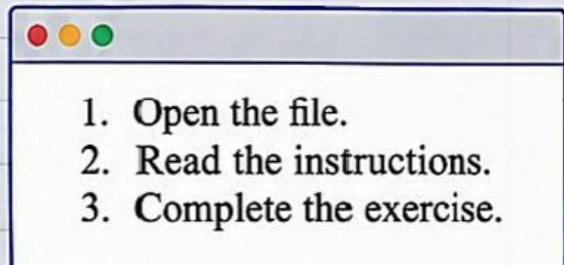
VISUAL RESULT

- 
- A browser window showing an unordered list with two items: 'Notebook' and 'Pen'. Each item is preceded by a black bullet point.
- Notebook
 - Pen

ORDERED: SEQUENCE MATTERS

```
<ol>
  <li>Open the file.</li>
  <li>Read the instructions.</li>
  <li>Complete the exercise.</li>
</ol>
```

VISUAL RESULT

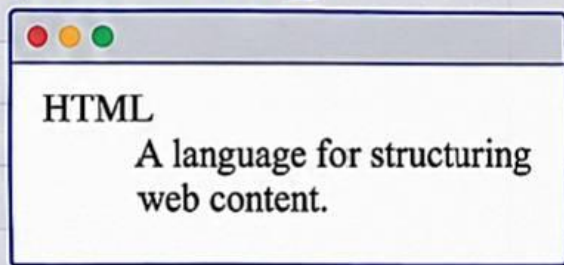
- 
- A browser window showing an ordered list with three items: 'Open the file.', 'Read the instructions.', and 'Complete the exercise.'. Each item is preceded by a black number.
1. Open the file.
 2. Read the instructions.
 3. Complete the exercise.

Use start="3" to begin at three, or reversed to count down.

DESCRIPTION LIST

```
<dl>
  <dt>HTML</dt>
  <dd>A language for structuring
    web content.</dd>
</dl>
```

VISUAL RESULT

A browser window showing a description list. The term 'HTML' is bolded, and its description 'A language for structuring web content.' is indented below it.

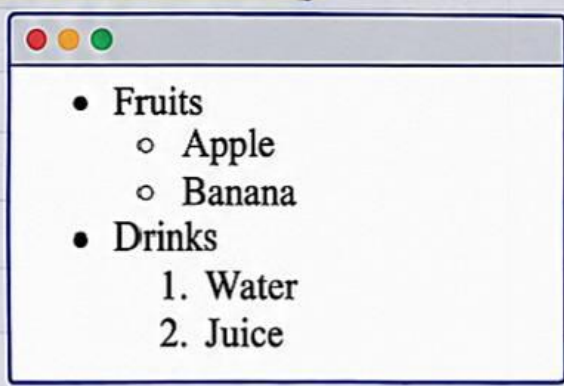
HTML
A language for structuring web content.

<dt> is the term; <dd> is its description.

NESTED LISTS

```
<ul>
  <li>Fruits
    <ul>
      <li>Apple</li>
      <li>Banana</li>
    </ul>
  </li>
  <li>Drinks
    <ol>
      <li>Water</li>
      <li>Juice</li>
    </ol>
  </li>
</ul>
```

VISUAL RESULT

- 
- A browser window showing a nested list. 'Fruits' is a bullet point with a nested unordered list of 'Apple' and 'Banana'. 'Drinks' is a bullet point with a nested ordered list of '1. Water' and '2. Juice'.
- Fruits
 - Apple
 - Banana
 - Drinks
 1. Water
 2. Juice

Place a nested or inside an .
Do not place paragraph text directly between list items.
Use real lists instead of typing bullet symbols by hand.

PRACTICE

Write a packing list and a numbered recipe.

