

HTML

Lesson Objectives:

By the end of this lesson, students will be able to:

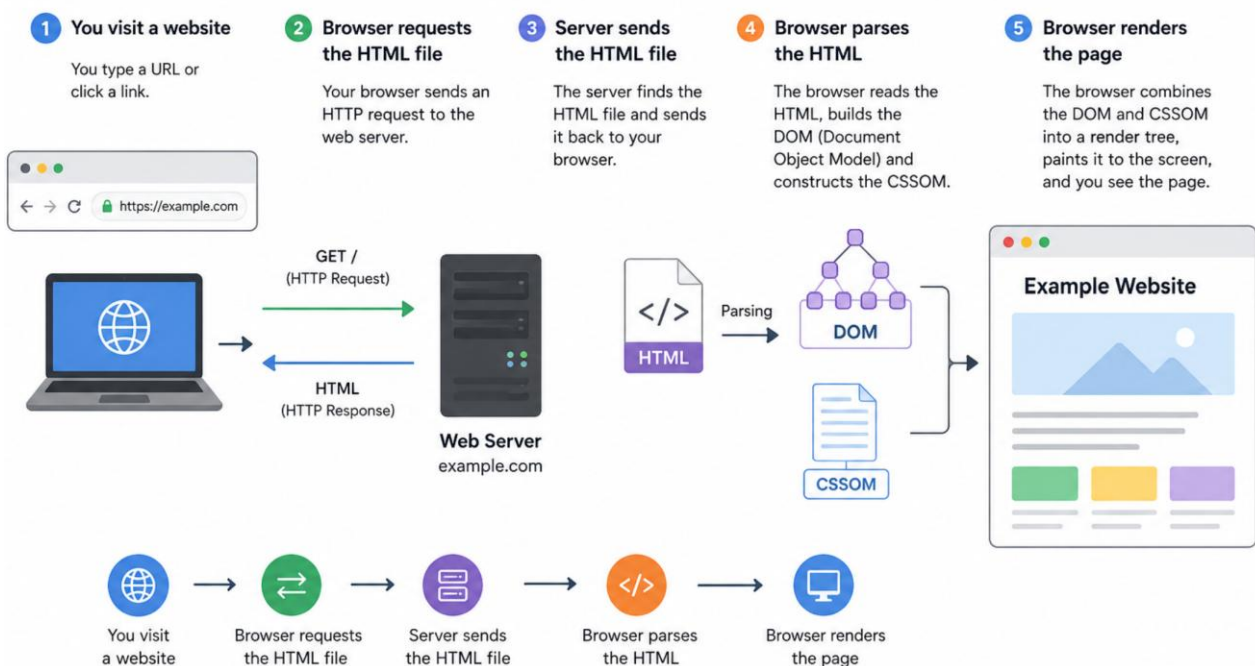
- **Define** what HTML is and explain its purpose in creating web pages.
- **Identify** basic HTML tags such as `<html>`, `<head>`, `<title>`, `<body>`, `<h1>` and `<p>`.
- **Create** a simple HTML document using headings and paragraphs.
- **Use** opening and closing tags correctly.
- **Save** an HTML file with the correct file extension: `.html`.
- **Open** an HTML file in a web browser to view the webpage.
- **Explain** the difference between the code written in an HTML file and the webpage displayed in a browser.

What is html?

HTML stands for **HyperText Markup Language**. It is the language used to define the structure and content of every page on the World Wide Web. When you visit a website, your browser downloads an HTML file from a web server, parses it, and renders the result as the page you see on screen.

The most important thing to understand about HTML is what it is *not*. HTML is a **markup language**, not a programming language. It has no variables, no loops, no selection and no logic of any kind. It cannot calculate, decide or repeat. All it does is annotate — “mark up” — content, telling the browser which piece of text is a heading, which is a paragraph, which is a link, and so on. Examiners frequently test this distinction, and calling HTML a programming language in an exam answer will cost marks.

When you visit a website, your browser downloads an HTML file from a web server, parses it, and renders the result as the page you see on screen.



HTML

How would you tell a web browser what this document should look like?

The Home Baker's Blog

Welcome to my kitchen! Baking bread at home is **easier than you think**, and the smell of a fresh loaf is *absolutely unbeatable*.

