

HTML5

From Beginner to Advanced

A Complete Tag-by-Tag Guide with Hands-On Exercises

Prepared by
zolaxtech.com.et

Written and practiced entirely in Visual Studio Code
August 2026

How to Use This Guide?

This guide takes you from a completely empty folder to a full understanding of every major HTML5 tag and attribute in use today, using Visual Studio Code as your editor throughout. Every tag is presented the same way, so you always know what to expect:

- **The tag itself:** shown in a monospace font, e.g. `<p>`.
- **What it does:** a plain-language explanation of its purpose.
- **Attributes:** a table of the tag's most important attributes and exactly what each one does.
- **Example:** a real, working code snippet you can type directly into VS Code.
- **Exercise:** a short hands-on task, using zolaxtech.com.et as the running example, so you practice each tag immediately instead of just reading about it.

Each chapter ends with a larger "Chapter Project" exercise that combines everything covered in that chapter into one realistic piece of the Zolax Tech website. By the final chapter, you will have assembled the working pieces of a real multi-page business website built, tested, and previewed entirely inside Visual Studio Code.

Table of Contents

How to Use This Guide.....	2
Table of Contents.....	3
Chapter 1: Getting Started with HTML in Visual Studio Code.....	7
Step 1: Install VS Code.....	7
Step 2: Install the Live Server extension	7
Step 3: Install the HTML CSS Support and Auto Rename Tag extensions	7
Step 4: Create your project folder	7
Step 5: Create your first file	7
Step 6: Use the Emmet shortcut.....	7
Step 7: Preview your page	7
Chapter Summary Project	7
Chapter 2: HTML Document Structure	8
<!DOCTYPE html>	8
<html>.....	8
<head>	9
<meta>	9
<title>	10
<link>.....	10
<style>	10
<script>	11
<body>	11
Chapter Summary Project	12
Chapter 3: Headings, Paragraphs & Basic Text Tags.....	12
<h1> to <h6>	12
<p>	12
	13
<hr>.....	13
<div>.....	13
.....	14
Chapter Summary Project	14
Chapter 4: Text Formatting & Inline Semantic Tags.....	14
	14
	15
	15
<i>	15
<mark>.....	15

<small>	16
	16
<ins>	16
<sub>	17
<sup>	17
<blockquote>	17
<q>	18
<abbr>	18
<address>	18
<cite>	19
<code>	19
<pre>	19
Chapter Summary Project	20
Chapter 5: Links & Navigation	20
<a>	20
Chapter Summary Project	21
Chapter 6: Images & Media	21
	21
<figure> & <figcaption>	22
<picture> & <source>	22
<audio>	22
<video>	23
<track>	24
Chapter Summary Project	24
Chapter 7: Lists	24
.....	24
.....	25
.....	25
<dl>, <dt>, <dd>	26
Chapter Summary Project	26
Chapter 8: Tables	26
<table>	26
<caption>	27
<thead>	27
<tbody>	27
<tfoot>	27
<tr>	28

<th>	28
<td>	28
<colgroup> & <col>	29
Chapter Summary Project	29
Chapter 9: Forms & User Input	30
<form>	30
<label>	30
<input>	31
<textarea>	32
<select>, <option>, <optgroup>	33
<button>	33
<fieldset> & <legend>	34
<datalist>	34
<output>	35
<progress>	35
<meter>	36
Chapter Summary Project	36
Chapter 10: Semantic HTML5 Layout Tags	36
<header>	36
<nav>	37
<main>	37
<section>	37
<article>	38
<aside>	38
<footer>	38
<details> & <summary>	39
<dialog>	39
<time>	40
Chapter Summary Project	40
Chapter 11: Embedded Content & Graphics	40
<iframe>	40
<embed>	41
<object>	41
<canvas>	42
<svg>	42
Chapter Summary Project	43
Chapter 12: Global Attributes & Data Attributes	43

id	43
class.....	43
style	44
title	44
data-*	44
tabindex.....	45
contenteditable	45
hidden.....	45
draggable.....	46
lang (on any element).....	46
Chapter Summary Project	46
Chapter 13: Advanced HTML5 SEO, Accessibility & Best Practices	47
Meta tags for SEO.....	47
ARIA attributes (accessibility)	47
Favicon setup	48
Character entities	48
Form validation attributes (recap & advanced)	49
Chapter Summary Project	50
Appendix A: Useful VS Code Shortcuts for HTML	50
Appendix B: Final Capstone Project.....	50

Chapter 1: Getting Started with HTML in Visual Studio Code

Before writing a single tag, you need a workspace. Visual Studio Code (VS Code) is the most popular free code editor for web development, made by Microsoft. This chapter walks you through installing it, setting it up for HTML, and writing your first page for `zolaxtech.com.et`.

Step 1: Install VS Code

Go to `code.visualstudio.com`, download the installer for your operating system (Windows, macOS, or Linux), and run it. Accept the default options during installation.

Step 2: Install the Live Server extension

Open VS Code, click the Extensions icon on the left sidebar (or press `Ctrl+Shift+X`), search for "Live Server" by Ritwick Dey, and click Install. This lets you preview your HTML in a browser that auto-refreshes every time you save.

Step 3: Install the HTML CSS Support and Auto Rename Tag extensions

Search for and install "Auto Rename Tag" (renames the closing tag automatically when you edit an opening tag) and "HTML CSS Support" for autocomplete. These are optional but save time.

Step 4: Create your project folder

On your computer, create a folder named `zolaxtech-site`. In VS Code, go to `File > Open Folder` and select it. This folder will hold every file for your website.

Step 5: Create your first file

In VS Code's Explorer panel, click the New File icon and name it `index.html`. The name `index.html` is important it is the default file a web server looks for when someone visits your domain.

Step 6: Use the Emmet shortcut

Inside `index.html`, type an exclamation mark `!` and press `Tab` (or `Enter`). VS Code will auto-generate a complete HTML5 boilerplate for you using a built-in tool called Emmet.

Step 7: Preview your page

Right-click anywhere inside `index.html` and choose "Open with Live Server", or click "Go Live" in the bottom-right status bar. Your default browser opens and shows the page. Every time you save (`Ctrl+S`), the browser refreshes automatically.

Chapter Summary Project

Chapter Project

Install VS Code and the Live Server extension. Create a folder called `zolaxtech-site`, open it in VS Code, create `index.html`, type `!` and press `Tab` to generate the boilerplate, then open it with Live Server. Take note of what appears in the browser tab and in the page body.

Chapter 2: HTML Document Structure

Every HTML5 page follows the same skeleton. This chapter explains each structural tag that must appear in (almost) every page you build, including the pages on zolaxtech.com.et.

<!DOCTYPE html>

This is not a tag in the strict sense it is a declaration placed at the very top of the file that tells the browser which version of HTML the document uses. Writing <!DOCTYPE html> tells the browser to use the modern HTML5 rendering rules (called "standards mode") instead of an old, quirky rendering mode from the 1990s.

Example

```
<!DOCTYPE html>
<html>
  <!-- rest of the page -->
</html>
```

Exercise

Create a new file called `doctype-test.html`. Leave out the <!DOCTYPE html> line entirely and add some text. Then add it back. There is no visible difference for simple pages, but note that professionals always include it write one sentence in a comment explaining why you think that matters for complex pages.

<html>

The root element that wraps every other element on the page. Everything visible or invisible belongs inside <html>...</html>.

Attributes

Attribute	What it does
lang	Declares the human language of the page, e.g. <code>lang="en"</code> for English or <code>lang="am"</code> for Amharic. This helps screen readers pronounce words correctly and helps search engines serve the right audience.
dir	Sets text direction: <code>"ltr"</code> (left-to-right, the default) or <code>"rtl"</code> (right-to-left, used for languages like Arabic or Hebrew).