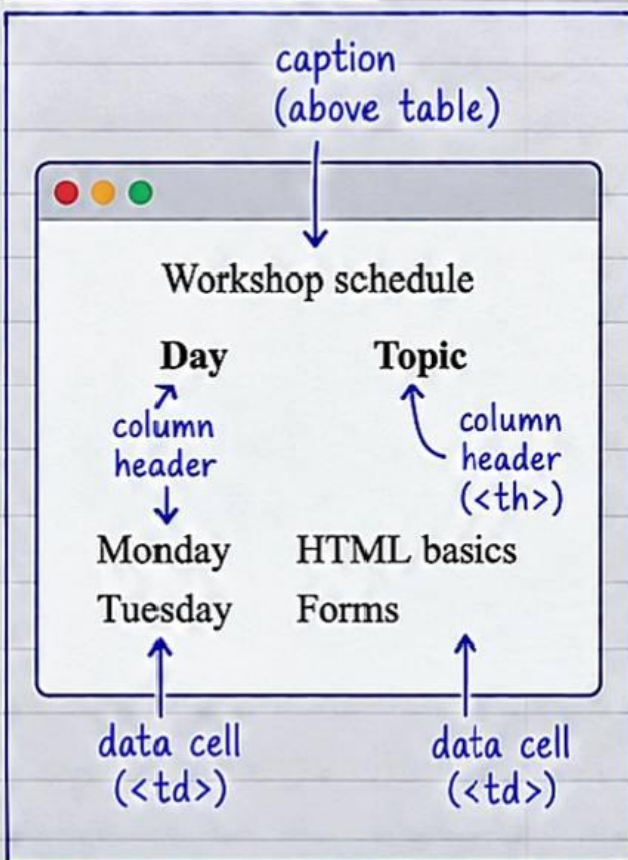
A SMALL DATA TABLE

```

<table>
  <caption>Workshop schedule </caption>
  <thead>
    <tr>
      <th scope="col">Day </th>
      <th scope="col">Topic </th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Monday </td> <td>HTML basics </td>
    </tr>
    <tr>
      <td>Tuesday </td> <td>Forms </td>
    </tr>
  </tbody>
</table>

```

VISUAL RESULT



KEY ELEMENTS

<table> : the complete table
 <tr> : one row
 <th> : a header cell
 <td> : a data cell
 <caption> : a table description
 <thead>, <tbody>, <tfoot> : row groups

GOOD HABITS

- Use tables for related data, not page layout.
- Use scope="row" for a row header.
- colspan and rowspan merge cells; keep complex tables understandable.
- Avoid obsolete presentational attributes such as cellpadding and bgcolor.

11 / 20 — HTML: SEMANTIC PAGE STRUCTURE

EXAMPLE

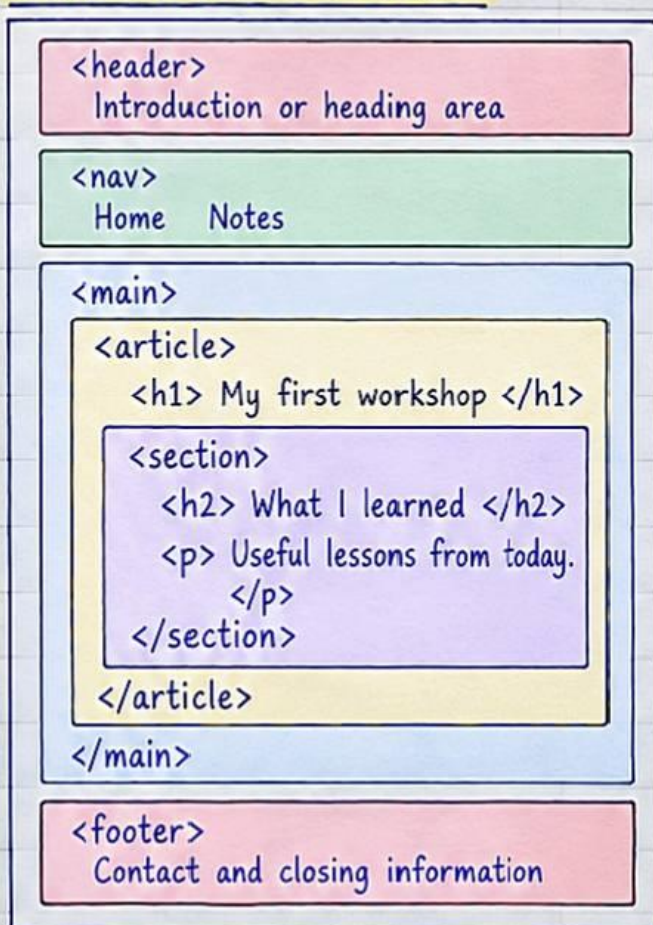
```
<header>
  Introduction or heading area
</header>

<nav aria-label="Main">
  <a href="index.html">Home</a>
  <a href="notes.html">Notes</a>
</nav>

<main>
  <article>
    <h1>My first workshop</h1>
    <section>
      <h2>What I learned</h2>
      <p>Useful lessons from today.</p>
    </section>
  </article>
</main>

<footer>Contact and closing information</footer>
```

STRUCTURE DIAGRAM



WHEN TO USE EACH

<main>: the page's dominant content.