 A fresh sourdough loaf on a cooling rack

Getting Started with Sourdough

Sourdough only needs three ingredients: flour, water and salt. The magic comes from a *starter* — a live culture of wild yeast that you feed daily.

What You'll Need

- 500g strong white bread flour
- 350ml lukewarm water
- 10g salt
- 100g active sourdough starter

The Basic Steps

1. Mix the flour and water, then rest for 30 minutes.
2. Add the starter and salt, and knead briefly.
3. Fold the dough every 30 minutes for 2 hours.
4. Shape, prove overnight in the fridge, then bake.

 A jar of bubbly sourdough starter

Further Reading

For a deeper dive into bread science, visit [the Wikipedia article on sourdough](#).

The three layers of the web

Modern web pages are built from three separate technologies, each with a distinct responsibility. Keeping the three concerns separate makes websites easier to build, maintain and restyle, and it is a principle you should be able to explain in the exam.

Technology	Responsibility	Example
HTML	Structure and content	"This text is a heading"
CSS	Style and presentation	"Headings are blue and centred"
JavaScript	Behaviour and interactivity	"When clicked, show a menu"

KEY FACT HTML defines **what the content is**; CSS defines **how it looks**; JavaScript defines **what it does**.
This separation of concerns is a classic two-mark explanation.

HTML

The Document Skeleton

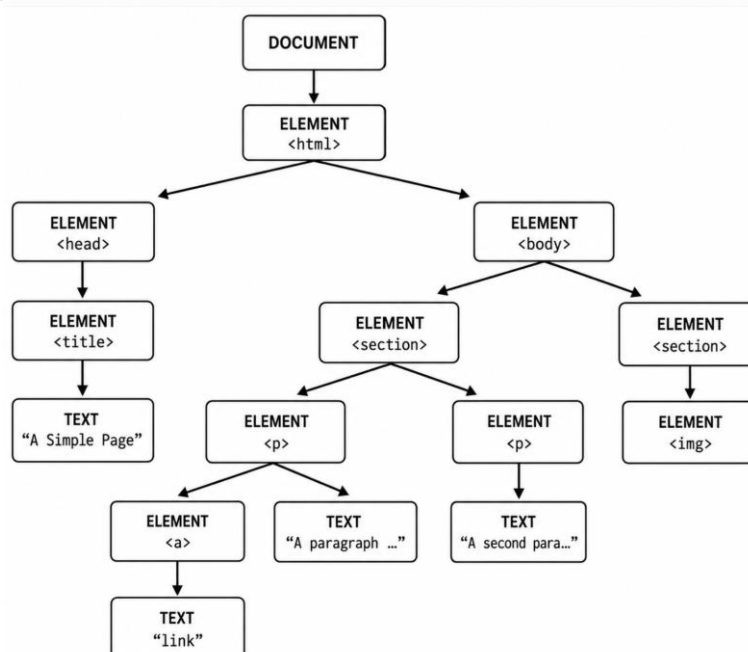
Every HTML page follows the same basic structure, and you should be able to write it from memory. A minimal, valid page looks like this:

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Page</title>
  </head>
  <body>
    <h1>Welcome</h1>
    <p>Page content...</p>
  </body>
</html>
```

The **<!DOCTYPE html>** declaration comes first and tells the browser that the document is written in HTML5. It is a declaration rather than an element, so it has no closing tag. The **<html>** element is the root of the document: every other element lives inside it.

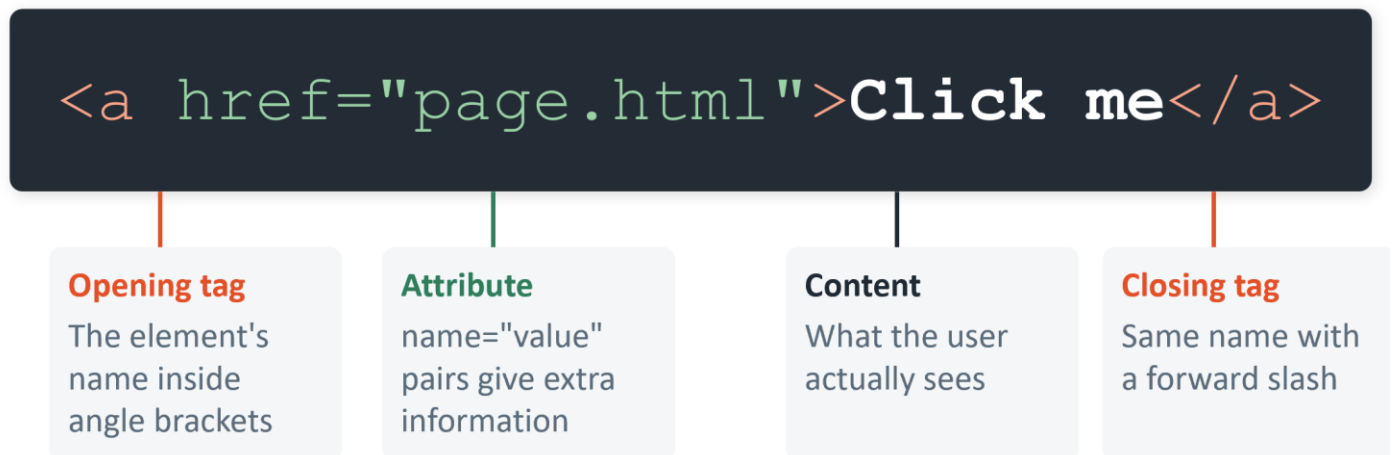
Inside the root are exactly two sections. The **<head>** contains metadata — information about the page that is not displayed in the page itself, such as the title, the character set, and links to CSS stylesheets. The **<body>** contains everything the user actually sees: headings, paragraphs, images, links and so on.

KEY FACT The **<title>** element sets the text shown in the browser tab and in bookmarks. It does **not** put a heading on the page — that is the job of **<h1>**.



HTML

Anatomy of an Element



HTML documents are built from **elements**. A typical element has three parts: an opening tag, some content, and a closing tag. Consider a simple hyperlink:

The **opening tag** `<a ...>` gives the element's name inside angle brackets. The **content** — here the words “Click me” — is what the user actually sees on the page. The **closing tag** `</a>` repeats the element's name with a forward slash, marking where the element ends.

Attributes

Attributes provide extra information about an element. They live inside the opening tag only, and are written as `name="value"` pairs — the value should always be wrapped in quotation marks. In the example above, `href="page.html"` is an attribute telling the browser where the link should go. An element can carry several attributes, separated by spaces.

Void elements

A small group of elements, called **void elements** (or empty elements), have no content and therefore no closing tag. The most important ones to remember are `<br>`, `<hr>`, `<img>` and `<input>`. Writing a closing tag such as `</img>` or `</br>` is an error.

EXAM TIP A very common exam slip is inventing closing tags for void elements. If you cannot put content between the tags, the element does not have a closing tag.

Nesting

Elements can be placed inside one another, which is called **nesting**. Nested tags must be closed in reverse order of opening — the last element opened is the first one closed. `<strong><em>text</em></strong>` is correct; `<strong><em>text</strong></em>` is invalid because the tags cross over.

HTML

Headings and Text

Headings

HTML provides six levels of heading, from `<h1>` (the largest and most important) down to `<h6>` (the smallest). Headings are not just about size: they describe the structure of the document, like the chapter and section titles of a book. A well-structured page uses a single `<h1>` for its main title and nests lower heading levels logically beneath it. This hierarchy matters for accessibility, because screen-reader users navigate by jumping between headings, and for search engines, which use headings to understand what a page is about.

Paragraphs

Body text is wrapped in `<p>` elements, one per paragraph. Line breaks and extra spaces inside the source code are ignored — the browser collapses all whitespace and reflows the text to fit the window. If you genuinely need a line break within a paragraph (for a poem or an address, for example), use the void element `<br>`. To draw a horizontal dividing line across the page, use `<hr>` (horizontal rule).

Bold and italics

The `<strong>` element marks text as important, which browsers display in bold. The `<em>` element marks text as emphasised, displayed in italics. Older tags `<b>` and `<i>` achieve the same visual result, but `<strong>` and `<em>` are preferred because they carry semantic meaning — they say why the text stands out, not merely how it should look.