Example

```
<html lang="en">
  <head>...</head>
  <body>...</body>
</html>
```

Exercise

Open index.html and add lang="en" to your <html> tag. Change it temporarily to lang="am" and lang="fr" and notice that nothing visually changes in the browser the attribute is metadata for machines, not humans.

<head>

A container for information ABOUT the page that is not displayed directly on the page itself: the title, character encoding, linked stylesheets, scripts, and search-engine metadata.

Example

```
<head>
<meta charset="UTF-8">
<title>Zolax Tech</title>
</head>
```

Exercise

In your <head>, add a comment (<!-- like this -->) listing three things you plan to put inside it for the Zolax Tech homepage.

<meta>

A self-closing tag that provides metadata such as character encoding, page description, author, and how the page should behave on mobile devices. A single page usually contains several <meta> tags.

Attributes

Attribute	What it does
charset	Sets the character encoding. Almost always "UTF-8", which supports virtually every character and language in the world, including Amharic script.
name + content	Used together, e.g. name="description" content="Zolax Tech builds websites in Ethiopia." this text often appears under your link in Google search results.
name="viewport"	content="width=device-width, initial-scale=1.0" tells mobile browsers to render the page at the phone's actual screen width instead of zooming out essential for responsive design.

Example

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="description" content="Zolax Tech - web development in Ethiopia">
<meta name="author" content="Zolax Tech">
```

Exercise

Add all four meta tags shown in the example to your index.html <head>. Change the description to something specific about zolaxtech.com.et and view the page source in the browser (right-click > View Page Source) to confirm it appears.

<title>

Sets the text shown in the browser tab and in search engine results as the clickable headline. Every page should have exactly one, and it should be unique and descriptive.

Example

```
<title>Zolax Tech - Web Development in Ethiopia</title>
```

Exercise

Set the title of your homepage to something for Zolax Tech. Save and check the browser tab text. Then open a second file, about.html, and give it a DIFFERENT, more specific title such as "About Us | Zolax Tech".

<link>

A self-closing tag used mainly to connect an external CSS stylesheet to the HTML page, though it is also used for favicons and fonts.

Attributes

Attribute	What it does
rel	The relationship between the current page and the linked file, e.g. "stylesheet" or "icon".
href	The path or URL to the linked file.
type	The MIME type of the linked file, e.g. "text/css" (often omitted in HTML5 since it's inferred).

Example

```
<link rel="stylesheet" href="styles.css">  
<link rel="icon" href="favicon.ico" type="image/x-icon">
```

Exercise

Create an empty file named styles.css in the same folder. Link it in your <head> using <link rel="stylesheet" href="styles.css">. Add body { background-color: lightyellow; } inside styles.css and refresh the browser to confirm the link works.

<style>

Let's you write CSS directly inside the HTML document instead of (or in addition to) an external file. Useful for quick tests, but external stylesheets are preferred for real sites.

Example

```
<style>
  body {font-family: Arial, sans-serif;}
  h1 { color: darkblue; }
</style>
```

✎ Exercise

Add a `<style>` block inside your `<head>` that makes all `<h1>` text a color of your choice.

`<script>`

Embeds or links JavaScript, which adds interactivity to a page (things HTML and CSS alone cannot do, like responding to a button click).

Attributes

Attribute	What it does
src	Path to an external JavaScript file, e.g. <code>src="main.js"</code> , instead of writing JS inline.
defer	Tells the browser to download the script in the background but wait to run it until the HTML has fully loaded recommended for scripts placed in <code><head></code> .
async	Downloads and runs the script as soon as it's ready, without waiting for the rest of the page used for independent scripts like ads or analytics.

Example

```
<script src="main.js" defer></script>
<script>
  console.log ("Zolax Tech site loaded");
</script>
```

✎ Exercise

Add an inline `<script>` just before the closing `</body>` tag that runs `alert("Welcome to Zolax Tech");`
Save and reload the page to see the popup.

`<body>`

Contains everything that is actually visible to the visitor: headings, paragraphs, images, links, forms all of it. A document has exactly one `<body>`.

Example

```
<body>
  <h1>Welcome to Zolax Tech</h1>
  <p>We build modern websites. </p>
</body>
```

Exercise

Inside `<body>`, add one heading and one paragraph introducing Zolax Tech in your own words.

Chapter Summary Project

Chapter Project

Build a complete, valid HTML5 skeleton for `zolaxtech.com.et` from scratch (without using the Emmet shortcut this time) including: doctype, html with lang, head with charset/viewport/title/description, a linked external stylesheet, and a body with a heading and paragraph. Validate it by pasting the code into validator.w3.org/nu and fixing any errors reported.

Chapter 3: Headings, Paragraphs & Basic Text Tags

This chapter covers the tags you'll use most often: organizing text into headings and paragraphs, and controlling line breaks.

`<h1>` to `<h6>`

Six levels of section headings, `<h1>` being the most important (usually your page's main title, used once per page) down to `<h6>`, the least important. Headings create the document's outline, which matters for accessibility and SEO never choose a heading level just because of its default font size.

Example

```
<h1>Zolax Tech</h1>
<h2>Our Services</h2>
<h3>Web Development</h3>
```

Exercise

Write a heading outline for a Zolax Tech services page: one `<h1>` for the page title, three `<h2>` tags for three services you offer, and one `<h3>` nested under one of them for a specific package.

`<p>`

Defines a paragraph of text. Browsers automatically add space above and below each paragraph. Never use empty paragraphs just to create blank space use CSS margin instead.

Example

```
<p>Zolax Tech is a software company based in Ethiopia, specializing in web and mobile development.</p>
```

Exercise

Write three paragraphs describing Zolax Tech's mission, the services offered, and how to get in touch one paragraph per idea.

A self-closing tag that inserts a single line break, forcing the next content onto a new line without starting a new paragraph. Use sparingly for addresses or poems, not for general spacing (that's what CSS is for).

Example

```
<p>Zolax Tech<br>
Addis Ababa, Ethiopia<br>
Email: info@zolaxtech.com.et</p>
```

Exercise

Write a short mailing address for Zolax Tech inside one <p>, using
 to put the street, city, and email each on their own line.

<hr>

A self-closing tag that draws a horizontal line, used to signal a thematic break between sections of content (not just decoration).

Example

```
<p>End of our services section. </p>
<hr>
<p>Start of our contact section.</p>
```

Exercise

Add an <hr> between two different sections of your homepage to visually separate them.

<div>

A generic block-level container with no inherent meaning used purely to group other elements together so you can style or position them as a unit with CSS. Overusing <div> when a more meaningful semantic tag exists (covered in Chapter 9) is considered poor practice.

Example

```
<div class="card">
```

```
<h3>Web Development</h3>
<p>Custom websites built for your business. </p>
</div>
```

Exercise

Wrap one of your service descriptions in a `<div>` with a class attribute of your choosing, e.g. `class="service-card"`.

`<span>`

The inline equivalent of `<div>` a generic container with no meaning of its own, used to style or target a small piece of text inside a larger block, such as a single word.

Example

```
<p>Zolax Tech is <span style="color:green;">now hiring</span> junior developers.</p>
```

Exercise

In one of your paragraphs, wrap a single important word or phrase in a `<span>` and give it a style attribute that changes its color.

Chapter Summary Project

Chapter Project

Build a short 'About' section for `zolaxtech.com.et` using: one `<h2>` section heading, two `<p>` paragraphs, an `<hr>` separating them from a 'Contact' section below, and a `<div>` that groups a mailing address written with `<br>` line breaks.

Chapter 4: Text Formatting & Inline Semantic Tags

HTML provides many inline tags for emphasizing, marking up, or giving special meaning to small pieces of text. This chapter covers the full set.

`<strong>`

Marks text as having strong importance, seriousness, or urgency. Browsers display it bold by default, but unlike `<b>` screen readers announce it with added emphasis because it carries real semantic meaning.

Example