<article>: a self-contained piece of content.

<section>: a thematic group, usually with a heading.

<aside>: related, supporting content.

<nav>: a major group of navigation links.

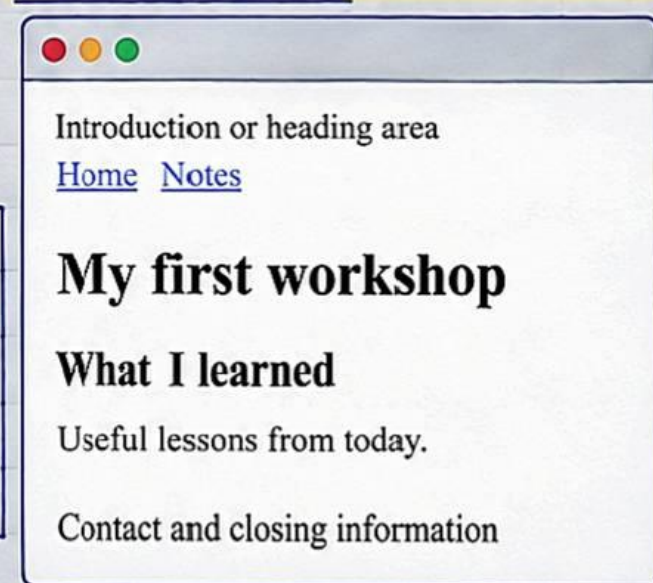
NEUTRAL CONTAINERS

<div> and **** do not add special meaning.

Use them when no meaningful element fits.

Do not add semantic tags only for appearance.

BROWSER PREVIEW (DEFAULT STYLING)

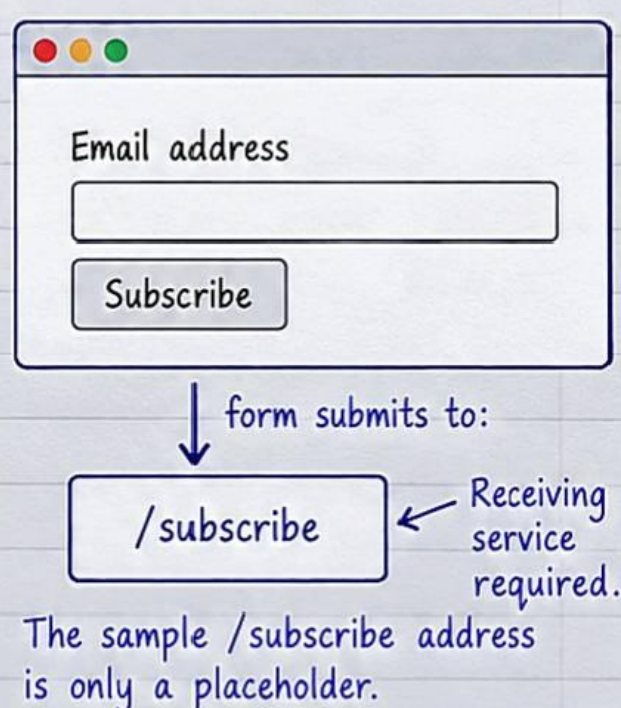


12/20 — FORMS: THE BASICS

A FORM COLLECTS VALUES

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

VISUAL RESULT



UNDERSTAND THE ATTRIBUTES

action: where the browser sends the form.
method: how it sends the values.
name: the key used in submitted data.
id: connects a control to its label.
required: requests a value before normal submission.

GET OR POST?

GET places values in the URL query: useful for searches.
POST sends values in the request body.
POST alone does not encrypt data; use HTTPS.

IMPORTANT

HTML can display and submit a form.
Saving entries or sending emails needs a working receiving service.
The sample /subscribe address is only a placeholder.