```
<h1>Biggest heading</h1>
<h2>Sub-heading</h2>
<p>A paragraph with <strong>important</strong>
and <em>emphasised</em> words.</p>
<hr>
```

Task 1: Create a Webpage About Morse Code

Open a notebook, call it morse.txt. Save your work and then save the final version as **morse.html**

Write an HTML document about **Morse code** using headings and paragraphs. It should look like the diagram below.

You must use: <h1>, <h2>, <h3>, <p>, , ,
 and <hr>

Morse Code

Morse code is a system used to send messages using dots and dashes.

What is Morse Code?

Morse code uses short signals called dots and long signals called dashes.

Each letter of the alphabet has its own pattern of dots and dashes.

How Morse Code Can Be Sent

Morse code can be sent using sound, light or electrical signals.

For example, a short flash of light can mean a dot, and a long flash of light can mean a dash.

Example

The letter A is written as **dot dash**.

The emergency signal SOS is written as *dot dot dot, dash dash dash, dot dot dot*.

Why Morse Code Was Useful

Morse code was useful because it allowed people to send messages over long distances.

It was used by sailors, telegraph operators and emergency services.

Morse code is still interesting to learn today because it shows how information can be sent using simple signals.

HTML

ANSWER

```
<!DOCTYPE html>

<html>

<head>
  <title>Morse Code</title>
</head>

<body>

  <h1>Morse Code</h1>

  <p>Morse code is a system used to send messages using dots and dashes.</p>

  <h2>What is Morse Code?</h2>

  <p>Morse code uses short signals called dots and long signals called dashes.</p>

  <p>Each letter of the alphabet has its own pattern of dots and dashes.</p>

  <h2>How Morse Code Can Be Sent</h2>

  <p>Morse code can be sent using sound, light or electrical signals.</p>
  <p>For example, a short flash of light can mean a dot, and a long flash of light can mean a
dash.</p>

  <h3>Example</h3>
  <p>The letter A is written as <strong>dot dash</strong>.</p>
  <p>The emergency signal SOS is written as <em>dot dot dot, dash dash dash, dot dot
dot</em>.</p>

  <hr>

  <h2>Why Morse Code Was Useful</h2>

  <p>Morse code was useful because it allowed people to send messages over long distances.</p>
  <p>It was used by sailors, telegraph operators and emergency services.</p>
  <br>

  <p>Morse code is still interesting to learn today because it shows how information can be sent
using simple signals.</p>
</body>

</html>
```

HTML

Links

Hyperlinks are the “HyperText” in HTML — they are what connects individual pages into a web. A link is created with the `<a>` (anchor) element, and the destination is given by its **href** attribute. The content between the tags becomes the clickable text.

```
<a href="https://www.bbc.co.uk">External link (absolute URL)</a>
<a href="contact.html">Same-site link (relative URL)</a>
<a href="report.pdf" target="_blank">Open in a new tab</a>
```

Absolute and relative URLs

An **absolute URL** is a complete address, including the protocol and domain name, such as `https://www.bbc.co.uk/news`. Use absolute URLs to link to other websites. A **relative URL** gives a path relative to the current page, such as `contact.html` or `images/logo.png`, and is used to link between pages of the same site.

EXAM TIP “Give one advantage of relative URLs” is a classic two-mark question. Answer: *relative links keep working if the whole site is moved to a different domain or folder, because they do not contain the domain name.*

Lists

HTML offers two main kinds of list. An **unordered list** `<ul>` displays its items with bullet points and is used when the order does not matter. An **ordered list** `<ol>` numbers its items automatically and is used for sequences such as instructions or rankings. In both cases, each item is wrapped in a `<li>` (list item) element — and only `<li>` elements may sit directly inside a list.

```
<ul>                                <ol>
  <li>HTML</li>                      <li>Boil water</li>
  <li>CSS</li>                       <li>Add teabag</li>
  <li>JavaScript</li>               <li>Pour and wait</li>
</ul>                               </ol>
```

KEY FACT Putting text directly inside `<ul>` or `<ol>` without `<li>` tags is invalid and loses marks. The pattern is always list element outside, list items inside.

HTML

Task 2: Add Lists to morse.html

Add lists Morse HTML. Modify your html to include

- ordered list, unordered list, bold, Italics
- Link to Wikipedia page on morse code
- Your webpage should look like the image below

Morse Code

Morse code is a system used to send messages using dots and dashes.

You can learn more about Morse code on [Wikipedia](#).

What is Morse Code?

Morse code uses short signals called **dots** and long signals called **dashes**.

Each letter of the alphabet has its own pattern of dots and dashes.

How Morse Code Can Be Sent

Morse code can be sent using:

- Sound
- Light
- Electrical signals

For example, a *short flash of light* can mean a dot, and a *long flash of light* can mean a dash.

Example

The letter A is written as **dot dash**.

The emergency signal SOS is written as *dot dot dot, dash dash dash, dot dot dot*.

Steps to Send Morse Code Using Light

1. Choose a short message.
2. Convert each letter into dots and dashes.
3. Use a short flash for a dot.
4. Use a long flash for a dash.
5. Pause between each letter.

Why Morse Code Was Useful

Morse code was useful because it allowed people to send messages over **long distances**.

It was used by sailors, telegraph operators and *emergency services*.

Morse code is still interesting to learn today because it shows how information can be sent using simple signals.

ANSWER