```
<p><strong>Warning:</strong> submit the form before midnight.</p>
```

Exercise

Write a sentence about a Zolax Tech deadline or offer, marking the most urgent word or phrase with `<strong>`.

`<b>`

Makes text bold purely for stylistic/visual reasons, with no extra importance implied for example, bolding a product name or keyword the first time it's used in an article.

Example

```
<p>Our flagship product is <b>ZolaxCRM</b>.</p>
```

Exercise

Bold the name of one Zolax Tech product or service using `<b>`, without implying it is more 'important' than the rest of the text.

`<em>`

Marks text with stress emphasis the kind of emphasis that changes the meaning of a sentence if read aloud. Displayed in italics by default.

Example

```
<p>We <em>really</em> care about client satisfaction.</p>
```

Exercise

Write a sentence where changing which word carries emphasis changes the sentence's meaning, and mark that word with `<em>`.

`<i>`

Italicizes text for stylistic reasons without implying emphasis commonly used for technical terms, foreign phrases, or names of ships/books.

Example

```
<p>Our company motto is <i>Innovation for Everyone</i>.</p>
```

Exercise

Write a sentence containing a foreign word or a book/product title and italicize it using `<i>`.

`<mark>`

Highlights text for reference purposes, as if marked with a highlighter pen displayed with a yellow background by default. Often used to highlight search terms.

Example

```
<p>Search results for <mark>web development</mark> in Addis Ababa.</p>
```

Exercise

Simulate a search-results snippet: write a sentence and highlight the search term using `<mark>`.

`<small>`

Represents side comments such as fine print, disclaimers, or copyright notices displayed in a smaller font.

Example

```
<p><small>&copy; 2026 Zolax Tech. All rights reserved.</small></p>
```

Exercise

Add a copyright line to the bottom of your page wrapped in `<small>`.

`<del>`

Represents text that has been deleted/removed from a document shown with a strikethrough by default. Commonly used to show price changes.

Attributes

Attribute	What it does
cite	URL explaining why the change was made.
datetime	The date/time of the deletion, in a machine-readable format.

Example

```
<p>Price: <del>2000 ETB</del> <ins>1500 ETB</ins></p>
```

Exercise

Show a 'before' and 'after' price for a Zolax Tech service using `<del>` for the old price.

`<ins>`

Represents text that has been inserted into a document shown underlined by default. Often paired with `<del>` to show edits.

Attributes

Attribute	What it does
cite	URL explaining the insertion.

Attribute	What it does
datetime	Date/time of the insertion.

Example

```
<p>Office hours: <del>9am-5pm</del> <ins>8am-6pm</ins></p>
```

✎ Exercise

Pair `<ins>` with the `<del>` price example above to show the new price.

`<sub>`

Renders text as subscript, dropped below the normal line and in a smaller font used for chemical formulas and footnote markers.

Example

```
<p>Water is H<sub>2</sub>O.</p>
```

✎ Exercise

Write a sentence containing a chemical formula (e.g. CO₂) using `<sub>` for the numbers.

`<sup>`

Renders text as superscript, raised above the normal line used for exponents, ordinal indicators (1st, 2nd), and footnotes.

Example

```
<p>Zolax Tech was founded in the 21<sup>st</sup> century.</p>
```

✎ Exercise

Write a sentence using an ordinal number (1st, 2nd, 3rd) with `<sup>` around the suffix.

`<blockquote>`

Marks a section as an extended quotation from another source, usually indented by browsers by default.

Attributes

Attribute	What it does
cite	URL of the source being quoted.

Example

```
<blockquote cite="https://example.com">
```

```
<p>Zolax Tech transformed our online presence completely. </p>
</blockquote>
```

Exercise

Add a client testimonial to your page wrapped in `<blockquote>`, with a `cite` attribute pointing to a (real or placeholder) source URL.

`<q>`

Marks a short, inline quotation. Browsers automatically add quotation marks around the content.

Attributes

Attribute	What it does
cite	URL of the quote's source.

Example

```
<p>Our client said, <q>Zolax Tech delivered on time.</q></p>
```

Exercise

Write a sentence containing a short quote from an imaginary client using `<q>`.

`<abbr>`

Marks an abbreviation or acronym; the `title` attribute provides the full expansion, shown as a tooltip on hover.

Attributes

Attribute	What it does
title	The full, spelled-out form of the abbreviation.

Example

```
<p><abbr title="HyperText Markup Language">HTML</abbr> is the standard markup language for the web.</p>
```

Exercise

Use `<abbr>` to mark up an abbreviation relevant to Zolax Tech (e.g. CRM, API, SEO) with a `title` attribute giving its full meaning.

`<address>`

Wraps contact information for the nearest `<article>` or `<body>` typically physical address, email, phone number.

Example

```
<address>
  Zolax Tech, Bole Road, Addis Ababa<br>
  Email: <a href="mailto:info@zolaxtech.com.et">info@zolaxtech.com.et</a>
</address>
```

Exercise

Wrap the Zolax Tech contact information (from Chapter 3's exercise) in an `<address>` tag.

`<cite>`

Marks the title of a creative work being referenced a book, article, film, or website name (not the author's name).

Example

```
<p>As explained in <cite>MDN Web Docs</cite>, semantic HTML improves accessibility.</p>
```

Exercise

Write a sentence referencing the title of a book, article, or website using `<cite>`.

`<code>`

Marks a short fragment of computer code, displayed in the browser's default monospace font.

Example

```
<p>Use the <code>&lt;h1&gt;</code> tag for your main heading.</p>
```

Exercise

Write a sentence that mentions an HTML tag name by wrapping it in `<code>`.

`<pre>`

Preserves whitespace and line breaks exactly as written in the HTML source used for displaying blocks of code or ASCII art. Almost always paired with `<code>` for actual code.

Example

```
<pre><code>function greet() {
  console.log("Hello, Zolax Tech!");
}</code></pre>
```

Exercise

Write a 3-line block of pretend code (any language) inside `<pre><code>...</code></pre>` and confirm the line breaks and indentation are preserved when viewed in the browser.

Chapter Summary Project

Chapter Project

Write a short 'Testimonials & Fine Print' section for zolaxtech.com.et that uses at least eight of the tags from this chapter together in a natural way: a client quote (`<blockquote>`), an emphasized word (`<em>` or `<strong>`), a highlighted phrase (`<mark>`), an abbreviation with a tooltip (`<abbr>`), a copyright notice (`<small>`), and a price change shown with `<del>/<ins>`.

Chapter 5: Links & Navigation

Links are what make the web a 'web' this chapter covers the anchor tag in full depth.

`<a>`

The anchor tag creates a hyperlink to another page, another part of the same page, an email address, a phone number, or a file to download.

Attributes

Attribute	What it does
href	The destination URL. Can be absolute (https://zolaxtech.com.et/about.html), relative (about.html), an anchor on the same page (#contact), an email (mailto:info@zolaxtech.com.et), or a phone number (tel:+251900000000).
target	Where to open the link. "_blank" opens it in a new tab; the default "_self" opens it in the same tab.
rel	Describes the relationship to the linked page. When using <code>target="_blank"</code> , add <code>rel="noopener noreferrer"</code> for security, preventing the new page from controlling the original one.
download	Tells the browser to download the linked file instead of navigating to it, e.g. <code></code> .
title	Extra descriptive text shown as a tooltip on hover.

Example

