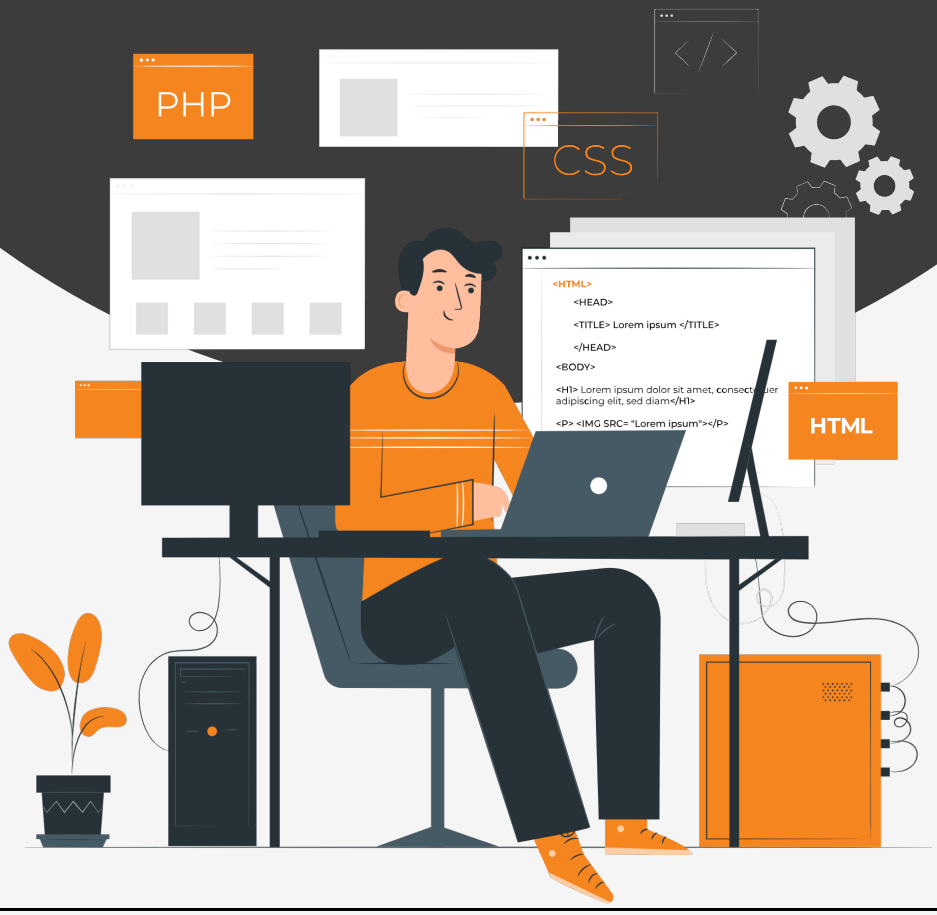
Lesson:

Ways to add CSS



PW SKILLS

Topics Covered

- Inline CSS
- Internal CSS
- External CSS
- Cascading Order of Inline, Internal and External CSS.

Cascading Style Sheets (CSS) is a powerful tool that allows web developers to control the visual appearance and layout of HTML documents. CSS provides a wide range of styling options to enhance the presentation of web pages, making them more visually appealing and user-friendly.

In this documentation, we will explore the following ways to add CSS to HTML documents:

- **Inline CSS:** This is the simplest way to add CSS to an HTML document, but it is not recommended for large or complex projects.
- **Internal CSS:** This is a more efficient way to add CSS to an HTML document, as it allows you to store all of your CSS code in a single file.
- **External CSS:** This is the most recommended way to add CSS to an HTML document, as it allows you to separate the content (HTML) from the presentation (CSS).

Styling plays a crucial role in creating attractive and professional-looking websites. By separating the content (HTML) from the presentation (CSS), web developers can maintain cleaner code and easily make global design changes without altering the content. Additionally, using external CSS files promotes reusability and consistency throughout a website.

Inline CSS

CSS, or Cascading Style Sheets, is a powerful tool for styling HTML elements. One way to apply CSS to HTML documents is through inline CSS. Inline CSS involves using the style attribute to apply specific styles directly to an HTML element. This means that the styles are defined within the element's opening tag, allowing you to add a unique look and feel to that particular element without the need for a separate CSS file or <style> element in the document.

Inline CSS is a straightforward and immediate way to style individual elements, but it is not the most flexible or reusable method. If you need to style multiple elements with the same properties, or if you want to be able to reuse your styles in other documents, then you should use an external CSS file.

