

HTML Cheat Sheet for Students: Forms, Tables & Important Tags (with Many Examples)

This guide explains core HTML concepts in clear English with many small, copy-pasteable examples.

1) HTML Basics

An HTML page is built with tags. Most tags come in pairs: `<tag>...</tag>`. The minimal structure:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My First Page</title>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <h1>Hello!</h1>
    <p>This is a paragraph.</p>
  </body>
</html>
```

2) Forms (Collecting User Input)

A form sends user data to a server. Important attributes of `<form>`: action (URL), method (GET/POST), autocomplete, novalidate.

2.1 Basic Form Example

```
<form action="/submit" method="POST" autocomplete="on">
  <div>
    <label for="name">Name</label>
    <input id="name" name="name" type="text" placeholder="Your full name" required>
  </div>

  <div>
    <label for="email">Email</label>
    <input id="email" name="email" type="email" placeholder="name@example.com" required>
  </div>

  <div>
    <label for="password">Password</label>
    <input id="password" name="password" type="password" minlength="6" required>
  </div>

  <button type="submit">Submit</button>
</form>
```

2.2 Useful `<input>` Types

```
<!-- Common input types -->
<input type="text">
<input type="email">
<input type="password">
```

```

<input type="number" min="0" max="100" step="5">
<input type="range" min="0" max="100" value="40">
<input type="date">
<input type="time">
<input type="datetime-local">
<input type="color">
<input type="url" placeholder="https://example.com">
<input type="tel" pattern="[0-9\-\\+\s\(\)]{10,}">
<input type="file" accept=".png,.jpg" multiple>
<input type="checkbox">
<input type="radio" name="choice">
<input type="hidden" name="token" value="abc123">

```

2.3 Labels & Accessibility

```

<label for="skill">Choose a skill</label>
<select id="skill" name="skill">
  <option>HTML</option>
  <option>CSS</option>
  <option>JavaScript</option>
</select>

<!-- Group related inputs -->
<fieldset>
  <legend>Contact Preferences</legend>
  <label><input type="checkbox" name="pref_email"> Email</label>
  <label><input type="checkbox" name="pref_sms"> SMS</label>
</fieldset>

```

Tip: Always pair `<label for="...">` with the input's id. Use `<fieldset>/<legend>` to group related fields. Screen readers love this.

2.4 Built-in Validation & Patterns

```

<form action="/register" method="POST">
  <label for="username">Username (3-12 letters/numbers)</label>
  <input id="username" name="username" required pattern="[a-zA-Z0-9]{3,12}" title="3-12 letters or numbers">

  <label for="age">Age (18-99)</label>
  <input id="age" name="age" type="number" min="18" max="99" required>

  <label for="website">Website (URL)</label>
  <input id="website" name="website" type="url" placeholder="https://example.com">

  <button type="submit">Create Account</button>
</form>

```

2.5 Autocomplete & Datalist

```

<label for="city">City</label>
<input list="cities" id="city" name="city" placeholder="Start typing...">
<datalist id="cities">
  <option value="Dhaka">
  <option value="Chattogram">
  <option value="Rajshahi">
</datalist>

```

2.6 GET vs POST

```

<!-- GET appends data to the URL (good for search forms) -->
<form action="/search" method="GET">

```

```

    <input name="q" type="search" placeholder="Search...">
    <button>Search</button>
</form>

<!-- POST sends data in the request body (good for login/register) -->
<form action="/login" method="POST">
    <input name="email" type="email" required>
    <input name="password" type="password" required>
    <button>Login</button>
</form>

```

2.7 File Upload (multipart/form-data)

```

<form action="/upload" method="POST" enctype="multipart/form-data">
    <label for="photo">Upload Photo (PNG/JPG)</label>
    <input id="photo" name="photo" type="file" accept=".png,.jpg" required>
    <button type="submit">Upload</button>
</form>

```

2.8 Helpful Attributes

```

<input type="text" placeholder="Hint only">
<input type="text" value="Prefilled">
<input type="text" readonly value="You can't edit this">
<input type="text" disabled value="Disabled field">
<input type="text" required>

```

3) Tables (Displaying Tabular Data)

Core tags: <table>, <tr>, <th>, <td>, plus <caption>, <thead>, <tbody>, <tfoot> for structure.

3.1 Basic Table