```
<a href="about.html">About Us</a>
<a href="https://zolaxtech.com.et" target="_blank" rel="noopener noreferrer">Visit our site</a>
<a href="mailto:info@zolaxtech.com.et">Email us</a>
<a href="tel:+251900000000">Call us</a>
<a href="#contact">Jump to Contact section</a>
```

Exercise

Create a simple navigation bar for zolaxtech.com.et with five links: Home, About, Services, Portfolio, and Contact, each pointing to a matching .html file (they don't need to exist yet). Add a sixth link that opens your (real or placeholder) Facebook page in a new tab using `target="_blank"` and the correct `rel` attribute. Add a `mailto:` link and a `tel:` link in your contact section.

Chapter Summary Project

Chapter Project

Build a full navigation menu using an unordered list of links (you'll learn `<ul>/<li>` properly in Chapter 6, but you may use them here) with at least 5 internal links, one external link opening safely in a new tab, one `mailto:` link, one `tel:` link, and one same-page anchor link (`#`) that jumps down to a section on the same page. Test every link in the browser.

Chapter 6: Images & Media

This chapter covers embedding images, audio, and video essential for a modern business website like zolaxtech.com.et.

`<img>`

A self-closing tag that embeds an image into the page.

Attributes

Attribute	What it does
src	Path or URL to the image file. Required.
alt	Alternative text describing the image, shown if the image fails to load and read aloud by screen readers. Required for accessibility never leave it out.
width / height	Sets the image's displayed dimensions in pixels. Specifying both helps the browser reserve space before the image loads, preventing layout shift.
loading	<code>loading="lazy"</code> tells the browser to delay loading the image until it's about to scroll into view, improving page speed.

Example

```

```

Exercise

Add a logo image and a team photo to your homepage, each with a meaningful `alt` attribute (not just "image" or the filename) and explicit `width/height`.

<figure> & <figcaption>

<figure> groups an image (or other media) together with a caption; <figcaption> holds the caption text itself. Use this whenever an image needs a visible caption or credit line.

Example

```
<figure>
  
  <figcaption>Our development team in Addis Ababa, 2026. </figcaption>
</figure>
```

Exercise

Wrap your team photo from the previous exercise in a <figure>, and add a <figcaption> describing it.

<picture> & <source>

<picture> lets you serve different image files depending on screen size or format support, wrapping multiple <source> tags plus a fallback . The browser picks the best match.

Attributes

Attribute	What it does
srcset (on <source>)	The image file to use for that condition.
media (on <source>)	A CSS media condition, e.g. (max-width: 600px), under which this source should be used.
type (on <source>)	The MIME type of the image, e.g. image/webp, letting the browser pick a supported format.

Example

```
<picture>
  <source media="(max-width: 600px)" srcset="logo-small.png">
  <source type="image/webp" srcset="logo.webp">
  
</picture>
```

Exercise

Write a <picture> element that shows a smaller logo image on screens under 600px wide and falls back to the regular logo otherwise.

<audio>

Embeds a sound file with browser-native playback controls.

Attributes

Attribute	What it does
controls	A boolean attribute that displays play/pause/volume controls. Without it, the audio is invisible and won't play unless controlled by JavaScript.
autoplay	Starts playing automatically on load. Most browsers block this unless the audio is also muted use sparingly, it can annoy visitors.
loop	Repeats the audio automatically when it ends.
src	Path to the audio file (can also be set via a nested <source> tag).

Example

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

Exercise

Add an <audio> element with controls to a page (a placeholder filename is fine), and include fallback text for browsers that don't support it.

<video>

Embeds a video file with browser-native playback controls.

Attributes

Attribute	What it does
controls	Shows play/pause/volume/fullscreen controls.
poster	An image shown before the video plays, e.g. poster="thumbnail.jpg".
autoplay / muted / loop	Same behavior as audio; autoplay generally requires muted to work in modern browsers.
width / height	Sets the display size of the video player.

Example

```
<video controls width="600" poster="thumbnail.jpg">
  <source src="promo.mp4" type="video/mp4">
  Your browser does not support the video tag.
</video>
```

Exercise

Add a `<video>` element that could showcase a Zolax Tech company promo, with controls, a poster image, and a width of 600 pixels.

`<track>`

A self-closing child of `<video>` or `<audio>` that adds subtitles, captions, or descriptions from a separate .vtt file important for accessibility.

Attributes

Attribute	What it does
kind	Type of track: "subtitles", "captions", "descriptions", or "chapters".
src	Path to the .vtt subtitle file.
srclang	Language code of the subtitles, e.g. "en".
label	Human-readable name shown in the subtitle-selection menu.

Example

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

Exercise

Add a `<track>` element to your `<video>` from the previous exercise offering English subtitles (the .vtt file doesn't need to exist for this exercise).

Chapter Summary Project

Chapter Project

Build a small 'Media Gallery' section for zolaxtech.com.et containing: one `<figure>` with an image and caption, one responsive `<picture>` element, and one `<video>` with controls, a poster, and a `<track>` for subtitles. Confirm every `<img>` and `<video>` has meaningful alt text or captions.

Chapter 7: Lists

Lists organize related items used constantly for navigation menus, service listings, and glossaries.

`<ul>`

Creates an unordered (bulleted) list, used when the order of items doesn't matter.

Example

```
<ul>
  <li>Web Development</li>
  <li>Mobile Apps</li>
  <li>UI/UX Design</li>
</ul>
```

Exercise

Create an unordered list of five services Zolax Tech might offer.

Creates an ordered (numbered) list, used when sequence matters such as steps in a process.

Attributes

Attribute	What it does
start	The number the list should begin counting from, e.g. start="3".
reversed	A boolean attribute that counts the list down instead of up.
type	The numbering style: "1" (default numbers), "A", "a", "I", or "i" (roman numerals).

Example

```
<ol>
  <li>Contact us for a quote</li>
  <li>Review the project proposal</li>
  <li>Sign the agreement</li>
  <li>Development begins</li>
</ol>
```

Exercise

Create an ordered list describing the four steps of how a client works with Zolax Tech, from first contact to project delivery.

Defines a single list item; must be a direct child of , , or <menu>.

Attributes

Attribute	What it does
value	Only valid inside overrides that item's number, and subsequent items continue counting from it.

Example

```
<li value="5">Skip ahead to item five</li>
```

Exercise

In your ordered list above, use the value attribute on one `<li>` to make the numbering jump, and observe how later items continue from that new number.

`<dl>`, `<dt>`, `<dd>`

A description list: `<dl>` is the container, `<dt>` holds a term, and `<dd>` holds its definition/description. Perfect for glossaries or FAQs.

Example

```
<dl>
  <dt>HTML</dt>
  <dd>The standard markup language used to structure web pages. </dd>
  <dt>CSS</dt>
  <dd>The language used to style and lay out web pages. </dd>
</dl>
```

Exercise

Build a small FAQ section using `<dl>`, with three questions as `<dt>` elements and their answers as `<dd>` elements.

Chapter Summary Project

Chapter Project

Build a Zolax Tech pricing/services page section that combines: an unordered list of 4 core services, a nested ordered list under one of the services listing the delivery steps (a `<ul>` can contain a nested `<ol>` inside one of its `<li>` items), and a short FAQ using `<dl>/<dt>/<dd>`.

Chapter 8: Tables

Tables display tabular (grid-like) data pricing plans, comparison charts, schedules. They should never be used just to lay out a page visually; that job belongs to CSS.

`<table>`

The container for the entire table.

Example

```
<table>
...
</table>
```

Exercise

Start a new file, pricing.html, with a bare <table> tag ready for content.

<caption>

A title for the table, displayed above it. Must be the first child of <table>.

Example

```
<table>
  <caption>Zolax Tech Pricing Plans</caption>
  ...
</table>
```

Exercise

Add a <caption> to your pricing table.

<thead>

Groups the header row(s) of the table, semantically separating them from the data.

Example