13 / 20 — INPUT TYPES & LABELS

CHOOSE A SUITABLE TYPE

text: a short text value
email: an email address
password: visually masked entry
number: a numeric value
date: a date picker
tel: a telephone number
url: a web address
file: a file selection
range: a value on a scale
color: a colour selection

VISUAL RESULT

Text input (type="text")

Full name

← default text field

Number input (type="number")

Number of guests

← default number field (with spinners)

Date input (type="date")



← date picker

Password input (type="password")

← masked entry

Colour input (type="color")

← colour picker

CLEAR LABELS

```
<label for="full-name">Full name</label>  
<input id="full-name" name="fullName"  
      type="text" autocomplete="name">
```

VISUAL RESULT

Full name

← label linked to input

NUMERIC EXAMPLE

```
<label for="guests">Number of guests</label>  
<input id="guests" name="guests"  
      type="number" min="1" max="8" step="1">
```

VISUAL RESULT

Number of guests

← native number input

REMEMBER

- A placeholder is a hint, not a replacement for a label.
- Use type="tel" for phone numbers: they can contain + and leading zeros.
- A masked password field does not secure transmission.
- Note: the appearance of form controls can vary between browsers.
- Do not rely on custom styling — the examples show default HTML controls.

GROUP RELATED CHOICES

```
<fieldset>
  <legend>Study mode</legend>
  <label><input type="radio" name="mode"
    value="online"> Online</label>
  <label><input type="radio" name="mode"
    value="campus"> On campus</label>
</fieldset>
```

Radios with the same name form one choice group.
Checkboxes allow independent selections.

VISUAL RESULT

Study mode

☐ Online

☐ On campus

A SELECT MENU

```
<label for="level">Level</label>
<select id="level" name="level">
  <option value="beginner">Beginner</option>
  <option value="intermediate">Intermediate</option>
</select>
```

Level Beginner ▼

VISUAL RESULT

LONGER TEXT

```
<label for="message">Message</label>
<textarea id="message" name="message"
  rows="4"></textarea>
```

Message

VISUAL RESULT

BUTTONS

```
<button type="submit">Send</button>
<button type="button">Do something</button>
<button type="reset">Reset</button>
```

type="submit" sends the form.

type="button" does not submit it.

Avoid an unexpected reset button: it clears user work.

VISUAL RESULT

Send

Do something

Reset

BROWSER VALIDATION

- required: a value is needed.
- minlength / maxlength: text length limits.
- min / max / step: numeric constraints.
- pattern: a pattern the whole value must match.

EXAMPLE

```
<label for="code">Three-letter code</label>
<input id="code" name="code"
      pattern="[A-Za-z]{3}" required
      aria-describedby="code-help">
<p id="code-help">Enter exactly three
  letters.</p>
```

VISUAL RESULT

Three-letter code

AB

Enter exactly three letters.

Invalid:
needs three
letters.

Three-letter code

ABC

Enter exactly three letters.

Valid format.

UPLOAD FORM

```
<form action="/upload" method="post"
      enctype="multipart/form-data">
  <label for="photo">Choose a photo</label>
  <input id="photo" name="photo"
        type="file" accept="image/*">
  <button type="submit">Upload</button>
</form>
```

VISUAL RESULT

Choose a photo

Choose File

No file chosen

Upload

IMPORTANT LIMITS

- accept is a picker hint, not a security check.
- Browser validation can be bypassed; the receiving server must validate inputs and files.
- Disabled controls are normally not submitted.
- Readonly text controls are normally submitted.

AUDIO PLAYER

```
<audio controls>  
  <source src="lesson.mp3"  
    type="audio/mpeg">  
  Your browser cannot play  
    this audio.  
</audio>
```

VISUAL RESULT

Audio player (controls vary by browser)



↑
Native audio controls
(play, timeline and volume)

VIDEO PLAYER

```
<video controls width="640"  
  poster="cover.jpg">  
  <source src="lesson.mp4"  
    type="video/mp4">  
  <track src="captions.vtt"  
    kind="captions"  
    srclang="en" label="English"  
    default>  
  Your browser cannot play  
    this video.  
</video>
```

VISUAL RESULT

Video player (controls vary by browser)



↑
Native video controls
(play, timeline, volume, captions, fullscreen)

USEFUL ATTRIBUTES

controls:	built-in playback controls
loop:	repeat playback
muted:	begin with sound muted
preload:	a loading hint, not a guarantee
poster:	an image before video playback

ACCESSIBLE MEDIA

- Provide accurate captions and a transcript.
- Caption files commonly use WebVTT.
- Avoid unexpected autoplay. Browsers may block it.
- Different browsers support different media formats; test your files.

