**Gemini**

**Applying Styles Directly to HTML Elements**

**Inline Styles** are a method of applying CSS styles directly to individual HTML elements using the style attribute. This attribute is placed within the opening tag of the element, and its value is a CSS declaration.

**Example:**

HTML

<!DOCTYPE html>

<html>

<head>

<title>Inline Styles Example</title>

</head>

<body>

<h1 style="color: red; font-size: 36px;">This is a red, large header</h1>

<p style="background-color: yellow; font-weight: bold;">This is a bold paragraph with a yellow background.</p>

</body>

</html>

Use code [with caution.](/faq#coding)

In this example:

* The h1 element has the style attribute with the declaration color: red; font-size: 36px;. This makes the header text red and large.
* The p element has the style attribute with the declaration background-color: yellow; font-weight: bold;. This gives the paragraph a yellow background and makes the text bold.

**Pros of Inline Styles:**

* **Quick and Easy:** Applying styles directly to elements can be a quick way to make changes.
* **Specific to Individual Elements:** Inline styles are tailored to a particular element, providing granular control.

**Cons of Inline Styles:**

* **Poor Readability:** Inline styles can clutter HTML code, making it harder to read and maintain.
* **Lack of Reusability:** Styles defined inline cannot be reused across multiple elements or documents.
* **Separation of Concerns:** Mixing styles with content violates the principle of separation of concerns.

**Best Practices:**

* **Use inline styles sparingly:** Reserve them for minor adjustments or temporary styling.
* **Prioritize external stylesheets:** For most styling needs, use external stylesheets to improve code organization and reusability.
* **Consider internal styles:** If you need to apply styles to multiple elements within a single HTML document, consider using internal styles within the <head> section.

By understanding the pros and cons of inline styles, you can use them effectively while maintaining a clean and maintainable codebase.