```
<thead>
  <tr><th>Plan</th><th>Price</th><th>Support</th></tr>
</thead>
```

Exercise

Add a <thead> to your pricing table with three column headers: Plan, Price, and Support.

<tbody>

Groups the main body rows of data, separate from the header and footer.

Example

```
<tbody>
  <tr><td>Basic</td><td>500 ETB</td><td>Email</td></tr>
  <tr><td>Pro</td><td>1500 ETB</td><td>Phone & Email</td></tr>
</tbody>
```

Exercise

Add a <tbody> with three rows of pricing data (Basic, Pro, Enterprise plans).

<tfoot>

Groups footer row(s), often used for totals or notes, and can be placed in the source before `<tbody>` while still rendering at the bottom.

Example

```
<tfoot>
  <tr><td colspan="3">Prices exclude VAT.</td></tr>
</tfoot>
```

Exercise

Add a `<tfoot>` row with a note about pricing, spanning all three columns.

`<tr>`

Defines a single table row, containing `<td>` or `<th>` cells.

Example

```
<tr><td>Basic</td><td>500 ETB</td><td>Email</td></tr>
```

Exercise

Add a fourth row to your table body for a custom/enterprise plan.

`<th>`

A header cell, bold and centered by default, used for column or row labels.

Attributes

Attribute	What it does
scope	Clarifies whether the header applies to a "col" or a "row" important for screen reader accessibility.
colspan	Number of columns this cell should span.
rowspan	Number of rows this cell should span.

Example

```
<th scope="col">Plan</th>
```

Exercise

Add `scope="col"` to each of your three `<th>` header cells.

`<td>`

A standard data cell within a table row.

Attributes

Attribute	What it does
colspan	Number of columns this cell should span, e.g. colspan="3" to stretch across the whole table.
rowspan	Number of rows this cell should span vertically.

Example

```
<td colspan="2">Includes hosting & domain</td>
```

Exercise

Use colspan on one cell in your table (for example in the footer note) to make it span all three columns.

<colgroup> & <col>

<colgroup> groups columns for styling purposes; each <col> inside it represents one column and can be styled (e.g. given a background color) without touching every individual cell.

Attributes

Attribute	What it does
span (on col)	Number of consecutive columns this <col> applies to.

Example

```
<colgroup>
  <col style="background-color:#f0f0f0;">
  <col span="2">
</colgroup>
```

Exercise

Add a <colgroup> to your pricing table that gives the first column a light background color, using a <col> element.

Chapter Summary Project

Chapter Project

Build a complete, valid pricing table for zolaxtech.com.et with a caption, thead (3 columns: Plan, Price, Support), tbody (3 rows of plans), tfoot (1 note row spanning all columns using colspan), and scope attributes on every header cell. Validate the result at validator.w3.org/nu.

Chapter 9: Forms & User Input

Forms let visitors send information to you contact forms, quote requests, newsletter sign-ups. This is one of the largest and most important tag families in HTML.

<form>

The container for all form controls; defines where and how the collected data is sent when submitted.

Attributes

Attribute	What it does
action	The URL that receives the form data when submitted, e.g. <code>action="/submit-contact.php"</code> .
method	How the data is sent: "get" (appends data to the URL, visible, used for searches) or "post" (sends data in the request body, used for sensitive or large data like contact forms).
autocomplete	"on" or "off" controls whether the browser may suggest previously entered values.
novalidate	A boolean attribute that turns off the browser's built-in validation (rarely used).

Example

```
<form action="/submit-contact.php" method="post">
...
</form>
```

Exercise

Start a new file, `contact.html`, with a `<form>` whose action points to a fictional `/submit-contact.php` endpoint and `method="post"`.

<label>

Provides an accessible, clickable text label for a form control. Clicking the label focuses/activates its associated input critical for accessibility and usability.

Attributes

Attribute	What it does
for	Must match the id of the associated input exactly, linking the two together.

Example

```
<label for="name">Full Name:</label>
<input type="text" id="name" name="name">
```

Exercise

Add a properly linked `<label>` and `<input>` pair for a visitor's name, using matching `for` and `id` values.

`<input>`

The most versatile form element a single self-closing tags whose behavior changes entirely based on its `type` attribute.

Attributes

Attribute	What it does
type="text"	A basic single-line text field (the default type if omitted).
type="email"	A text field that validates the value looks like an email address and shows an email-optimized keyboard on mobile.
type="password"	Masks the entered characters as dots.
type="number"	Only accepts numeric input; often paired with <code>min</code> , <code>max</code> , and <code>step</code> .
type="tel"	A field for phone numbers; shows a numeric keypad on mobile.
type="checkbox"	A toggle box for yes/no or multi-select choices.
type="radio"	A round button where only one option in a group (matched by the same name) can be selected.
type="date"	Shows a native date picker.
type="file"	Lets the user upload a file from their device.
type="range"	A slider control between a <code>min</code> and <code>max</code> value.
type="color"	A native color-picker swatch.
type="submit"	A button that submits the form.
type="hidden"	A field not shown to the user but still submitted with the form (e.g. for tracking data).
name	The key under which this field's value is submitted required for the data to be sent at all.
placeholder	Faint example text shown inside an empty field, disappearing once the user types.
required	A boolean attribute preventing form submission until this field has a value.
value	The field's default (or current) value.
min / max / step	For number/range/date types: the allowed minimum, maximum, and increment.
pattern	A regular expression the value must match to be considered valid.

Attribute	What it does
disabled	A boolean attribute that greys out the field and prevents interaction or submission.
readonly	Prevents editing but still submits the value and still allows focus/selection, unlike disabled.

Example

```
<input type="text" name="name" placeholder="Your full name" required>
<input type="email" name="email" placeholder="you@example.com" required>
<input type="tel" name="phone" pattern="[0-9]{10}">
<input type="number" name="budget" min="100" max="100000" step="100">
<input type="date" name="deadline">
<input type="checkbox" name="newsletter" id="newsletter">
<input type="radio" name="plan" value="basic" id="basic">
<input type="radio" name="plan" value="pro" id="pro">
<input type="file" name="attachment">
<input type="submit" value="Send Request">
```

Exercise

Build a Zolax Tech quote-request form containing: a required text input for name, a required email input, a tel input with a pattern, a number input for budget with min/max, a date input for the preferred deadline, two radio buttons for choosing Basic or Pro plan, one checkbox to opt into a newsletter, a file input to attach a brief, and a submit button.

<textarea>

A multi-line free-text input box, used for longer messages such as project descriptions.

Attributes

Attribute	What it does
rows / cols	Sets the visible height (rows) and width (cols) of the box in character units.
placeholder	Faint hint text shown when empty.
maxlength	The maximum number of characters allowed.

Example

```
<label for="message">Project Details:</label>
<textarea id="message" name="message" rows="5" cols="40" maxlength="500"
placeholder="Describe your project..."></textarea>
```

Exercise

Add a `<textarea>` to your quote form for the client to describe their project, with a placeholder and a 500-character limit.

`<select>`, `<option>`, `<optgroup>`

`<select>` creates a dropdown menu; each `<option>` is one choice inside it; `<optgroup>` groups related options under a shared label within the dropdown.

Attributes

Attribute	What it does
name (on select)	The key this field's chosen value is submitted under.
multiple (on select)	A boolean attribute allowing more than one option to be selected at once.
value (on option)	The actual value sent to the server; if omitted, the option's text content is used instead.
selected (on option)	A boolean attribute marking this option as chosen by default.
label (on optgroup)	The heading text shown above a group of related options.

Example