17/20 — IFRAME & BUILT-IN DISCLOSURE

EMBED ANOTHER PAGE

```
<iframe src="https://example.com"
        title="Example reference page"
        width="600" height="350"
        loading="lazy"></iframe>
```

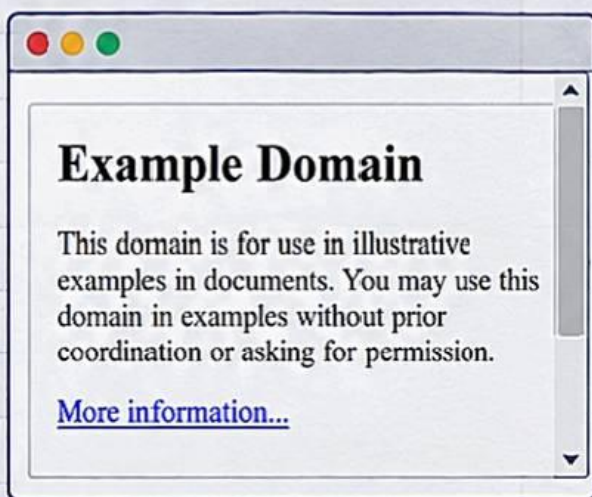
The title describes the frame to assistive technology.

Only embed sources you trust.

Some websites block embedding.

sandbox can restrict iframe capabilities, but its permissions need careful selection.

VISUAL RESULT



↑
Embedded page (if permitted)

EXPANDABLE CONTENT

```
<details>
  <summary>What should I bring?</summary>
  <p>Bring a notebook and your laptop.</p>
</details>
```

The browser supplies the open/close behaviour.

Add the open attribute to start expanded.

VISUAL RESULT

Closed (default)

▶ What should I bring?

Opened

▼ What should I bring?

Bring a notebook and your laptop.

PROGRESS & MEASUREMENT

```
<progress value="3" max="5">3 of 5</progress>
<meter min="0" max="100" value="65">65%</meter>
```

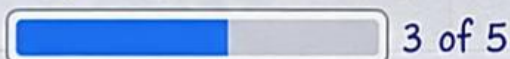
Use progress for task completion.

Use meter for a measurement within a known range.

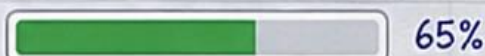
Give each control a meaningful label.

VISUAL RESULT

Progress (3 of 5)



Meter (65%)



18 / 20 — METADATA & SPECIAL TEXT

INSIDE THE HEAD

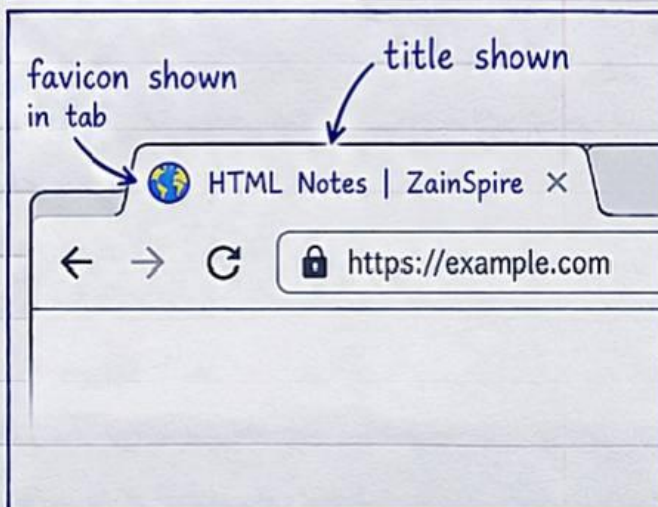
```
<title>HTML Notes | ZainSpire</title>
<meta name="description"
      content="Practical HTML notes
              for beginners.">
<link rel="icon" href="favicon.png">
```

A clear title identifies the page.

A description summarises it; search engines may use a different snippet.

Metadata cannot guarantee a search ranking.

