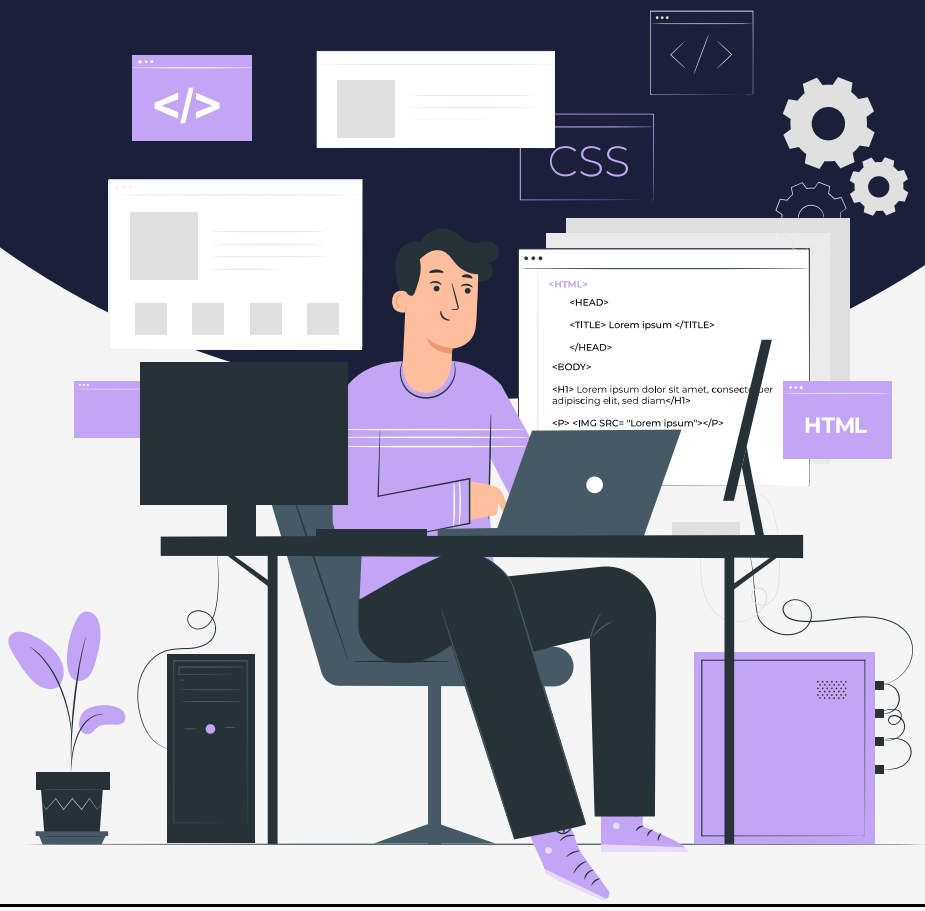
Lesson:

Inline, Internal and External CSS styling and their priority



PW SKILLS

Topics Covered

- Introduction to link CSS to HTML
- Inline CSS styling with example and its priority
- Internal CSS styling with example and its priority
- External CSS styling with example and its priority
- Pros and Cons of inline, internal, external CSS styling
- Which CSS styling should be used.

Introduction to link CSS to HTML

Linking CSS to HTML is a fundamental step in web development to control the presentation and style of web pages. Linking CSS to HTML involves establishing a connection between an HTML document and an external stylesheet or internal style. This allows you to define styles in a separate file and apply them to multiple HTML documents, providing consistency and maintainability across your website.

Inline CSS styling with examples and its priority

Inline styles are one of the ways to add CSS to an HTML document. A style attribute can be added to any HTML tag and CSS properties are added to it.

Inline styles override any CSS Inline or External style sheet

The styles in the following example apply directly to the elements to which they are attached.

index.html

```
Unset
<h1 style="color:green; text-decoration:underline;"> Hello
world </h1>
```

Browser output-

Hello world

Internal CSS styling with example and its priority

An internal stylesheet is a method of adding CSS rules directly within the <style></style> element in the <head> section of an HTML document. This allows developers to define styles for specific HTML elements of classes within the same HTML file, without the need for an external stylesheet

It has priority less than the inline styles and will be able to override external styles but not the inline styles.