```
<label for="service">Service Needed:</label>
<select id="service" name="service">
  <optgroup label="Development">
    <option value="web">Web Development</option>
    <option value="mobile" selected>Mobile App</option>
  </optgroup>
  <optgroup label="Design">
    <option value="uiux">UI/UX Design</option>
  </optgroup>
</select>
```

Exercise

Add a `<select>` dropdown to your quote form letting the client choose a service, with at least four `<option>` elements grouped into two `<optgroup>` categories.

`<button>`

A clickable button that can submit a form, reset a form, or (with JavaScript) trigger any custom action, depending on its type.

Attributes

Attribute	What it does
type="submit"	Submits the enclosing form (this is the default type inside a <form>).
type="reset"	Clears all fields in the form back to their default values.
type="button"	Does nothing on its own used when you want JavaScript to control the click behavior entirely.

Example

```
<button type="submit">Send Request</button>
<button type="reset">Clear Form</button>
```

Exercise

Add a Reset button next to your Submit button on the quote form and test that it clears all fields.

<fieldset> & <legend>

<fieldset> visually and semantically groups related form controls inside a border; <legend> provides a caption for that group, read aloud first by screen readers.

Attributes

Attribute	What it does
disabled (on fieldset)	A boolean attribute that disables every control inside the group at once.

Example

```
<fieldset>
  <legend>Contact Preference</legend>
  <input type="radio" name="pref" id="pref-email" value="email">
  <label for="pref-email">Email</label>
  <input type="radio" name="pref" id="pref-phone" value="phone">
  <label for="pref-phone">Phone</label>
</fieldset>
```

Exercise

Group your two plan radio buttons from the <input> exercise inside a <fieldset> with a <legend> reading 'Choose a Plan'.

<datalist>

Provides a list of pre-defined autocomplete suggestions for a connected text input, while still allowing the user to type a custom value.

Example

```
<input list="cities" name="city">
<datalist id="cities">
  <option value="Addis Ababa">
  <option value="Adama">
  <option value="Hawassa">
</datalist>
```

Exercise

Add a text input with a connected `<datalist>` suggesting three Ethiopian cities for a 'City' field.

<output>

Represents the result of a calculation, typically updated by JavaScript, for example showing the sum of two inputs.

Attributes

Attribute	What it does
for	Space-separated list of input IDs whose values this output is calculated from.

Example

```
<input type="number" id="a"> + <input type="number" id="b">
<output for="a b">0</output>
```

Exercise

Add two number inputs and an `<output>` element between them showing a placeholder value of 0 (you do not need to wire up the JavaScript).

<progress>

Displays a progress bar showing completion toward a task, such as a file upload percentage.

Attributes

Attribute	What it does
value	Current progress amount.
max	The value that represents 100% completion.

Example

```
<progress value="70" max="100">70%</progress>
```

Exercise

Add a `<progress>` bar showing 3 out of 5 steps completed in a fictional project.

`<meter>`

Displays a scalar value within a known range, such as disk usage or a rating visually similar to `<progress>` but semantically different (a gauge/measurement, not a task's completion).

Attributes

Attribute	What it does
value	The current value.
min / max	The range boundaries.
low / high / optimum	Define thresholds used to color the gauge red/yellow/green.

Example

```
<label for="rating">Client Satisfaction:</label>
<meter id="rating" value="4.5" min="0" max="5">4.5 out of 5</meter>
```

Exercise

Add a `<meter>` element showing a 4.5-out-of-5 client satisfaction score.

Chapter Summary Project

Chapter Project

Build a complete, realistic 'Request a Quote' form for `zolaxtech.com.et` in `contact.html` combining at least twelve of the tags/attributes covered in this chapter: form, labeled text/email/tel/number/date inputs, radio buttons grouped in a fieldset with a legend, a select with optgroups, a textarea, a file input, a datalist-powered input, and submit/reset buttons. Open it with Live Server and manually test that the required fields block submission when empty.

Chapter 10: Semantic HTML5 Layout Tags

HTML5 introduced tags that describe the MEANING of a section of a page, not just its box shape. Using them instead of generic `<div>`s all over the place improves SEO, accessibility, and code readability this is what separates 'beginner HTML' from 'professional HTML5'.

`<header>`

Represents introductory content for its nearest section or the whole page typically a logo, site title, and main navigation. A page can have more than one `<header>` (e.g. one for the page, one inside an `<article>`).

Example

```
<header>
  <h1>Zolax Tech</h1>
  <nav>...</nav>
</header>
```

Exercise

Wrap your site logo, title, and navigation menu in a `<header>` element.

`<nav>`

Marks a section containing major navigation links (main menu, breadcrumbs, table of contents). Not every group of links needs `<nav>` reserve it for significant navigation blocks.

Example

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

Exercise

Wrap your existing navigation link list from Chapter 5 in a `<nav>` element.

`<main>`

Represents the dominant, unique content of the page excludes anything repeated across pages like headers, footers, and sidebars. A page should have only one visible `<main>`.

Example

```
<main>
  <h1>Welcome to Zolax Tech</h1>
  <p>We build modern websites and apps for businesses across Ethiopia. </p>
</main>
```

Exercise

Wrap the primary unique content of your homepage (excluding header/footer/nav) in a single `<main>` element.

`<section>`

Represents a distinct, thematically-grouped section of content, usually with its own heading. Ask yourself: 'would this make sense as its own entry in a table of contents?' If yes, `<section>` fits.

Example

```
<section>
  <h2>Our Services</h2>
  <p>We offer web development, mobile apps, and UI/UX design. </p>
</section>
```

Exercise

Turn your 'Services', 'About', and 'Contact' blocks into three separate `<section>` elements, each starting with its own heading.

`<article>`

Represents a self-contained piece of content that could be distributed or reused independently a blog post, news story, or product card. Test: would this make sense as a standalone RSS feed item?

Example

```
<article>
  <h2>Zolax Tech Launches New Mobile App</h2>
  <p>Published August 2026...</p>
</article>
```

Exercise

Create a small 'News' section with two `<article>` elements, each with its own heading and one paragraph, simulating company announcements.

`<aside>`

Represents content tangentially related to the surrounding content a sidebar, pull quote, or related-links box.

Example

```
<aside>
  <h3>Related Services</h3>
  <ul><li>SEO Optimization</li><li>Hosting</li></ul>
</aside>
```

Exercise

Add an `<aside>` next to your Services section listing two related services.

`<footer>`

Represents footer content for its nearest section or the whole page typically copyright, contact links, and site map links. Like `<header>`, more than one is allowed per page.

Example

```
<footer>
  <p><small>&copy; 2026 Zolax Tech. All rights reserved. </small></p>
</footer>
```

Exercise

Wrap your copyright notice and secondary links at the bottom of the page in a `<footer>` element.

`<details>` & `<summary>`

`<details>` creates a native, JavaScript-free expand/collapse widget; `<summary>` is the always-visible label the user clicks to toggle it.

Attributes

Attribute	What it does
open (on details)	A boolean attribute that makes the widget expanded by default instead of collapsed.

Example

```
<details>
  <summary>What is your refund policy? </summary>
  <p>We offer a 14-day money-back guarantee on all packages. </p>
</details>
```

Exercise

Rebuild your FAQ section from Chapter 7 using `<details>/<summary>` pairs instead of `<dl>`, and compare the interactive behavior.

`<dialog>`

Represents a native modal or non-modal popup dialog box, which can be shown/hidden with JavaScript (`dialog.showModal()` / `dialog.close()`).

Attributes

Attribute	What it does
open	A boolean attribute controlling whether the dialog is currently visible.

Example

```
<dialog open>
  <p>Thank you for contacting Zolax Tech! We'll respond within 24 hours. </p>
  <button onclick="this.closest('dialog').close()">Close</button>
</dialog>
```

Exercise

Add a `<dialog>` that could serve as a 'thank you' confirmation after form submission, with a close button.

`<time>`

Marks content representing a specific date and/or time in a machine-readable way, useful for calendars and search engines.

Attributes

Attribute	What it does
datetime	The machine-readable date/time value, e.g. <code>datetime="2026-08-07"</code> , independent of how the text is displayed.

Example

