HTML CODING STANDARDS AND GUIDELINES

INTRODUCTION

This document presents a set of coding standards and best practices for crafting HTML (Hypertext Markup Language) to maintain uniformity, enhance maintainability, and improve readability throughout our projects. Strict adherence to these standards is crucial for delivering top-notch HTML code quality.

Table of Contents

1. Document Type Declaration

1.1. Document Type Declaration

Always include a valid document type declaration at the beginning of HTML documents to specify the document type. For HTML5, use **<!DOCTYPE html>**.

1.2. HTML Element

You should always include the lang attribute inside the <html> tag, to declare the language of the Web page. This is meant to assist search engines and browsers.

1.3. Character Encoding

Set the character encoding to UTF-8 in the document's **`<head>`** using **`<meta charset="UTF-8">`**.
The **<head>** should also at a minimum include **"viewport"** and **"charset"** meta tags.

1.3. Meta Data

To ensure proper interpretation and correct search engine indexing, both the language and the character encoding <meta charset="charset"> should be defined as early as possible in an HTML document:

SAMPLE CODE FOR DOCUMENT STRUCTURE

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Example Site</title>
  </head>
  <body></body>
</html>
```

2. Indentation, Formatting And Naming Convention

2.1. Indentation

Use consistent indentation with either spaces or tabs (typically 2 or 4 spaces) to improve readability. Choose one and stick with it throughout the project.

2.2. Lowercase Element and Attribute Names

Use lowercase for HTML element names and attributes to ensure consistency.

Example:
```
<a href="https://www.w3schools.com/html/">Visit our HTML
        tutorial</a>
```

2.3. Attribute Quotes

Enclose attribute values in double quotes (`"`), not single quotes (`'`) or without quotes.

2.4. Self-Closing Elements

Use self-closing tags for void elements like `<img>`, `<input>`, and `<br>` without trailing slashes (e.g., `<img src="image.jpg">`).

2.5 Naming conventions

- Class names, IDs and data attributes should be written with a dash, e.g. .item-form or data-item-id.
- Class names should be chosen based on their semantic values, not on their appearance or a particular location on the layout, e.g. .error instead of .red, or .main-menu. instead of .top-menu.

3. Document Structure

3.1. HTML Structure

Follow a logical and semantic structure for HTML documents. Use `**<header>**`, `**<nav>**`, `**<main>**`, `**<article>**`, `**<section>**`, `**<aside>**`, and `**<footer>**` elements appropriately.

3.2. Head Section

Always place metadata elements (e.g., `**<meta>**`, `**<title>**`, `**<link>**`) within the `**<head>**` section.

3.3. Never Skip the <title> Element

The **<title>** element:

- defines a title in the browser toolbar
- provides a title for the page when it is added to favorites
- displays a title for the page in search-engine results

So, try to make the title as accurate and meaningful as possible:

3.4. Script and Stylesheet Placement

Place external scripts and stylesheets before the closing `</body>` tag for faster page rendering.

Example:
```
<link rel="stylesheet" href="styles.css">
<script src="myscript.js">
```

4. Comments

4.1. Code Comments

Use comments to explain complex sections or to provide context. Follow a consistent comment style
(e.g., **`<!-- This is a comment -->`**).

5. Attributes, Values, File Names

5.1. Attribute Order

Maintain a consistent order for attributes (e.g., `class`, `id`, `src`, `alt`, `href`) to improve readability.

5.2. Use Lowercase Attribute Names

- Mixing uppercase and lowercase names looks bad
- Developers normally use lowercase names
- Lowercase looks cleaner
- Lowercase is easier to write

Example:
```
<a href="https://www.w3schools.com/html/">Visit our HTML
tutorial</a>
```

5.2. Always Quote Attribute Values

- Quoted values are easier to read
- You must use quotes if the value contains spaces

Example:
```
<table class="striped">
```

5.3. Use Lower Case File Names

Some web servers (Apache, Unix) are case sensitive about file names: "london.jpg" cannot be accessed as "London.jpg".

If you move from a case-insensitive to a case-sensitive server, even small errors will break your webpage.

To avoid these problems, always use lowercase file names.

5.3. File Extensions

HTML files should have a **.html** extension (**.htm** is allowed).

6. Images Attribute Structure

6.1. Alt Text

Always provide descriptive `**alt**` text for images to enhance accessibility.

6.2. Always define width and height of images

Also, always define the **width** and **height** of images. This reduces flickering, because the browser can reserve space for the image before loading.

Example:

```
<img src="html5.gif" alt="HTML5" style="width:128px;height:128px">
```

7. Forms

7.1. Form Attribute Structure

Form fields must always include a <label> element with a "for" attribute matching the "id" on the input. This helps accessibility by focusing the input when the label is clicked, it also helps screen readers match labels to their respective inputs.

Example:

```
<label for="field-email">email</label><input type="email" id="field-email" name="email" value="" />
```

8. Viewport

8.1. Setting The Viewport

You should include the following <meta> element in all your web pages.It gives the browser instructions on how to control the page's dimensions and scaling.

Example:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

9. Best practices

9.1. Spaces and Equal Signs

HTML allows spaces around equal signs. But space-less is easier to read and groups entities better together.

Example: `<link rel="stylesheet" href="styles.css">`

9.2. Avoid Long Code Lines

When using an HTML editor, it is not convenient to scroll right and left to read the HTML code. Try to avoid too long code lines.

9.3. File Organization

Organize HTML files into a clear directory structure.