Example of Internal CSS style

```
Unset
<!DOCTYPE html>
<html>
  <head>
    <title>Example of Semantic HTML Tags</title>
    <style>
      h1 {
        color: green;
        text-decoration: underline;
      }
    </style>
  </head>
  <body>
    <h1>Hello World</h1>
  </body>
</html>
```

Browser output-

Hello world

External CSS styling with example and its priority

An external stylesheet is a separate file that contains CSS rules and is linked to an HTML document using the <link> element. This allows you to define styles in a separate file, which can be reused across multiple HTML documents, making it easier to maintain and update the styles across your website.

The External CSS has less priority compared to Inline and Internal CSS styles as a result it will not be able to override the styles in the Inline and Internal styles.

Example of External CSS styling

Using <link> tag to link the HTML document with the external Style sheet.

```
Unset
  <link rel="stylesheet" href="style.css">
```

The rel attributes are required too, as they tell the browser which kind of file we are linking.

index.html

```
Unset
<!DOCTYPE html>
<html>
  <head>
    <title>Example of External style sheet</title>
    <link rel="stylesheet" href="style.css" />
    <style></style>
  </head>
  <body>
    <h1>Hello World</h1>
  </body>
</html>
```

style.css

```
Unset
h1 {
  color: green;
  text-decoration: underline;
}
```

Browser output-

Hello world

Pros and Cons of Inline, Internal, External CSS styling -

CSS styling	Pros	Cons
Inline	• Provides complete control over the styling of a specific element • Easier to override global styles for specific elements since it has the highest specificity.	• Can be difficult to maintain, especially on larger projects • Increase page load times if used excessively • Not very scalable, as each individual element needs to have its own inline style.

Internal	• More maintainable than inline CSS, as styles can be defined once in the head section of the HTML document and applied to multiple elements. • It provides greater flexibility than inline CSS • It is easier to override styles than inline CSS.	• - It can be difficult to apply styles consistently across multiple pages, as style must be defined in each HTML document individually • - It can still lead to code duplication if the same styles are defined in multiple sections of the same HTML document.
External	• It is the most maintainable option, as styles can be defined in a single file and applied to multiple pages. • It is the most efficient option, as the CSS code can be cached by the browser, reused across multiple files, and easily edited. • It allows for greater organisation and consistency, as styles can be separated into different files and easily edited • it is the most recommended approach for small to big projects.	• It may not be as flexible as inline or external CSS, as styles must be defined in a separate file and cannot be applied directly to an HTML element. • It requires an additional HTTP request, which can slow down the page load time. • it has lower specificity than inline and internal CSS, which means it can be overridden by other styles applied to the same element.

Which CSS styling should be used -

When it comes to choosing between inline, internal and external CSS, there are a few factors to consider -

- **Re usability -** for working with multiple pages, it's best to use an external CSS styling so that the style can be easily reused across multiple pages.
- **Maintenance -** for frequent updates, it's best to use an external CSS styling. If it's a single page the inline or internal styles may be more appropriate.
- **Performance -** External styling files can be cached by the browser which can improve performance by reducing the amount of data that needs to be downloaded each time a page is loaded.
- **Specificity -** if you need to override styles from a third-party library or other external sources, inline or internal styles with higher specificity may be necessary.
- **Organization** - Using external CSS files can help keep your code organized and easier to read, especially if you have a lot of styles or complex style-sheets. Inline and internal styles can make your code more cluttered and hard to read and debug.

As a conclusion, it's best to use external CSS files for most styles, as this offers the best balance of reusability, maintenance, and performance. Inline and internal styles should be reserved for cases where they are necessary, such as for overriding styles or for very specific styles that are only used on one page or element.