```
<!DOCTYPE html>
<html>
<head>
  <title>Morse Code</title>
</head>
<body>

  <h1>Morse Code</h1>
```

<p>Morse code is a system used to send messages using dots and dashes.</p>

<p>

You can learn more about Morse code on

Wikipedia.

</p>

<h2>What is Morse Code?</h2>

<p>Morse code uses short signals called dots and long signals called dashes.</p>

<p>Each letter of the alphabet has its own pattern of dots and dashes.</p>

<h2>How Morse Code Can Be Sent</h2>

<p>Morse code can be sent using:</p>

Sound

Light

Electrical signals

<p>For example, a short flash of light can mean a dot, and a long flash of light can mean a dash.</p>

<h3>Example</h3>

<p>The letter A is written as dot dash.</p>

<p>The emergency signal SOS is written as dot dot dot, dash dash dash, dot dot dot.</p>

<h3>Steps to Send Morse Code Using Light</h3>

Choose a short message.

Convert each letter into dots and dashes.

Use a short flash for a dot.

Use a long flash for a dash.

Pause between each letter.

<hr>

HTML

```
<h2>Why Morse Code Was Useful</h2>
```

```
<p>Morse code was useful because it allowed people to send messages over <strong>long
distances</strong>.</p>
```

```
<p>It was used by sailors, telegraph operators and <em>emergency services</em>.</p>
```

```
<br>
```

```
<p>Morse code is still interesting to learn today because it shows how information can be sent
using simple signals.</p>
```

```
</body>
```

```
</html>
```

Tables

Tables present data in rows and columns. The structure is rows-first: a **<table>** contains **<tr>** (table row) elements, and each row contains cells — either **<th>** (table header, displayed bold and centred) or **<td>** (table data).

```
<table>
  <tr>
    <th>Name</th>
    <th>Score</th>
  </tr>
  <tr>
    <td>Amara</td>
    <td>92</td>
  </tr>
  <tr>
    <td>Ben</td>
    <td>87</td>
  </tr>
</table>
```

This produces a table with a header row (Name, Score) and two data rows. Note that there is no “column” element in HTML — columns emerge from the cells lining up across the rows, so every row should contain the same number of cells.

A final point of good practice: tables are for tabular data. In the early days of the web, tables were often used to lay out whole pages; that approach is now obsolete, because page layout is the job of CSS.

HTML

Task 3A: Add A Table

Add a table to your Morse.html. Your webpage needs to look like the image on below.

Morse Code Table

Letter	Morse Code	Meaning
A	dot dash	A common example of Morse code
S	dot dot dot	Used in the SOS signal
O	dash dash dash	Used in the SOS signal

Morse Code

Morse code is a system used to send messages using dots and dashes.

You can learn more about Morse code on [Wikipedia](#).

What is Morse Code?

Morse code uses short signals called **dots** and long signals called **dashes**.

Each letter of the alphabet has its own pattern of dots and dashes.

How Morse Code Can Be Sent

Morse code can be sent using:

- Sound
- Light
- Electrical signals

For example, a *short flash of light* can mean a dot, and a *long flash of light* can mean a dash.

Example

The letter A is written as **dot dash**.

The emergency signal SOS is written as *dot dot dot, dash dash dash, dot dot dot*.

Morse Code Table

Letter	Morse Code	Meaning
A	dot dash	A common example of Morse code
S	dot dot dot	Used in the SOS signal
O	dash dash dash	Used in the SOS signal

Steps to Send Morse Code Using Light

1. Choose a short message.
2. Convert each letter into dots and dashes.
3. Use a short flash for a dot.
4. Use a long flash for a dash.
5. Pause between each letter.

HTML

ANSWER