```
<p>Published on <time datetime="2026-08-07">August 7, 2026</time>.</p>
```

Exercise

Add a `<time>` element to one of your news articles from the `<article>` exercise, showing a human-readable date with a matching machine-readable `datetime` attribute.

Chapter Summary Project

Chapter Project

Rebuild your ENTIRE `zolaxtech.com.et` homepage using only semantic tags for structure replace generic `<div>`s with `header`, `nav`, `main`, `section`, `article`, `aside`, and `footer` wherever appropriate. Include at least one `<details>/<summary>` FAQ item and one `<time>` element. When finished, use the browser's Accessibility/DOM inspector (right-click > Inspect) to confirm the outline reads logically.

Chapter 11: Embedded Content & Graphics

Beyond images, audio, and video, HTML5 can embed entire other pages, plugins, and even draw graphics directly with code. These are considered more advanced tags.

`<iframe>`

Embeds another, entirely separate HTML page inside the current page commonly used for embedding Google Maps, YouTube videos, or payment widgets.

Attributes

Attribute	What it does
src	URL of the page to embed.

Attribute	What it does
width / height	Dimensions of the embedded frame in pixels.
title	A required accessible description of the embedded content for screen readers.
loading	loading="lazy" defers loading until the iframe is near the viewport.
allowfullscreen	A boolean attribute permitting the embedded content to be shown fullscreen (needed for embedded video players).

Example

```
<iframe src="https://www.google.com/maps/embed?." width="600" height="450" title="Zolax Tech office location" loading="lazy"></iframe>
```

Exercise

Embed a Google Maps iframe (use Google Maps' 'Share > Embed a map' feature to get real embed code, or a placeholder src) showing a fictional Zolax Tech office location, with a descriptive title attribute.

<embed>

A generic self-closing tag for embedding external content handled by a plugin, such as a PDF viewer.

Attributes

Attribute	What it does
src	Path to the resource.
type	MIME type of the resource, e.g. application/pdf.
width / height	Display dimensions.

Example

```
<embed src="brochure.pdf" type="application/pdf" width="600" height="400">
```

Exercise

Add an <embed> tag that could display a company brochure PDF at 600x400 pixels.

<object>

Similar to <embed> but with a standardized fallback: content placed inside the tags is shown only if the object type is not supported.

Attributes

Attribute	What it does
data	Path to the resource being embedded.
type	MIME type of the resource.
width / height	Display dimensions.

Example

```
<object data="brochure.pdf" type="application/pdf" width="600" height="400">
  <p>Your browser can't display PDFs. <a href="brochure.pdf">Download it here</a>.</p>
</object>
```

Exercise

Rewrite the PDF example above using `<object>` instead of `<embed>`, including a fallback paragraph with a direct download link.

`<canvas>`

Creates a blank drawing surface that JavaScript can use to draw graphics, charts, animations, or games pixel-by-pixel. The tag itself does nothing without JavaScript.

Attributes

Attribute	What it does
width / height	The canvas's drawing surface dimensions in pixels (set as attributes, not CSS, to avoid distortion).

Example

```
<canvas id="logo-animation" width="400" height="200"></canvas>
<script>
  const ctx = document.getElementById("logo-animation").getContext("2d");
  ctx.fillStyle = "navy";
  ctx.fillRect(10, 10, 150, 80);
</script>
```

Exercise

Add a `<canvas>` element 400x200 pixels, then write a short inline `<script>` that draws a filled rectangle on it using `getContext("2d")` and `fillRect()`, as shown in the example.

`<svg>`

Embeds Scalable Vector Graphics directly in the HTML resolution-independent shapes, icons, and logos defined with markup rather than pixels, so they stay crisp at any zoom level.

Attributes

Attribute	What it does
viewBox	Defines the coordinate system of the drawing, e.g. viewBox="0 0 100 100".
width / height	The displayed size of the SVG on the page.

Example

```
<svg viewBox="0 0 100 100" width="100" height="100">
  <circle cx="50" cy="50" r="40" fill="darkblue" />
</svg>
```

Exercise

Add an inline `<svg>` that draws a simple colored circle or rectangle logo mark, sized 100x100 pixels.

Chapter Summary Project

Chapter Project

Add an 'Our Location & Brochure' section to zolaxtech.com.et containing a lazy-loaded `<iframe>` map embed with a title attribute, an `<object>` showing a PDF brochure with a fallback download link, and a small inline `<svg>` icon used as a decorative divider.

Chapter 12: Global Attributes & Data Attributes

Global attributes can be used on almost ANY HTML element, regardless of tag. Understanding them well lets you style, script, and structure any part of your page.

id

Assigns a unique identifier to a single element no two elements on the same page should share an id. Used to target one specific element with CSS (`#id`), JavaScript (`getElementById`), or as a link anchor (`#id` in a URL).

Example

```
<section id="contact">...</section>
<!-- linked from elsewhere: -->
<a href="#contact">Jump to Contact</a>
```

Exercise

Give each major `<section>` on your homepage a unique id, then create a 'back to top' link that jumps to an id placed on your `<header>`.

class

Assigns one or more (space-separated) reusable names to an element, used to apply the same CSS styling or JavaScript behavior to many elements at once. Unlike id, many elements can share a class.

Example

```
<div class="card featured">...</div>
<div class="card">...</div>
```

✎ Exercise

Add a shared class of 'service-card' to each of your service <div>s or <article>s so they can all be styled identically with one CSS rule.

style

Applies inline CSS directly to a single element. Convenient for quick tests, but generally discouraged for real projects in favor of external stylesheets with classes, since inline styles are hard to maintain and override.

Example

```
<p style="color: red; font-weight: bold;">Limited-time offer!</p>
```

✎ Exercise

Find one place in your page where you used a style attribute earlier in this course, and refactor it into a class defined in styles.css instead.

title

Adds advisory text shown as a tooltip when the user hovers over the element.

Example

```
<abbr title="Search Engine Optimization">SEO</abbr>
```

✎ Exercise

Add a title attribute to your site logo image giving extra context, e.g. 'Zolax Tech Web Development in Ethiopia'.

data-*

Custom attributes for storing extra data on an element that JavaScript can read, without affecting the page's appearance or validity. The name after 'data-' can be anything you choose.

Example

```
<button data-service-id="42" data-price="1500">Book Now</button>
<script>
```

```
const btn = document.querySelector("button");
console.log (btn.dataset.serviceId, btn.dataset.price);
</script>
```

Exercise

Add data-service-id and data-price attributes to one of your 'Book Now' or 'Get Quote' buttons, then write a short script that logs both values to the console when the button is clicked.

tabindex

Controls whether and in what order an element can be focused using the Tab key. `tabindex="0"` adds an element to the natural tab order; a positive number sets a custom order (avoid this it usually creates confusing navigation); `tabindex="-1"` removes it from tab order but allows programmatic focus.

Example

```
<div tabindex="0">This div can now be focused with the keyboard.</div>
```

Exercise

Add `tabindex="0"` to one custom interactive element (like a `<div>` acting as a card) and test that it can be reached by repeatedly pressing Tab in the browser.

contenteditable

A boolean-like attribute ("`true`"/"`false`") that lets the visitor directly edit an element's text in the browser the foundation of simple rich-text editors.

Example

```
<div contenteditable="true">Click here and start typing to edit this text.</div>
```

Exercise

Add a `<div contenteditable="true">` to a scratch test page and confirm you can click into it and type directly in the browser.

hidden

A boolean attribute that hides an element completely (equivalent to `display:none`), removing it from the visible page and from screen readers.

Example

```
<p hidden>This paragraph will not be visible on the page.</p>
```

☛ Exercise

Add a paragraph with the hidden attribute to a test page and confirm it does not appear, then remove the attribute and confirm it does.

draggable

A boolean-like attribute ("true"/"false") enabling the HTML5 native drag-and-drop API on an element.

Example