Syntax:
To use inline CSS, you add the style attribute to an HTML element, followed by CSS property-value pairs within double quotes. The syntax is as follows:

```JavaScript
<element style="property: value;">
```

Example:
Let's say you want to style a paragraph with blue text color and a font size of 16 pixels. Here's how you can achieve it using inline CSS:

```javascript
<p style="color: blue; font-size: 16px;">This is a paragraph with inline CSS.</p>
```

Browser output –

This is a paragraph with inline CSS.

Pros of Inline CSS:

- **Quick application:** Inline CSS allows you to apply styles to specific elements directly in the HTML code, without the need to create a separate CSS file. This can be useful for making quick changes or for applying styles to elements that are not part of a larger design system.

- **Specificity:** Styles applied through inline CSS have a higher specificity than styles defined in external or internal stylesheets. This means that they will override other styles, which can be useful for applying temporary or critical styles that need to take precedence over other styles.
- **Inline conditional styling:** Inline CSS can be used to dynamically apply styles based on conditions or variables generated by server-side scripts. This can be used to customize the appearance of elements as needed, such as displaying different styles for different screen sizes or user roles.

Cons of Inline CSS:

- **Lack of separation:** Inline CSS mixes content and presentation, which can make it harder to maintain, especially in larger projects. This is because inline CSS is embedded within the HTML code, which can make it difficult to distinguish between the content and the styles.
-
- **Code repetition:** If the same styles are applied to multiple elements, you'll need to repeat the same inline CSS for each element. This can lead to code duplication and make it harder to maintain the code.
-
- **Limited reusability:** Styles applied inline cannot be reused across multiple pages or elements. This means that if you make a change to the inline styles, you'll need to make the same change to every element that uses those styles.
- **Difficult to manage:** As the project grows, managing inline CSS can become cumbersome and may lead to maintainability issues. This is because inline CSS can be difficult to track and troubleshoot, and it can be difficult to make changes to the styles without affecting other parts of the code.

Overall, inline CSS can be a useful tool for making quick changes or for applying styles to elements that are not part of a larger design system. However, it is important to be aware of the limitations of inline CSS before using it. If you need to apply styles to multiple elements or if you need to be able to reuse your styles, then you should use an external CSS file.

Internal CSS

When working on a web page, you may need to style specific elements in a way that is unique to that page. In these cases, you may not want to create a separate CSS file for just a few styles. This is where internal CSS comes in. Internal CSS, also known as embedded CSS or <style> tag CSS, allows you to define CSS rules directly within the HTML document itself.

Unlike external CSS, which requires linking to a separate CSS file, internal CSS keeps the styling information within the HTML document's <head> section. This approach provides a more localized and self-contained solution for applying styles to specific elements on the page.

Syntax:
To use Internal CSS, you include a <style> element within the <head> section of your HTML document. Inside the <style> element, you can write CSS rules using standard CSS syntax.

```
JavaScript
<!DOCTYPE html>
<html>
<head>
    <title>Internal CSS Example</title>
    <style>
        /* CSS rules go here */
        selector {
            property: value;
        }
    </style>
</head>
<body>
    <!-- HTML content goes here -->
</body>
</html>
```

Example:
Let's say you have a simple HTML page with two paragraphs, and you want to style those paragraphs differently. Here's how you can achieve it using Internal CSS:

```
JavaScript
<!DOCTYPE html>
<html>
<head>

    <title>Internal CSS Example</title>
     <style>
        p {
            color: blue;
            font-size: 16px;}
        .special-paragraph {
            color: green;
            font-size: 18px;
            font-weight: bold; }
    </style>
</head>
<body>
    <p>This is a regular paragraph with internal CSS.</p>
    <p class="special-paragraph">This is a special paragraph with
internal CSS.</p>
</body>
</html>
```

Browser output -

This is a regular paragraph with internal CSS.

This is a special paragraph with internal CSS.

Pros of Internal CSS:

- **Separation of concerns:** Internal CSS keeps the presentational (CSS) code within the same HTML document as the content, which promotes better code organization and maintainability. This is because it helps to separate the concerns of content and presentation, making it easier to understand and maintain the code.

- **Specificity and reusability:** Internal CSS has higher specificity than external CSS, meaning it can override styles defined in external stylesheets. Additionally, styles defined internally can be reused across multiple elements within the same HTML document. This can save time and effort, as you don't need to define the same styles multiple times.
- **Efficient for small projects:** For small projects with limited styling needs, internal CSS can be a quick and efficient way to apply styles without the overhead of creating an external CSS file. This is because the CSS rules are embedded directly in the HTML document, which can improve performance.

