

HTML Cheat Sheet for Beginners

Essential Tags, Tips, and Examples



Basic HTML Structure

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Page Title</title>
  </head>
  <body>
    Content goes here...
  </body>
</html>
```

This is the skeleton of every HTML page.



Links

```
<a href="https://example.com">Visit Site</a>
<a href="mailto:example@example.com?subject=My Subject">Send us an email</a>
```

Creates a clickable hyperlink.

- ◆ The first link opens a webpage.
- ◆ A "mailto" link opens your email client to send an email with the defined subject.



Images

```

```

Displays an image.

- ◆ "alt" provides a text alternative for screen readers and SEO.
- ◆ Use loading="lazy" for better performance on scroll-heavy pages.
- ◆ Width and height 'reserve' a space for the image so the content doesn't move once the image loads

Headings

```
<h1>Main Heading</h1>
<h2>Subheading</h2>
...
<h6>Smallest Heading</h6>
```

- ◆ Use only one <h1> per page for SEO, then structure with <h2>, <h3>, etc.

Text Formatting

```
<p>Paragraph</p>
<br>Line Break</br>
<strong>Bold Text</strong>
<em>Italic Text</em>
```

- ◆ Paragraph has spacing before/after
- ◆ Line break is a new line with no spacing before or after

Text Style

```
<span style="color:red;">Red text</span>
<span style="font-size:16px;">Text size</span>
<span style="font-size:16px;color:red;">Text colour and size</span>
  <div style="margin-left:15px;">Text size</span>
```

- ◆ Many types of style can be added to an element
- ◆ Style can be added to many types of elements. Try it!

Lists

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
</ul>
<ol>
  <li>Item 1</li>
</ol>
```

- ◆ creates an unordered (bulleted) list.
- ◆ creates an ordered (numbered or lettered) list.



Divs and Spans

```
<div>Block element</div>
<span>Inline element</span>
```

- ◆ `<div>` is used for layout blocks - creating a 'break' in the content.
- ◆ `` is used for inline styling or wrapping small parts - no break.



SEO Tags

```
<title>Page Title</title>
<meta name="description" content="Short page description">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<h1>Main Keyword Heading</h1>
```

Use these for better search engine visibility.

- ◆ Title and description show in search results - place in head section.
- ◆ Viewport ensures mobile responsiveness, a factor in SEO - place in head section.
- ◆ Only one `<h1>` per page - place in body section.



Social Media Tags

```
<meta property="og:title" content="Page Title">
<meta property="og:description" content="Shown in social shares">
<meta property="og:image" content="https://example.com/image.jpg">
<meta property="og:url" content="https://example.com/page">
```

Use Open Graph tags to control how your content appears on social platforms like Facebook and LinkedIn.

- ◆ There are more tags than this, but these are the basic



Preloading CSS

```
<link rel="preload" href="style.css" as="style">
<link rel="stylesheet" href="style.css">
```

- ◆ The "preload" tells your browser to start downloading the stylesheet, even while downloading other resources.

Forms

```
<form action="/submit" method="post">
  <input type="text" name="name">
  <input type="submit" value="Send">
</form>
```

Used to collect user input.

- ◆ "action" sets where data is sent - typically a server-side form, or connecting to an external service.

Comments

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

You can't see comments on the page, but they are useful for documentation.

- ◆ Use comments to explain complex code or mark sections of your page.
- ◆ They help you (and others) understand your code when revisiting it later.
- ◆ Comments do not affect how the page displays in the browser.
- ◆ Don't include sensitive information (like passwords) in comments – they're visible in page source.

Semantic HTML

Semantic HTML provides structure and meaning to your web pages by using elements that reflect the purpose of their content. It allows you to lay out a page into logical sections—such as headers, navigation, main content, and footers—making your code more readable and easier to maintain.

```
<header>Site or section header</header>
<nav>Navigation links</nav>
<main>Main page content</main>
<section>Related content grouped together</section>
<article>Independent content like blog posts</article>
<aside>Sidebar or related info</aside>
<footer>Bottom of page or section</footer>
```

Semantic HTML tags give meaning to your content.

- ◆ They improve SEO and accessibility.
- ◆ Use them to create a logical structure for your site.
- ◆ Help screen readers and search engines understand page hierarchy.

Accessibility with the `role` Attribute

The `role` attribute enhances accessibility by describing the purpose of elements for assistive technologies like screen readers. It is especially helpful when using non-semantic elements such as `<div>` or `` to indicate their function. The following describes the element as navigation, which improves both usability and SEO.

◆ Example: Using `role` for Navigation

```
<div role="navigation">
  <ul>
    <li><a href="#home">Home</a></li>
    <li><a href="#about">About</a></li>
    <li><a href="#contact">Contact</a></li>
  </ul>
</div>
```

◆ Other Common `role` Examples

- `role="navigation"`: Defines a section that contains navigation links.
- `role="main"`: Defines the main content of the page.
- `role="banner"`: Defines the header section, similar to the `<header>` tag.
- `role="complementary"`: Indicates content that complements the main content, like sidebars.

Tables

```
<table>
  <caption>Monthly Expenses</caption>
  <thead>
    <tr>
      <th scope="col">Category</th>
      <th scope="col">Amount</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Rent</td>
      <td>$1000</td>
    </tr>
  </tbody>
</table>
```

Tables display **tabular data**—data arranged in rows and columns.