```
<!DOCTYPE html>
<html>
<head>
  <title>Morse Code</title>
</head>
<body>

  <h1>Morse Code</h1>

  <p><strong>Morse code</strong> is a system used to send messages using dots and dashes.</p>

  <p>
    You can learn more about Morse code on
    <a href="https://en.wikipedia.org/wiki/Morse_code" target="_blank">Wikipedia</a>.
  </p>

  <h2>What is Morse Code?</h2>

  <p>Morse code uses short signals called <strong>dots</strong> and long signals called
  <strong>dashes</strong>.</p>

  <p>Each letter of the alphabet has its own pattern of dots and dashes.</p>

  <h2>How Morse Code Can Be Sent</h2>

  <p>Morse code can be sent using:</p>

  <ul>
    <li>Sound</li>
    <li>Light</li>
    <li>Electrical signals</li>
  </ul>

  <p>For example, a <em>short flash of light</em> can mean a dot, and a <em>long flash of light</em> can
  mean a dash.</p>

  <h3>Example</h3>

  <p>The letter A is written as <strong>dot dash</strong>.</p>

  <p>The emergency signal SOS is written as <em>dot dot dot, dash dash dash, dot dot dot</em>.</p>
```

<h3>Morse Code Table</h3>

```
<table border="1">
  <tr>
    <th>Letter</th>
    <th>Morse Code</th>
    <th>Meaning</th>
  </tr>
  <tr>
    <td>A</td>
    <td>dot dash</td>
    <td>A common example of Morse code</td>
  </tr>
  <tr>
    <td>S</td>
    <td>dot dot dot</td>
    <td>Used in the SOS signal</td>
  </tr>
  <tr>
    <td>O</td>
    <td>dash dash dash</td>
    <td>Used in the SOS signal</td>
  </tr>
</table>
```

<h3>Steps to Send Morse Code Using Light</h3>

```
<ol>
  <li>Choose a short message.</li>
  <li>Convert each letter into dots and dashes.</li>
  <li>Use a short flash for a dot.</li>
  <li>Use a long flash for a dash.</li>
  <li>Pause between each letter.</li>
</ol>
```

<hr>

<h2>Why Morse Code Was Useful</h2>

<p>Morse code was useful because it allowed people to send messages over long distances.</p>

<p>It was used by sailors, telegraph operators and emergency services.</p>

<p>Morse code is still interesting to learn today because it shows how information can be sent using simple signals.</p>

</body>

</html>

HTML

Task 3B: Add a Morse Code Table HTML

Create another html document.

Name it morsetable.html

Create this table on the right side, alongside the elements on the web page.

On your morse.html, add a link to morsetable.html

Morse Code

Decode by looking up each pattern one letter at a time.

Letter	Morse Code	Letter	Morse Code
A	.-	N	-.
B	-...	O	---
C	-. .	P	.-.
D	-..	Q	--.
E	.	R	.-.
F	..-.	S	...
G	--.	T	-
H		U	..-
I	..	V	...-
J	.-...	W	.-.
K	-. -	X	-..-
L	.-..	Y	-. --
M	--	Z	--..

Decode this message

.... / . / .-. / .-. / ---

Your answer:

Check

HTML

ANSWER

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Decode the Morse Code Message</title>
</head>
<body>
  <h1>Morse Code</h1>
  <p>Decode by looking up each pattern one letter at a time.</p>
  <table border="1">
    <tr>
      <th>Letter</th>
      <th>Morse Code</th>
      <th>Letter</th>
      <th>Morse Code</th>
    </tr>
    <tr><td>A</td><td>.-</td><td>N</td><td>-.</td></tr>
    <tr><td>B</td><td>-...</td><td>O</td><td>---</td></tr>
    <tr><td>C</td><td>-.-.</td><td>P</td><td>...-</td></tr>
    <tr><td>D</td><td>-..</td><td>Q</td><td>--.</td></tr>
    <tr><td>E</td><td>.</td><td>R</td><td>.-.</td></tr>
    <tr><td>F</td><td>...-</td><td>S</td><td>...</td></tr>
    <tr><td>G</td><td>--.</td><td>T</td><td>-</td></tr>
    <tr><td>H</td><td>....</td><td>U</td><td>..-</td></tr>
    <tr><td>I</td><td>..</td><td>V</td><td>...-</td></tr>
    <tr><td>J</td><td>.-.-</td><td>W</td><td>.--</td></tr>
    <tr><td>K</td><td>-.-</td><td>X</td><td>-..</td></tr>
    <tr><td>L</td><td>.-..</td><td>Y</td><td>-...</td></tr>
    <tr><td>M</td><td>--</td><td>Z</td><td>--..</td></tr>
  </table>
  <h2>Decode this message</h2>
  <p>.... / . / .-.. / .-.. / ---</p>
  <form action="check.php" method="get">
    <p>
      <label for="answer">Your answer:</label>
      <input type="text" id="answer" name="answer">
      <input type="submit" value="Check">
    </p>
  </form>
</body>
</html>
```