Cons of Internal CSS:

- **Limited scope:** Internal CSS only applies to the HTML document in which it is defined. If you have multiple HTML pages with similar styling requirements, you'll need to duplicate the internal CSS across each document. This can be time-consuming and error-prone.
- **Code repetition:** When using internal CSS, you may need to repeat the same CSS rules for specific elements if they appear on different pages. This can make the code more difficult to read and maintain.
- **Difficult collaboration:** In larger projects with multiple developers, managing styles within the HTML document might lead to conflicts and difficulties in collaborative development. This is because multiple developers may be editing the same styles, which can lead to inconsistencies.

External CSS

External CSS is a widely-used method of styling HTML documents. It involves creating a separate CSS file that contains all the styles for a website. This file is then linked to the HTML file.

External CSS promotes separation of concerns, which means that the presentation (CSS) is kept separate from the content (HTML). This makes it easier to maintain a cleaner and more organized codebase.

External CSS can help you maintain consistency, reuse styles across multiple HTML pages, and efficiently manage your styles across an entire website.

Creating an External CSS File:
To implement External CSS, start by creating a new file with a .css extension (e.g., styles.css) and define your CSS rules inside it. Each CSS rule should follow the standard CSS syntax.

style.css

```javascript
/* CSS rules go here */
selector {
    property: value;
}

/* Example styles */
p {
    color: blue;
    font-size: 16px;
}
```

Linking External CSS to HTML:
Once you've created your external CSS file, you need to link it to your HTML documents. This is achieved by adding a <link> element in the <head> section of your HTML file, specifying the path to the CSS file using the href attribute.

```javascript
<!DOCTYPE html>
<html>
<head>
    <title>External CSS Example</title>
    <link rel="stylesheet" type="text/css" href="./styles.css">
</head>
<body>
    <!-- HTML content goes here -->
</body>
</html>
```

Pros of External CSS

- **Separation of concerns:** External CSS allows you to keep the content and presentation of your website separate. This makes your HTML code cleaner and easier to read and manage.
- **Reusability:** External CSS allows you to reuse styles across multiple HTML pages. This can save you time and effort, and it helps to ensure that your website has a consistent look and feel.
- **Efficient development:** External CSS makes it easier for multiple developers to work on the same website simultaneously. This is because each developer can focus on their own area of expertise, such as the content or the presentation.
- **Caching and performance:** External CSS files can be cached by browsers. This means that they will only be downloaded once, even if they are linked to from multiple HTML pages. This can improve the performance of your website by reducing the number of HTTP requests that need to be made.

Cons of External CSS

- **Dependency on file linking:** External CSS requires that each HTML page correctly link to the CSS file. If the link is missing or incorrect, the styles will not be applied.
- **Extra HTTP request:** Each external CSS file requires a separate HTTP request. This can slightly impact the initial loading time of the web page.

Cascading Order of Inline, Internal & External CSS

CSS follows a specific cascading order to determine which styles should be applied to HTML elements when multiple CSS rules target the same element.

Cascading Order:

1. **CSS follows a specific cascading order to determine which styles should be applied to HTML elements when Inline CSS (Highest Specificity)**
2. **Internal CSS**
3. **External CSS (Lowest Specificity)**

1. Inline CSS (Highest Specificity)

Inline CSS is applied directly to individual HTML elements using the style attribute. It has the highest specificity among all CSS types, which means it takes precedence over external and internal CSS. When an inline style is defined, it will override any styles from internal and external stylesheets that target the same element.

2. Internal CSS

Internal CSS, also known as embedded CSS, is defined within the <style> element in the <head> section of the HTML document. It has a higher specificity than external CSS but a lower specificity than inline CSS. Internal styles are applied to all elements that match the specified selectors within the same HTML document.

3. External CSS (Lowest Specificity)

External CSS is defined in a separate CSS file (e.g., styles.css) and linked to the HTML document using the <link> element in the <head> section. It has the lowest specificity among all CSS types. External styles are applied to all elements that match the specified selectors across multiple HTML documents linked to the same CSS file.

Note-

Out of the above three approaches the External is always preferred, some of the reason are given below -

- Modularity- Using external style confines conflict to specific stylesheet files, enabling focused conflict identification and resolution for each section.
- Isolated conflicts - Conflicts are isolated to designated external stylesheets, enabling localized conflict resolution without impacting other styles.
- Specificity Management precision - Utilizing separate CSS files enables effective specificity management, preventing unintended style overrides and ensuring intended styles.