```

```

☛ Exercise

Add draggable="true" to an image on a test page and confirm in the browser that you can begin dragging it (full drag-and-drop functionality requires JavaScript event listeners, which are outside the scope of this HTML course).

lang (on any element)

Can override the page's overall language for a specific element, e.g. marking a single foreign-language phrase inside otherwise English content.

Example

```
<p>Our greeting in Amharic is <span lang="am">ሰላም</span>.</p>
```

☛ Exercise

Find or write a short non-English phrase somewhere on your page and mark it with the correct lang attribute on a .

Chapter Summary Project

🏁 Chapter Project

Go back through your entire zolaxtech.com.et project and deliberately apply at least six different global attributes from this chapter across various elements: ids for navigation anchors, shared classes for repeated components, a data-* attribute on an interactive button, and a hidden element you can reveal later with JavaScript.

Chapter 13: Advanced HTML5 SEO, Accessibility & Best Practices

This final content chapter ties everything together with the practices that separate a working page from a professional, discoverable, accessible one critical for a real business site like zolaxtech.com.et.

Meta tags for SEO

Beyond description, several meta and link tags directly influence how search engines and social platforms understand and display your site.

Attributes

Attribute	What it does
<code><meta name="keywords"></code>	Largely ignored by modern search engines but still sometimes included, e.g. <code>content="web development, Ethiopia, Addis Ababa"</code> .
<code><meta property="og:title"></code>	Part of the Open Graph protocol used by Facebook/LinkedIn to show a rich preview card when your link is shared.
<code><meta property="og:description"></code>	The preview description shown in that social card.
<code><meta property="og:image"></code>	The preview image shown in that social card.
<code><link rel="canonical"></code>	Tells search engines the single 'official' URL for a page, preventing duplicate-content penalties, e.g. <code>href="https://zolaxtech.com.et/"</code> .

Example

```
<meta property="og:title" content="Zolax Tech - Web Development in Ethiopia">
<meta property="og:description" content="We build fast, modern websites for Ethiopian businesses.">
<meta property="og:image" content="https://zolaxtech.com.et/og-image.jpg">
<link rel="canonical" href="https://zolaxtech.com.et/">
```

Exercise

Add Open Graph meta tags and a canonical link to your homepage's `<head>`, then test how it would preview by pasting your (real or placeholder) URL into a social media share-preview debugger tool.

ARIA attributes (accessibility)

Accessible Rich Internet Applications attributes add extra information for assistive technology (screen readers) when semantic HTML alone isn't enough most often needed on custom, JavaScript-driven widgets.

Attributes

Attribute	What it does
aria-label	Provides an accessible name for an element with no visible text, e.g. an icon-only button.
aria-hidden="true"	Hides purely decorative content (like a background icon) from screen readers while keeping it visible on screen.
role	Clarifies the purpose of a generic element when no semantic tag fits, e.g. role="alert" on a <div> that shows form-error messages.
aria-required="true"	Explicitly signals a required field to assistive tech, supplementing the required attribute.

Example

```
<button aria-label="Close menu">✕</button>
<div role="alert">Please fill in all required fields.</div>
```

Exercise

Add an icon-only close button (using an X character) to a test menu, with an aria-label describing its function, and add role="alert" to an error-message <div> on your contact form.

Favicon setup

The small icon shown in browser tabs and bookmarks, set with one or more <link rel="icon"> tags, ideally in multiple sizes for different devices.

Attributes

Attribute	What it does
sizes	Specifies the icon's pixel dimensions, e.g. sizes="32x32", letting the browser pick the best fit.
type	MIME type of the icon file, e.g. "image/png" or "image/x-icon".

Example

```
<link rel="icon" href="favicon-32.png" sizes="32x32" type="image/png">
<link rel="apple-touch-icon" href="apple-touch-icon.png">
```

Exercise

Add a favicon link (a placeholder filename is fine) to your <head>, plus an apple-touch-icon link for iOS home-screen bookmarks.

Character entities

Special reserved characters (like < or &) and symbols not on a standard keyboard must be written as HTML entities so the browser doesn't misinterpret them as code.

Attributes

Attribute	What it does
&lt; / &gt;	Display a literal < or > character (otherwise interpreted as the start of a tag).
&amp;	Displays a literal ampersand &.
&copy;	Displays the copyright symbol ©.
&nbsp;	A non-breaking space a space that won't allow a line break at that point.
&quot;	Displays a literal double-quote character inside an attribute value.

Example

```
<p>Use the &lt;p&gt; tag for paragraphs. Tom &amp; Jerry. &copy; 2026 Zolax Tech.</p>
```

Exercise

Rewrite the sentence 'Use <div> & together' safely using the correct HTML entities so it displays correctly in the browser instead of being interpreted as broken tags.

Form validation attributes (recap & advanced)

HTML5 can validate many common cases without any JavaScript at all, giving instant feedback and reducing bad data reaching your server.

Attributes

Attribute	What it does
required	Field must be filled before submission.
pattern	A regular expression the value must match.
min / max / minlength / maxlength	Numeric or length boundaries on the input's value.
novalidate (on form)	Disables all of the above for the entire form at once, usually when validation will instead be handled entirely by JavaScript.

Example

```
<input type="text" name="promo-code" pattern="[A-Z]{3}-[0-9]{4}" minlength="8" maxlength="8" required title="Format: ABC-1234">
```

Exercise

Add a promo-code field to your quote form from Chapter 9 that requires a pattern of three uppercase letters, a hyphen, and four digits (e.g. ZLX-2026), with a title attribute explaining the expected format.

Chapter Summary Project

Chapter Project

Perform a full 'launch readiness' pass on zolaxtech.com.et: add Open Graph tags and a canonical link, add a favicon (multiple sizes), audit every image for meaningful alt text, add at least two ARIA attributes where custom widgets need them, replace any raw `<`, `>`, or `&` characters in visible text with proper entities, and run the entire site through validator.w3.org/nu until zero errors remain.

Appendix A: Useful VS Code Shortcuts for HTML

Shortcut	What it does
! + Tab	Generate a full HTML5 boilerplate (Emmet).
Ctrl + S	Save the current file (triggers Live Server auto-refresh).
Ctrl + /	Comment or uncomment the selected line(s).
Alt + Shift + F	Auto-format/indent the whole document.
Ctrl + Space	Trigger autocomplete suggestions for tags and attributes.
Ctrl + D	Select the next occurrence of the current word (great for renaming repeated class names).
div.card + Tab	Emmet shortcut: expands to <code><div class="card"></div></code> .
ul>li*5 + Tab	Emmet shortcut: expands to a <code></code> with five <code></code> children.
Ctrl + `	Open/close the integrated terminal.

Appendix B: Final Capstone Project

Using everything from this guide, build a complete, multi-page website for zolaxtech.com.et in Visual Studio Code with Live Server, consisting of:

1. `index.html` homepage using semantic layout tags (header, nav, main, section, footer), a hero heading, an image, and a summary of services.
2. `about.html` company story using headings, paragraphs, a `<figure>` team photo, and an `<address>` block.
3. `services.html` services listed with headings, an unordered list, and a pricing `<table>` with `thead/tbody/tfoot`.
4. `portfolio.html` a gallery section using `<figure>/<figcaption>` and a responsive `<picture>` element.
5. `contact.html` a full contact/quote `<form>` with at least eight different `<input>` types, a `<textarea>`, a `<select>`, and validation attributes.
6. A shared `styles.css` linked from every page, and a shared `<nav>` menu (copy-pasted or, for advanced students, built once and reused).
7. Every page passes validator.w3.org/nu with zero errors, and every image has meaningful alt text.