HTML

Images

Images are embedded with the void element `<img>`, which has no content and no closing tag. Two attributes matter most: `src` gives the URL or file path of the image, and `alt` gives a text alternative.

```

```

Attribute	Purpose
<code>src</code>	URL or path of the image file — required
<code>alt</code>	Text alternative, read aloud by screen readers and displayed if the image fails to load
<code>width / height</code>	Size in pixels (in practice, better handled with CSS)

The `alt` attribute deserves special attention because accessibility is a named theme in the specification. Alt text allows visually-impaired users to understand images through a screen reader, provides a fallback if the image cannot be loaded, and helps search engines index the page.

EXAM TIP Mark schemes frequently award a mark specifically for including a sensible `alt` attribute. Make writing it a habit, even when the question does not mention it.

Task 4: Add Image

Add images to your Morse.html

Find 3 images on the web. Download the images into the same folder as your morse.html.



HTML

ANSWER

```
<p>This webpage shows an image of a Telegraph Key.</p>
```

```

```

```
<p>This webpage shows an image of a Flashlight.</p>
```

```

```

```
<p>This webpage shows an image of a Buzzer.</p>
```

```

```

TASK 5

Create a new file called morse-table.html and build a page with the following:

1. Set up a basic HTML page with the title "Decode the Morse Code Message".
2. Add a main heading: **Morse Code**
3. Add a paragraph: "Decode by looking up each pattern one letter at a time."
4. Create a **table** with visible borders and four columns: Letter, Morse Code, Letter, Morse Code.
 - The top row must use **header cells** (<th>), not ordinary cells.
 - Fill in all 26 letters — A to M down the left pair of columns, N to Z down the right (use the Morse chart on the board/handout).

Letter	Morse Code	Letter	Morse Code
A	.-	N	-.
B	-...	O	---
C	-. -.	P	.-. .
D	-..	Q	--. -
E	.	R	.-.
F	..-.	S	...
G	--.	T	-
H		U	..-
I	..	V	...-
J	.---	W	.-.
K	-. -	X	-..-
L	.-..	Y	-. --
M	--	Z	--..

5. Below the table, add a smaller heading: **Decode this message**

HTML

6. Add a paragraph containing this code: / . / .-. / .-. / ---
7. Add a small form with a text box named answer and a submit button labelled "Check", so you can type in the decoded word.
8. Add the link to the main html (morse.html)

Morse Code

Decode by looking up each pattern one letter at a time.

Letter	Morse Code	Letter	Morse Code
A	.-	N	-.
B	-...	O	---
C	-.-.	P	.-.
D	-..	Q	---.
E	.	R	.-.
F	..-.	S	...
G	--.	T	-
H		U	..-
I	..	V	...-
J	.-.-	W	.-.
K	-. -	X	-..-
L	.-..	Y	-.--
M	--	Z	--..

Decode this message

.... / . / .-. / .-. / ---

Your answer:

ANSWER

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Decode the Morse Code Message</title>
</head>
<body>
```

HTML

```
<h1>Morse Code</h1>
<p>Decode by looking up each pattern one letter at a time.</p>

