

HTML Reference

Standalone HTML references for structuring text, links, images, lists, tables, forms and embedded content.

- [HTML Text and Headings](#)
- [HTML Links and Images](#)
- [HTML Lists and Tables](#)
- [HTML Forms](#)
- [HTML Embeds and Comments](#)

HTML Text and Headings

HTML provides elements for displaying text, headings and page structure.

Headings Headings are used to organise content on a webpage.

HTML provides six heading levels.

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

Result:

Heading 1 Heading 2 Heading 3 Heading 4 Heading 5 Heading 6 Paragraphs Paragraphs are used for normal text content.

This is a paragraph.

This is another paragraph.

Result:

This is a paragraph.

This is another paragraph.

Bold Text Use the strong element to make text bold.

Important information Result:

Important information

Italic Text Use the em element to emphasise text.

Important information Result:

Important information

Line Breaks Use the br element to move text onto a new line.

Line One

Line Two Result:

Line One Line Two

Horizontal Lines Use the hr element to create a horizontal divider.

Result:

Subscript Subscript is commonly used in science and mathematics.

H₂O Result:

H2O

Superscript Superscript is commonly used for powers and exponents.

x² Result:

x2

Combining Elements Multiple text elements can be combined.

Science Report

Water is written as H₂O.

The formula for area is x².

Result:

Complete Example

<h1>Heading 1</h1>

```
<h2>Heading 2</h2>
```

```
<p>This is a paragraph.</p>
```

```
<strong>Bold text</strong>
```

```
<br><br>
```

```
<em>Italic text</em>
```

```
<hr>
```

```
H<sub>2</sub>O
```

```
<br>
```

```
x<sup>2</sup>
```

Quick Reference Headings

Heading 1

Heading 6

Paragraph

Paragraph text

Bold **Bold text** Italic *Italic text* Line Break

Horizontal Rule

Subscript

Superscript

You now know how to create headings and display text content in HTML.

HTML Links and Images

Links support navigation and images communicate visual information. Both need meaningful text alternatives.

Links

```
<a href="results.php">View tournament results</a>
```

Link text should describe the destination without relying on “click here”.

For an external link opened in a new tab:

```
<a href="https://example.com"
  target="_blank"
  rel="noopener">Visit the example website</a>
```

Only open a new tab when useful, and make the behaviour clear in surrounding text.

Email and page sections

```
<a href="mailto:help@example.com">Email fictional support</a>
<a href="#testing">Jump to testing evidence</a>

<h2 id="testing">Testing evidence</h2>
```

Use `example.com` addresses in demonstrations rather than real school or personal details.

Images

```

```

Alternative text communicates the image's purpose. A decorative image uses an empty alternative:

```

```

Do not put essential words only inside an image. Avoid using width and height attributes to distort the aspect ratio; use responsive CSS.

Figures

```
<figure>
  
  <figcaption>Example leaderboard produced from fictional data.</figcaption>
</figure>
```

Check

- ☐ Link text is meaningful.
- ☐ External new tabs use `rel="noopener"`.
- ☐ Demonstrations use fictional addresses and data.
- ☐ Informative images have purposeful alternative text.
- ☐ Decorative images use `alt=""`.
- ☐ Images resize without distortion.

HTML Lists and Tables

Lists organise related items. Tables represent information with meaningful row and column relationships.

Unordered and ordered lists

```
<ul>
  <li>Team registration</li>
  <li>Match results</li>
  <li>Leaderboard</li>
</ul>

<ol>
  <li>Choose a fictional dataset.</li>
  <li>Validate its headings.</li>
  <li>Import valid rows.</li>
</ol>
```

Use an ordered list when sequence matters. Do not create visual bullets with hyphens and line breaks.

Description list

```
<dl>
  <dt>Administrator</dt>
  <dd>Imports and manages fictional tournament data.</dd>
  <dt>User</dt>
  <dd>Views processed results and statistics.</dd>
</dl>
```

Accessible table

```
<div class="table-wrap">
<table>
  <caption>Fictional team standings</caption>
  <thead>
    <tr>
      <th scope="col">Team</th>
      <th scope="col">Wins</th>
      <th scope="col">Points</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">Falcons</th>
      <td>4</td>
      <td>12</td>
    </tr>
  </tbody>
</table>
</div>
```

The caption identifies the table. Header cells and `scope` communicate relationships to assistive technology.

Do not use tables for page layout. For narrow screens, keep the semantic table and place it inside a horizontally scrollable wrapper.

Check

- ☐ List type matches the content.
- ☐ Table has a caption.
- ☐ Header cells use suitable scope.
- ☐ Values remain understandable without colour.
- ☐ Layout tables are avoided.
- ☐ Screenshots have meaningful alternative text or are marked decorative.

HTML Forms

Forms collect user input. Accessible forms connect every control to a visible label and provide instructions and errors that do not rely on placeholders or colour.