VISUAL RESULT



CHARACTER REFERENCES

`<`; displays `<`
`>`; displays `>`
`&`; displays `&`
`"`; displays a double quote
` `; is a non-breaking space, not a layout tool.

VISUAL RESULT

`<` → `<`
`>` → `>`
`&` → `&`
`"` → `"`
` ` → (a space)

CODE & QUOTATIONS

```
<pre><code>&lt;h1&gt;Hello&lt;/h1&gt;</code></pre>
<code> marks code; <pre> preserves whitespace.
<blockquote> marks an extended quotation.
<cite> identifies the title of a referenced work.
```

VISUAL RESULT

```
<h1>Hello</h1>
```

shown as code text
(angle brackets visible)

DATE & ABBREVIATION

```
<time datetime="2026-10-29">
  29 October 2026</time>
<abbr title="HyperText Markup Language">
  HTML</abbr>
```

VISUAL RESULT

29 October 2026 ← date shown as normal text
HTML ← abbreviation shown with its title/meaning

20 / 20 — HTML-ONLY MINI PROJECT

BUILD A WORKSHOP PAGE

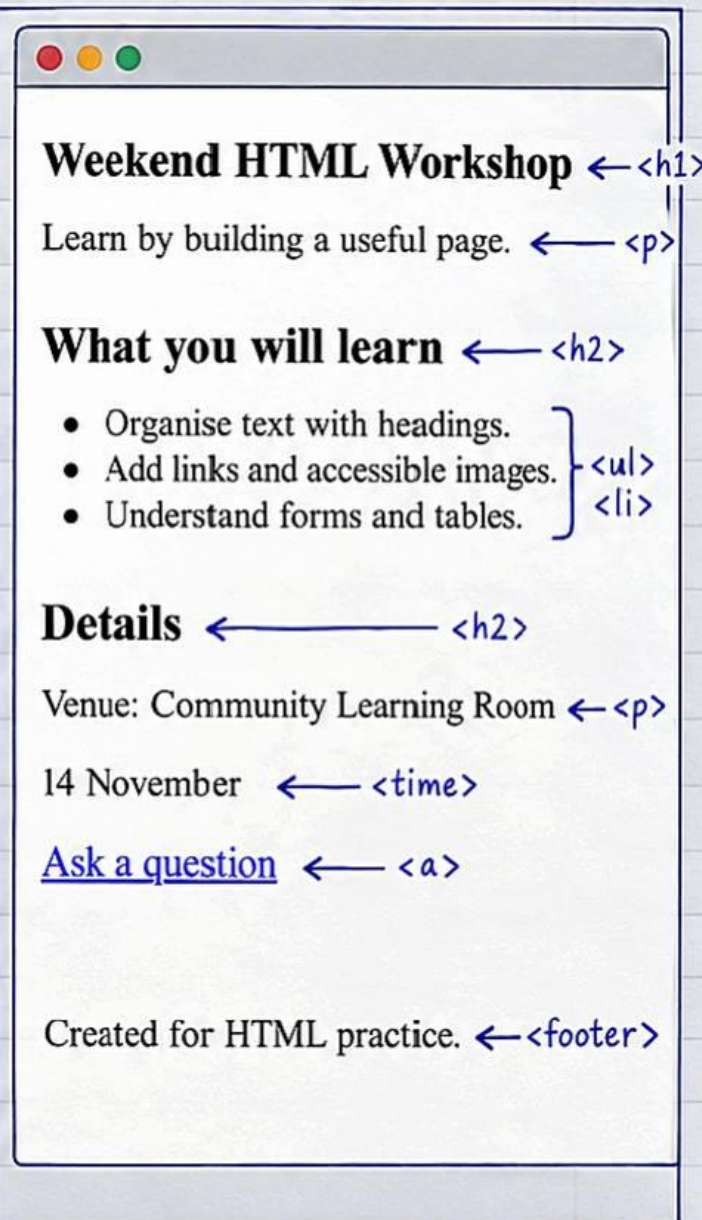
Create index.html with the document skeleton from page 2.

