

# HTML Cheatsheet

## Essential elements and techniques for web structure and content markup

This cheatsheet provides a quick reference to fundamental HTML operations, syntax, and modern features, ideal for both beginners and experienced web developers for efficient web page structure and content organization.

<b>HTML Basics</b> Structure, elements, and attributes	<b>Text Content</b> Headings, paragraphs, and formatting	<b>Forms &amp; Input</b> User interaction and data collection
<b>Media &amp; Links</b> Images, videos, and navigation	<b>Semantic HTML</b> Modern structure and accessibility	

## HTML Document Structure

### Basic HTML Document

Every HTML document starts with a document type declaration and follows a standard structure.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Page Title</title>
</head>
<body>
  <!-- Page content goes here -->
</body>
</html>
```

### Head Elements

The head section contains metadata about the document.

```
<!-- Character encoding -->
<meta charset="UTF-8">

<!-- Viewport for responsive design -->
<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!-- Page description -->
<meta name="description" content="Page description">

<!-- Link to CSS -->
<link rel="stylesheet" href="styles.css">

<!-- Link to favicon -->
<link rel="icon" href="favicon.ico">
```

## Text Content Elements

HTML provides various elements for structuring and formatting text content effectively.

01	02	03
<b>Headings: h1 to h6</b> Define the hierarchy and importance of content sections.	<b>Paragraphs: p</b> The most common element for text content blocks.	<b>Text Formatting</b> Elements for emphasizing and styling text inline.
<pre>&lt;h1&gt;Main Title&lt;/h1&gt; &lt;h2&gt;Section Title&lt;/h2&gt; &lt;h3&gt;Subsection Title&lt;/h3&gt; &lt;h4&gt;Sub-subsection Title&lt;/h4&gt; &lt;h5&gt;Minor Heading&lt;/h5&gt; &lt;h6&gt;Smallest Heading&lt;/h6&gt;</pre>	<pre>&lt;p&gt;This is a paragraph of text. It can contain multiple sentences and will wrap automatically.&lt;/p&gt;  &lt;p&gt;This is another paragraph. Paragraphs are separated by margin space.&lt;/p&gt;</pre>	<pre>&lt;strong&gt;Strong importance (bold)&lt;/strong&gt; &lt;em&gt;Emphasis (italic)&lt;/em&gt; &lt;b&gt;Bold text&lt;/b&gt; &lt;i&gt;Italic text&lt;/i&gt; &lt;u&gt;Underlined text&lt;/u&gt; &lt;mark&gt;Highlighted text&lt;/mark&gt; &lt;small&gt;Small text&lt;/small&gt; &lt;del&gt;Deleted text&lt;/del&gt; &lt;ins&gt;Inserted text&lt;/ins&gt;</pre>

### Line Breaks & Spacing

Control text flow and spacing within content.

```
<!-- Line break -->
Line 1<br>
Line 2

<!-- Horizontal rule -->
<hr>

<!-- Preformatted text -->
<pre>
Text with
preserved  spacing
and line breaks
</pre>

<!-- Code formatting -->
<code>console.log("Hello");</code>
```

## Lists & Navigation

### Unordered Lists

Create bullet point lists for non-sequential items.

```
<ul>
  <li>First item</li>
  <li>Second item</li>
  <li>Third item</li>
</ul>

<!-- Nested lists -->
<ul>
  <li>Main item
    <ul>
      <li>Sub item 1</li>
      <li>Sub item 2</li>
    </ul>
  </li>
</ul>
```

### Ordered Lists

Create numbered lists for sequential items.

```
<ol>
  <li>First step</li>
  <li>Second step</li>
  <li>Third step</li>
</ol>

<!-- Custom numbering -->
<ol start="5">
  <li>Item 5</li>
  <li>Item 6</li>
</ol>

<!-- Different numbering types -->
<ol type="A">
  <li>Item A</li>
  <li>Item B</li>
</ol>
```

## Forms & Input Elements

Forms are essential for user interaction and data collection in web applications.

### Basic Form Structure

The foundation of user input collection.

```
<form action="/submit" method="POST">
  <label for="username">Username:</label>
  <input type="text" id="username" name="username" required>

  <label for="email">Email:</label>
  <input type="email" id="email" name="email" required>

  <input type="submit" value="Submit">
</form>
```

### Input Types

Various input types for different data collection needs.

```
<!-- Text inputs -->
<input type="text" placeholder="Enter text">
<input type="email" placeholder="email@example.com">
<input type="password" placeholder="Password">
<input type="url" placeholder="https://example.com">
<input type="tel" placeholder="+1234567890">

<!-- Number inputs -->
<input type="number" min="1" max="100" step="1">
<input type="range" min="0" max="100" value="50">

<!-- Date and time -->
<input type="date">
<input type="time">
<input type="datetime-local">
```

## Media Elements

### Images

Display images with various attributes and options.

```
<!-- Basic image -->


<!-- Responsive image -->


<!-- Image with size -->


<!-- Picture element for responsive images -->
<picture>
  <source media="(min-width: 800px)" srcset="large.jpg">
  <source media="(min-width: 400px)" srcset="medium.jpg">
  
</picture>
```

### Audio

Embed audio content with playback controls.

```
<!-- Basic audio -->
<audio controls>
  <source src="audio.mp3" type="audio/mpeg">
  <source src="audio.ogg" type="audio/ogg">
  Your browser does not support audio.
</audio>

<!-- Audio with autoplay -->
<audio controls autoplay loop>
  <source src="background.mp3" type="audio/mpeg">
</audio>
```

## Tables

Tables organize data in rows and columns for easy comparison and analysis.

### Basic Table Structure

Create structured data displays with tables.