Basic form

```
<form action="register.php" method="post">
  <div class="form-group">
    <label for="username">Username</label>
    <p id="username-help">Use 3–50 letters, numbers or underscores.</p>
    <input id="username" name="username"
      autocomplete="username"
      aria-describedby="username-help"
      maxlength="50" required>
  </div>

  <div class="form-group">
    <label for="password">Password</label>
    <input id="password" name="password"
      type="password"
      autocomplete="new-password"
      required>
  </div>

  <button type="submit">Create account</button>
</form>
```

The label's `for` value matches the input's `id`. The `name` identifies the value submitted to PHP.

Useful input types

```
<input type="email" name="email">
<input type="number" name="capacity" min="1" max="100">
```

```
<input type="date" name="event_date">
<input type="file" name="dataset" accept=".csv,text/csv">
```

Input types and HTML constraints help users, but PHP must validate every submitted value again.

Related choices

```
<fieldset>
  <legend>Preferred output</legend>
  <label><input type="radio" name="output" value="table"> Table</label>
  <label><input type="radio" name="output" value="chart"> Chart</label>
</fieldset>
```

Use `fieldset` and `legend` for a related group.

Error summary

```
<div role="alert">
  <p>Please correct the following:</p>
  <ul>
    <li>Capacity must be from 1 to 100.</li>
  </ul>
</div>
```

Preserve valid values after an error when safe, and never return passwords to the form.

GET or POST

Use GET for non-sensitive searches and filters that may be bookmarked. Use POST for registration, login, file upload and operations that change stored data.

Check

- ☐ Every control has an associated label.

- ☐ Instructions remain visible.
- ☐ Appropriate input types and autocomplete values are used.
- ☐ PHP repeats validation.
- ☐ Errors are specific and accessible.
- ☐ Forms contain only fictional data.

HTML Embeds and Comments

HTML can be used to embed videos, maps and other content into a webpage.

Comments can also be added to help explain code without displaying anything on the page.

HTML Comments

Comments are ignored by the browser.

They are useful for explaining code and leaving notes for yourself or other developers.

```
<!-- This is a comment -->
```

Example:

```
<!-- Navigation Menu -->

<nav>

</nav>
```

Comments are not visible on the webpage.

Embedding a YouTube Video

YouTube provides embed code for videos.

Example:

```
<iframe
  width="560"
  height="315"
  src="https://www.youtube.com/embed/W-Q7RMpINVo"
  allowfullscreen>

</iframe>
```

Result:

A YouTube video displayed on the page.

<https://www.youtube.com/embed/dQw4w9WgXcQ>

Responsive YouTube Videos

To help videos work on phones and tablets:

```
<iframe
  width="100%"
  height="315"
  src="https://www.youtube.com/embed/W-Q7RMpINVo"
  allowfullscreen>

</iframe>
```

The video will automatically adjust to the available width.

Embedding a Google Map

Google Maps can also be embedded.

Example:

```
<iframe
  src="https://www.google.com/maps/embed..."
```

```
width="600"  
height="450">
```

```
</iframe>
```

Result:

An interactive map displayed on the page.

<https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3520.366280272018!2d153.37404667577476!3d-28.07436817597404!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x6b911cb972c15b2f%3A0xc34ef94511e30812!2sRobina%20State%20High%20School!5e0!3m2!1sen!2sau!4v1781436815960!5m2!1sen!2sau>

Displaying a Video File

Videos stored in your project can be displayed using the `video` element.

```
<video  
  width="600"  
  controls>  
  
  <source  
    src="videos/demo.mp4"  
    type="video/mp4">  
  
</video>
```

Result:

A video player with play and pause controls.

Displaying Audio Files

Audio can be embedded using the `audio` element.

```
<audio controls>

  <source
    src="audio/music.mp3"
    type="audio/mpeg">

</audio>
```

Result:

An audio player with playback controls.

Displaying PDF Files

PDF documents can be embedded directly into a webpage.

```
<iframe
  src="files/document.pdf"
  width="100%"
  height="600">

</iframe>
```

Result:

The PDF document is displayed inside the webpage.

Common Uses for Embeds

Embeds are commonly used for:

- YouTube videos
- Google Maps
- PDF documents
- Audio players

- Demonstration videos
 - Interactive content
-

Complete Example

```
<!DOCTYPE html>

<html>

<head>
  <title>Embeds Example</title>
</head>

<body>

  <h1>Media Examples</h1>

  <!-- YouTube Video -->

  <iframe
    width="560"
    height="315"
    src="https://www.youtube.com/embed/W-Q7RMpINVo"
    allowfullscreen>

  </iframe>

  <br><br>

  <!-- Audio Player -->

  <audio controls>

    <source
      src="audio/music.mp3"
      type="audio/mpeg">
```

```
</audio>
```

```
</body>
```

```
</html>
```

Quick Reference

Comment

```
<!-- Comment -->
```

YouTube Video

```
<iframe>
```

Google Map

```
<iframe>
```

Video File

```
<video>
```

Audio File

```
<audio>
```

PDF

```
<iframe src="document.pdf">
```

You now know how to embed media and add comments in HTML.

Next chapter: **CSS Reference**