- ◆ Use `<table>` when content has a natural row/column relationship
- ◆ Avoid using tables for layout—use CSS and `<div>`s for that
- ◆ Always include `<thead>`, `<tbody>`, `<th>`, and `scope` attributes for accessibility
- ◆ Use `<caption>` to describe the table's purpose

Structured Data (Schema)

Structured data helps search engines understand your content better, leading to enhanced search results like rich snippets. Schema.org is a popular way to mark up content using a standard vocabulary that search engines like Google can easily read.

By adding JSON-LD markup to your pages, you provide detailed information about your content—whether it's a local business, product, event, or article—making it more discoverable and relevant.

Using Schema can improve SEO and visibility in search results, and it's an easy way to boost your site's presence without altering visible content.

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "LocalBusiness",
  "name": "Joe's Coffee",
  "image": "https://example.com/images/joes-coffee.jpg",
  "description": "Local café serving coffee, tea, and fresh pastries in Melbourne.",
  "address": {
    "@type": "PostalAddress",
    "streetAddress": "123 Laneway St",
    "addressLocality": "Melbourne",
    "addressRegion": "VIC",
    "postalCode": "3000",
    "addressCountry": "AU"
  },
  "url": "https://joescoffee.com.au",
  "telephone": "+61 3 9123 4567",
  "openingHours": "Mo-Fr 07:00-16:00",
  "priceRange": "$$"
}
</script>
```

Structured data (also known as Schema) helps search engines understand your content.

- ◆ Use inside <head> or before </body>.
- ◆ It improves SEO and may enhance your search result with rich snippets.
- ◆ Use for local businesses, products, articles, reviews, events, and more.

The @type in structured data depends on the type of website or business you're marking up, and the fields included in the schema vary based on the chosen @type. There are many different forms of structured data, such as those for local businesses, images, reviews, and business hours.

Learn more: <https://schema.org/docs/schemas.html>

Tips for Building Cleaner, Optimised Websites

Web Structure & Accessibility

- 1. Use Semantic HTML Tags:** Use meaningful tags like `<header>`, `<footer>`, `<article>` to improve your site's structure and accessibility. Avoid excessive `<div>` tags.
- 2. Accessibility Best Practices:** Add `alt` text to images and use ARIA roles (e.g., `role="navigation"`) for screen readers. This ensures your website is accessible to users with disabilities.

Performance Optimisation

- 3. Optimise Your Images:** Keep images under 100KB for large images and 10KB for small ones. Use `loading="lazy"` for images that are not immediately visible to users to improve page load times. Optimising images can drastically reduce the overall size of your pages and improve user experience.
- 4. Minify Resources:** Minify HTML, CSS, and JavaScript to reduce page size and enable GZIP compression for faster loading. This helps improve site performance, especially on slower connections.
- 5. Preload Important Resources:** Use `<link rel="preload">` to load critical resources like stylesheets early, improving load times for important elements.

SEO Best Practices

- 6. Improve SEO with Meta Tags:** Use `<meta name="description">` and `<title>` tags to improve search visibility. Maintain a clear heading hierarchy (`<h1>`, `<h2>`, etc.) for better crawling.

Mobile Optimisation

- 7. Mobile-Friendly Design:** Use the `<meta name="viewport">` tag to ensure the site adapts to different screen sizes, improving the mobile experience.

Code Quality

- 8. Clean, Readable Code:** Maintain clean, indented code with comments where necessary. This makes your code easier to maintain and scale.

Structured Data

- 9. Use Structured Data:** Add JSON-LD or Microdata to help search engines understand your content better, improving visibility in search results.

Bonus Tip

- 10. Test Website Performance Regularly:** Use tools like Google PageSpeed Insights or Lighthouse to analyse your website's performance. Regular testing allows you to make data-driven improvements that can increase loading speed, accessibility, and overall user experience.

☀ Complete Example: Putting It All Together

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="description" content="HTML, SEO, and performance optimisation tips">
  <title>HTML Cheat Sheet Example</title>
  <link rel="preload" href="styles.css" as="style">
</head>
<body>

<!-- This is the main page structure using semantic HTML elements -->
<header role="banner">
  <h1>Welcome to the HTML Cheat Sheet</h1>
  <p>Your guide to cleaner, optimised websites</p>
</header>

<article>
  <h2>Understanding SEO</h2>
  <p>Search Engine Optimisation (SEO) is vital for ranking in search engines.
  By structuring your page with meaningful HTML tags and optimising content, you can
  improve your site's visibility.</p>
  <ul>
    <li>Use semantic HTML elements</li>
    <li>Include meta tags like <meta name="description"></li>
    <li>Optimise images for faster loading</li>
  </ul>
</article>
<section>
  <h3>Image Example</h3>
  
</section>

<footer role="contentinfo">
  <p><strong>&copy; 2023</strong> - Your website</p>
  <p>Find more at: <a href="https://htmlbasix.com">HTML Basics</a></p>
</footer>

<!-- Structured data contracted for brevity -->
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  ...etc...
}</script>
</body>
</html>
```

Need more help? Visit www.htmlbasix.com for tools, generators, and tutorials.