

# HTML top 10 Coding and Tips



|                                                      |          |
|------------------------------------------------------|----------|
| <b>Use semantic HTML elements</b>                    | <b>2</b> |
| <b>Use CSS to separate presentation from content</b> | <b>3</b> |
| <b>Use HTML tables for tabular data</b>              | <b>3</b> |
| <b>Use alt attributes for images</b>                 | <b>4</b> |
| <b>Use HTML forms for user input</b>                 | <b>4</b> |
| <b>Use CSS classes and IDs for styling</b>           | <b>5</b> |
| <b>Validate your HTML</b>                            | <b>6</b> |
| <b>Use a CSS reset</b>                               | <b>6</b> |
| <b>Use CSS grid and flexbox for layout</b>           | <b>7</b> |
| <b>Minimize the use of HTML inline styles</b>        | <b>8</b> |

# Use semantic HTML elements

Use semantic HTML elements: Semantic HTML elements provide meaning to the structure of your web pages. Examples of semantic elements include `<header>`, `<nav>`, `<main>`, `<article>`, `<section>`, `<aside>`, `<footer>`, etc. For example:

```
<header>
  <h1>Header</h1>
  <nav>
    <ul>
      <li><a href="#">Home</a></li>
      <li><a href="#">About</a></li>
      <li><a href="#">Contact</a></li>
    </ul>
  </nav>
</header>
<main>
  <h2>Main Content</h2>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit.</p>
</main>
<footer>
  <p>Footer</p>
</footer>
```

## Use CSS to separate presentation from content

Use CSS to separate presentation from content: CSS provides a way to separate the presentation of your web pages from the content, making it easier to maintain and change the look and feel of your site. For example:

```
<style>
  h1 {
    font-size: 36px;
    color: blue;
    text-align: center;
  }
  p {
    font-size: 18px;
    color: green;
    text-align: justify;
  }
</style>
<h1>Header</h1>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit.</p>
```

## Use HTML tables for tabular data

Use HTML tables for tabular data: HTML tables provide a way to display tabular data, such as schedules, financial data, etc. For example:

```
<table>
```

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

## Use alt attributes for images

Use alt attributes for images: The alt attribute provides a text description for images, which is important for accessibility and SEO. For example:

```

```

## Use HTML forms for user input

Use HTML forms for user input: HTML forms provide a way for users to enter data into your web pages. For example:

```
<form>
```

```
<label for="name">Name:</label>
<input type="text" id="name" name="name">
<br>
<label for="email">Email:</label>
<input type="email" id="email" name="email">
<br>
<input type="submit" value="Submit">
</form>
```

## Use CSS classes and IDs for styling

Use CSS classes and IDs for styling: CSS classes and IDs provide a way to select and style specific elements on your web pages. For example:

```
<style>
  /* Define a CSS class */
  .highlight {
    background-color: yellow;
    font-weight: bold;
  }
  /* Define a CSS ID */
  #header {
    background-color: lightgray;
    padding: 20px;
    text-align: center;
  }
</style>
<body>
  <div id="header">
    <h1>Welcome to my website</h1>
```

```
</div>
<p>This is some example text.</p>
<p class="highlight">This text is highlighted.</p>
</body>
```

In this example, the CSS class `.highlight` is used to apply a yellow background and bold font to the text in the second paragraph. The CSS ID `#header` is used to apply a light gray background, padding, and centered text to the div with the ID of header. By using CSS classes and IDs to style elements, you can separate presentation from content, making your HTML code easier to maintain.

## Validate your HTML

Validate your HTML: Validating your HTML helps ensure that your web pages are well-formed and comply with web standards. This can help prevent issues with rendering and accessibility. You can validate your HTML using online tools such as the W3C HTML Validator. <https://validator.w3.org/>

## Use a CSS reset

Use a CSS reset to ensure consistent styling across different browsers: A CSS reset provides a way to reset the default styling for HTML elements, so that you can start with a consistent base across all browsers. This helps to prevent inconsistencies in the appearance of your web pages in different browsers. For example:

```
<style>
  /* CSS Reset */
  * {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
  }
  /* Your custom styles */
  body {
    font-family: Arial, sans-serif;
    font-size: 16px;
    background-color: #f2f2f2;
  }
</style>
```

## Use CSS grid and flexbox for layout

Use CSS grid and flexbox for layout: CSS grid and flexbox provide modern and efficient ways to layout your web pages. For example:

```
<style>
  .container {
    display: grid;
    grid-template-columns: 1fr 1fr;
    grid-gap: 20px;
  }
  .item {
    background-color: lightblue;
    padding: 20px;
    text-align: center;
  }
```

```
    }  
</style>  
<div class="container">  
    <div class="item">Item 1</div>  
    <div class="item">Item 2</div>  
    <div class="item">Item 3</div>  
    <div class="item">Item 4</div>  
</div>
```

## Minimize the use of HTML inline styles

Minimize the use of HTML inline styles: While it is possible to use inline styles in HTML, it is better practice to use CSS classes and IDs to separate presentation from content. Inline styles make your HTML harder to maintain and can result in code duplication. It is better to define styles in a separate CSS file or in the `<style>` element in the `<head>` of your HTML document.