Put this example inside `<body>`:

EXAMPLE (index.html)

```
<header>
  <h1>Weekend HTML Workshop</h1>
  <p>Learn by building a useful page.</p>
</header>
<main>
  <section>
    <h2>What you will learn</h2>
    <ul>
      <li>Organise text with headings.</li>
      <li>Add links and accessible images.</li>
      <li>Understand forms and tables.</li>
    </ul>
  </section>
  <section>
    <h2>Details</h2>
    <p>Venue: Community Learning Room</p>
    <p><time datetime="2026-11-14">
      14 November</time></p>
    <a href="mailto:hello@example.com">
      Ask a question</a>
  </section>
</main>
<footer><p>Created for HTML practice.</p></footer>
```

VISUAL RESULT (in browser)



YOUR CHALLENGE

Add a schedule table, one captioned image and a details/summary FAQ.

Use your own text and files.

Replace the example email before publishing.

Check every link and label.

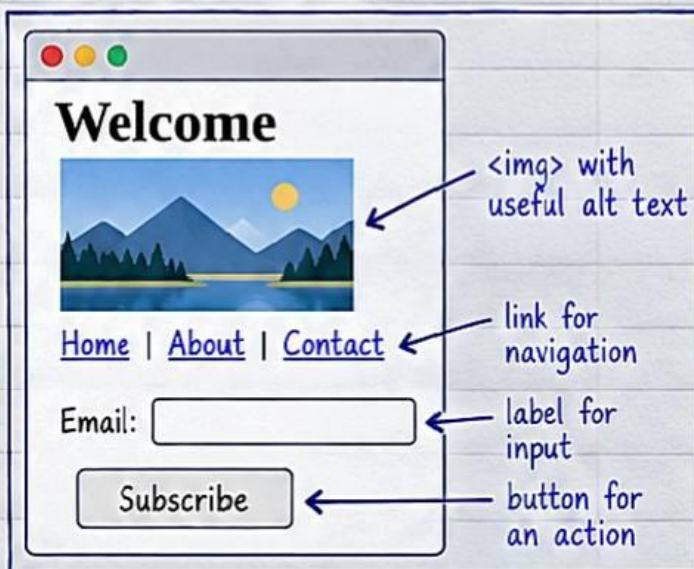
No CSS or JavaScript is needed.

19/20 — ACCESSIBILITY & FINAL CHECKS

BUILD FOR PEOPLE

- Use headings in a logical order.
- Give informative images useful alt text.
- Give every form control a clear label.
- Use a link for navigation and a button for an action.
- Prefer native HTML controls before adding ARIA.
- Do not rely only on a tooltip to explain something.

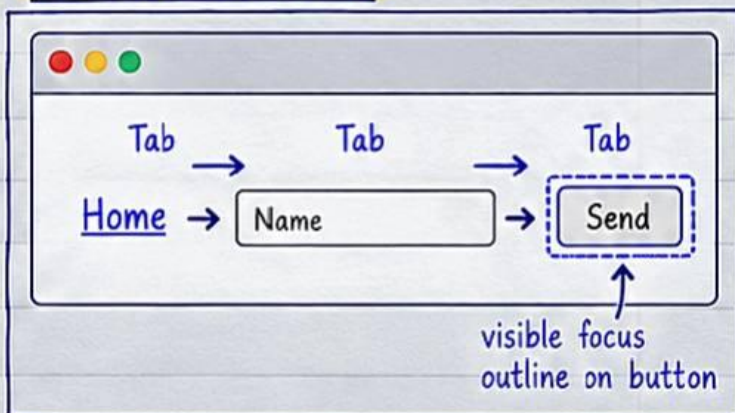
VISUAL RESULT



KEYBOARD CHECK

- Press Tab to move through links and controls.
- Try buttons with Enter or Space.
- Make sure the content order is understandable.
- Avoid positive tabindex values that reorder navigation.

VISUAL RESULT



BEFORE SHARING

- Check that links and file paths work.
- Check unique IDs and correct nesting.
- Use a descriptive title and correct lang value.
- Check on a narrow screen.
- Check that forms have a real destination if submission is needed.
- Validate markup with the Nu HTML Checker.

VISUAL RESULT

- ✓ Use a descriptive title and lang
- ✓ Unique IDs and correct nesting
- ✓ Links and file paths work
- ✓ Works on a narrow screen

COMMON MISTAKES

Missing alt; labels not connected to inputs; duplicate IDs; using tables for layout; placing secrets in comments.

VISUAL RESULT