<table border="1">
  <tr>
    <th>Letter</th>
    <th>Morse Code</th>
    <th>Letter</th>
    <th>Morse Code</th>
  </tr>
  <tr><td>A</td><td>.-</td><td>N</td><td>-.</td></tr>
  <tr><td>B</td><td>-...</td><td>O</td><td>---</td></tr>
  <tr><td>C</td><td>-.-.</td><td>P</td><td>...</td></tr>
  <tr><td>D</td><td>-..</td><td>Q</td><td>--.</td></tr>
  <tr><td>E</td><td>.</td><td>R</td><td>.-</td></tr>
  <tr><td>F</td><td>..-</td><td>S</td><td>...</td></tr>
  <tr><td>G</td><td>--.</td><td>T</td><td>-</td></tr>
  <tr><td>H</td><td>....</td><td>U</td><td>..-</td></tr>
  <tr><td>I</td><td>..</td><td>V</td><td>...-</td></tr>
  <tr><td>J</td><td>.-.-</td><td>W</td><td>.--</td></tr>
  <tr><td>K</td><td>-.-</td><td>X</td><td>-..</td></tr>
  <tr><td>L</td><td>.-.</td><td>Y</td><td>-.-</td></tr>
  <tr><td>M</td><td>--</td><td>Z</td><td>--.</td></tr>
</table>

<h2>Decode this message</h2>
<p>.... / . / .-.. / -.. / ---</p>

<form action="check.php" method="get">
  <p>
    <label for="answer">Your answer:</label>
    <input type="text" id="answer" name="answer">
    <input type="submit" value="Check">
  </p>
</form>

</body>
</html>
```

HTML

Forms

Forms are how web pages collect input from the user — search boxes, login screens and checkouts are all forms. A form is created with the `<form>` element, which needs two attributes: **action**, the URL that the data is sent to when the form is submitted, and **method**, which is either `get` (data appended to the URL) or `post` (data carried in the body of the request — the usual choice for anything private).

```
<form action="signup.php" method="post">
  <label for="user">Username:</label>
  <input type="text" id="user" name="user">

  <label for="pw">Password:</label>
  <input type="password" id="pw" name="pw">

  <input type="submit" value="Sign up">
</form>
```

Sign Up

Username:

Password:

Sign up

The input element

Most controls inside a form are made with the void element `<input>`. Its **type** attribute decides what kind of control appears: `text`, `password`, `email`, `number`, `checkbox`, `radio` and `submit` are the ones to know. Every input that sends data must also have a **name** attribute — this is the label attached to the data when it reaches the server, and without it the value is simply not sent.

Labels

Each input should have a `<label>` describing it. The label's **for** attribute matches the input's **id**, which links the two: clicking the label focuses the input, and screen readers announce the label when the user reaches the field.

EXAM TIP Writing a small form from a specification is the single most common HTML exam task.

Checklist: form with action and method; every input has a type and a name; the submit button's text is set with value; and `<input>` never has a closing tag.

TASK 6

Morse Code Translator — Build the Form

Create a new file called morseform.html and build a form with the following:

1. Set up a basic HTML page with the title "Morse Code Form".
2. Add a heading: **Morse Code Translator**
3. Create a form that sends data to translate.php using the **GET** method.
4. Inside the form, add:
 - A **text box** named message, with the label "Your message:"
 - Two **radio buttons** named direction: "Text to Morse" and "Morse to Text" — only one should be selectable at a time
 - A **drop-down list** named speed with three options: Slow, Standard, Fast
 - A **checkbox** labelled "Play the beeps"
 - A **submit button** labelled "Translate"

Morse Code Translator

Your message:

Direction:

- ☒ Text to Morse
☐ Morse to Text

Sending speed:

☐ Play the beeps

HTML

ANSWER

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Morse Code Form</title>
</head>
<body>

  <h1>Morse Code Translator</h1>
  <form action="translate.php" method="get">

    <p>
      <label for="message">Your message:</label><br>
      <input type="text" id="message" name="message">
    </p>

    <p>Direction:</p>
    <p>
      <input type="radio" id="encode" name="direction" value="encode" checked>
      <label for="encode">Text to Morse</label><br>
      <input type="radio" id="decode" name="direction" value="decode">
      <label for="decode">Morse to Text</label>
    </p>

    <p>
      <label for="speed">Sending speed:</label>
      <select id="speed" name="speed">
        <option value="slow">Slow</option>
        <option value="standard" selected>Standard</option>
        <option value="fast">Fast</option>
      </select>
    </p>

    <p>
      <input type="checkbox" id="sound" name="sound">
      <label for="sound">Play the beeps</label>
    </p>

    <p>
      <input type="submit" value="Translate">
    </p>
  </form>

</body>
</html>
```