What are inline, internal, and external CSS, and when should you use each one?

CSS can be applied to a webpage in three main ways: inline, internal, or external.

Each method has its own use case, advantages, and limitations, and knowing when to use each one is essential for writing clean, efficient, and maintainable code.

Let’s break down the three types of CSS and when you should use them.

Inline CSS is written directly within an HTML element using the style attribute. It applies styles to a specific element.

Here's an example using inline CSS:

<p style="color: red;">This is an inline-styled paragraph.</p>

In this example, we are using the style attribute to set the paragraph text to red.

Inline CSS is generally used for quick, one-off styles or to override other styles for a specific element.

However, it should be avoided in most cases because it can clutter the HTML and make the code harder to maintain.

Most of the time, it's better to use internal or external CSS to keep your styles organized and maintainable.

Internal CSS is written within the style tags inside the head section of an HTML document. It applies styles to the entire page and is useful when you need to style a single document.

Here's an example of internal CSS:

<head>

<style>

p {

color: blue;

}

</style>

</head>

<body>

<p>This paragraph is styled using internal CSS.</p>

</body>

In the above example, the internal CSS applies blue text to all p elements on the page.

Internal CSS is best used when you need to apply styles to a specific page rather than across multiple pages. It’s useful for single-page websites or when the styles don’t need to be reused elsewhere.

However, there are some downsides, such as not promoting reusability across multiple pages. Additionally, like inline CSS, it mixes HTML and CSS, making the code harder to maintain in larger projects.

External CSS is written in a separate .css file and linked to the HTML document using the link element in the head section.

It allows you to style multiple pages consistently and is the preferred method in professional web development.

Here's the HTML part of our code example:

<head>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<p>This paragraph is styled using external CSS.</p>

</body>

In an earlier lecture video, you learned that the link element has a rel attribute that specifies the relationship between the current document and the linked resource, such as linking to a stylesheet or an external resource.

On the other hand, the href attribute specifies the URL of the linked resource, indicating where the resource should be retrieved from.

Here the CSS portion of the same example in separate file named style.css.

p {

color: green;

}

This example targets all paragraph elements on the page and sets the text color to green.

External CSS is ideal for large projects where you want to maintain a consistent style across multiple pages.

It promotes separation of concerns by having HTML handle the structure and CSS handle the styling, which makes the code more maintainable and scalable.

Understanding when to use each type of CSS is crucial for efficient and effective web development.

In most cases, external CSS should be your go-to approach, especially for larger and more complex projects.