```
<table>
  <thead>
    <tr>
      <th>Name</th>
      <th>Age</th>
      <th>City</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>John</td>
      <td>25</td>
      <td>New York</td>
    </tr>
    <tr>
      <td>Jane</td>
      <td>30</td>
      <td>London</td>
    </tr>
  </tbody>
</table>
```

## Semantic HTML5 Elements

Semantic elements provide meaning and structure for better accessibility and SEO.

### Page Structure Elements

Define the main sections of your page layout.

```
<!-- Page header -->
<header>
  <nav>
    <ul>
      <li><a href="#home">Home</a></li>
      <li><a href="#about">About</a></li>
    </ul>
  </nav>
</header>

<!-- Main content -->
<main>
  <article>
    <h1>Article Title</h1>
    <p>Article content...</p>
  </article>
</main>

<!-- Sidebar -->
<aside>
  <h2>Related Links</h2>
  <ul>
    <li><a href="#">Link 1</a></li>
  </ul>
</aside>

<!-- Page footer -->
<footer>
  <p>© 2024 Company Name</p>
</footer>
```

## HTML Attributes

### Global Attributes

Attributes that can be used on any HTML element.

```
<!-- ID for unique identification -->
<div id="unique-element">Content</div>

<!-- Class for styling and selection -->
<p class="highlight important">Text</p>

<!-- Title for tooltips -->
<span title="This is a tooltip">Hover me</span>

<!-- Data attributes -->
<div data-user-id="123" data-role="admin">User</div>

<!-- Language -->
<p lang="es">Hola mundo</p>

<!-- Direction -->
<p dir="rtl">Right to left text</p>

<!-- Hidden elements -->
<div hidden>This won't be displayed</div>
```

## HTML5 Modern Features

### New Input Features

HTML5 introduced new input types and attributes.

```
<!-- New input types -->
<input type="color" value="#ff0000">
<input type="search" placeholder="Search...">
<input type="file" accept="image/*" multiple>

<!-- Datalist for autocomplete -->
<input list="browsers" name="browser">
<datalist id="browsers">
  <option value="Chrome">
  <option value="Firefox">
</datalist>

<!-- Progress and meter -->
<progress value="70" max="100">70%</progress>
<meter value="0.6">60%</meter>
```

### Canvas & SVG

Graphics and drawing capabilities in HTML5.

```
<!-- Canvas for dynamic graphics -->
<canvas id="myCanvas" width="400" height="200">
  Your browser does not support canvas.
</canvas>

<!-- Inline SVG -->
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40"
    fill="blue" stroke="black" stroke-width="2"/>
</svg>
```

## Best Practices & Validation

### HTML Best Practices

Write clean, maintainable, and accessible HTML.

```
<!-- Always declare doctype -->
<!DOCTYPE html>

<!-- Use semantic elements -->
<header>...</header>
<main>...</main>
<footer>...</footer>

<!-- Proper nesting -->
<div>
  <p>Properly nested content</p>
</div>

<!-- Use lowercase for elements and attributes -->


<!-- Close all tags -->
<p>Always close your tags</p>

<!-- Use meaningful alt text -->

```

### HTML Validation & Debugging

Ensure your HTML is valid and accessible.

```
<!-- Use W3C HTML Validator -->
<!-- https://validator.w3.org -->

<!-- Common validation errors -->
<!-- Missing alt attributes -->
 <!-- Provide alt text -->

<!-- Unclosed tags -->
<p>Text content</p> <!-- Always close tags -->

<!-- Invalid nesting -->
<p>
  Valid paragraph content
  <!-- Don't put block elements inside paragraphs -->
</p>

<!-- Use developer tools -->
<!-- Right-click -> Inspect Element -->
<!-- Check console for errors -->
<!-- Validate accessibility with WAVE or axe -->
```

## HTML Templates & Frameworks

Modern approaches to HTML development and organization.

<b>Template Engines</b> Dynamic HTML generation with template languages. <pre>&lt;!-- Handlebars template --&gt; &lt;div&gt;   &lt;h1&gt;{{title}}&lt;/h1&gt;   {{#each items}}     &lt;p&gt;{{this}}&lt;/p&gt;   {{/each}} &lt;/div&gt;  &lt;!-- Mustache template --&gt; &lt;div&gt;   &lt;h1&gt;{{title}}&lt;/h1&gt;   {{#items}}     &lt;p&gt;{{.}}&lt;/p&gt;   {{/items}} &lt;/div&gt;</pre>	<b>Web Components</b> Reusable custom HTML elements. <pre>&lt;!-- Custom element definition --&gt; &lt;template id="my-component"&gt;   &lt;style&gt;     p { color: blue; }   &lt;/style&gt;   &lt;p&gt;&lt;slot&gt;&lt;/slot&gt;&lt;/p&gt; &lt;/template&gt;  &lt;!-- Usage --&gt; &lt;my-component&gt;Hello World&lt;/my-component&gt;  &lt;script&gt; class MyComponent extends HTML.Element {   // Component logic } customElements.define("my-component", MyComponent); &lt;/script&gt;</pre>	<b>Framework Integration</b> HTML within modern JavaScript frameworks. <pre>&lt;!-- React JSX --&gt; function Component() {   return (     &lt;div className="container"&gt;       &lt;h1&gt;{title}&lt;/h1&gt;       &lt;p&gt;Content here&lt;/p&gt;     &lt;/div&gt;   ); }  &lt;!-- Vue template --&gt; &lt;template&gt;   &lt;div class="container"&gt;     &lt;h1&gt;{{ title }}&lt;/h1&gt;     &lt;p v-if="showContent"&gt;Content here&lt;/p&gt;   &lt;/div&gt; &lt;/template&gt;</pre>
---	---	--

**Reference:** This cheatsheet covers essential HTML elements and modern practices for efficient web structure and content markup in modern web development workflows.