```

<table>
    <caption>Student Marks</caption>
    <thead>
        <tr>
            <th scope="col">Name</th>
            <th scope="col">Class</th>
            <th scope="col">Marks</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>Rahim</td>
            <td>10</td>
            <td>85</td>
        </tr>
        <tr>
            <td>Karima</td>
            <td>10</td>
            <td>91</td>
        </tr>
    </tbody>
    <tfoot>
        <tr>
            <td colspan="2">Average</td>
            <td>88</td>
        </tr>
    </tfoot>
</table>

```

```
</table>
```

3.2 Colspan & Rowspan

```
<table>
  <tr>
    <th rowspan="2">Product</th>
    <th colspan="2">Q1</th>
  </tr>
  <tr>
    <th>Sales</th>
    <th>Profit</th>
  </tr>
  <tr>
    <td>Widget A</td>
    <td>1200</td>
    <td>$300</td>
  </tr>
</table>
```

3.3 Simple CSS for Tables (Zebra Stripes)

```
<style>
table { border-collapse: collapse; width: 100%; }
th, td { border: 1px solid #ccc; padding: 8px; text-align: left; }
thead th { background: #f2f2f2; }
tbody tr:nth-child(odd) { background: #fafafa; }
caption { caption-side: top; font-weight: bold; padding: 6px; }
</style>
```

4) Other Important Tags

- `<a>`: hyperlink. Use `rel="noopener"` with `target="_blank"`.
- ``: images. Always provide meaningful alt text.
- `//`: lists (unordered/ordered).
- `<div>` and ``: generic containers.
- Semantic: `<header>`, `<nav>`, `<main>`, `<section>`, `<article>`, `<aside>`, `<footer>`. Improves structure and accessibility.
- `<figure>/<figcaption>`: image + caption grouping.
- `<audio>` / `<video>`: media playback with controls attribute.
- `<iframe>`: embed another page or YouTube video.
- `<meta>`: metadata (charset, viewport).
- `<link>`: connect CSS files or icons.
- `<script>`: add JavaScript (defer for performance).
- `
` and `<hr>`: line break and horizontal rule.
- Comments: `<!-- note for developers -->`

4.1 Links, Images, Lists

```
<!-- Link -->
<a href="https://example.com" target="_blank" rel="noopener">My site</a>
```

```

<!-- Image -->


<!-- Lists -->
<ol>
  <li>HTML</li>
  <li>CSS</li>
  <li>JavaScript</li>
</ol>

```

4.2 Figure & Figcaption

```

<figure>
  
  <figcaption>Figure 1: System architecture at a glance.</figcaption>
</figure>

```

4.3 Audio & Video

```

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

<video controls width="480" poster="thumb.jpg">
  <source src="movie.mp4" type="video/mp4">
  Sorry, your browser can't play this video.
</video>

```

4.4 iFrame (embed YouTube)

```

<iframe width="560" height="315"
  src="https://www.youtube.com/embed/dQw4w9WgXcQ"
  title="YouTube video player" frameborder="0"
  allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture"
  allowfullscreen></iframe>

```

4.5 Headings & Text Formatting

```

<h1>Main Title</h1>
<h2>Section Title</h2>
<h3>Subsection</h3>
<p><strong>Bold</strong>, <em>Italic</em>, <mark>Highlighted</mark></p>
<p><code>Inline code</code> and <kbd>Ctrl</kbd>+<kbd>S</kbd></p>

```

5) Bonus: Small Form + JS Validation

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Quick Validation</title>
  <script defer>
    function checkForm(e) {
      const email = document.getElementById('e');
      if (!email.value.endsWith('.com')) {
        e.preventDefault();
        alert('Email must end with .com for this demo');
      }
    }
    addEventListener('DOMContentLoaded', () => {

```

```
        document.querySelector('form').addEventListener('submit', checkForm);
    });
</script>
</head>
<body>
    <form action="/ok" method="POST">
        <label>Email <input id="e" type="email" required></label>
        <button>Send</button>
    </form>
</body>
</html>
```

6) Best Practices

- Use semantic tags for structure; they help SEO and screen readers.
- Always connect labels to inputs. Provide alt text for images.
- Prefer CSS for presentation; avoid deprecated HTML attributes.
- Validate on both client (nice UX) and server (security).
- Use defer or module scripts to avoid blocking page render.
- Keep tables for tabular data only; use CSS Grid/Flexbox for layout.