

Implementing CSS to Make Pages Appealing

Demo - 1

edureka!

© Brain4ce Education Solutions Pvt. Ltd.

Problem Statement:

Linking a Stylish Webpage with CSS

In this problem, you will be creating an aesthetically captivating webpage for a fictional technology-oriented company renowned for its innovative solutions. Your objective is to harness your proficiency in CSS (Cascading Style Sheets) to elevate the webpage's design and layout, utilizing a spectrum of advanced CSS techniques. Your task involves elevating the webpage's appeal by strategically implementing diverse CSS selectors. Demonstrating your adeptness with varied selector types — encompassing element name, ID, class, universal, and grouping selectors — you will precisely target distinct elements, including headings, paragraphs, lists, and navigation components. This endeavor aims to exemplify your prowess in crafting tailored styles, showcasing an intricate comprehension of selector mechanics.

Solution: The problem is to link the CSS file with HTML Webpage to make it stylish and visually appealing. There are three ways to link a Cascading Style Sheet with an HTML Webpage are:

1. Inline CSS
2. Internal CSS
3. External CSS

Let's explore each one of them one-by-one:

- 1. Inline CSS** - In HTML, you can use the inline way to apply CSS styles directly to individual HTML elements using the style attribute. This approach is known as Inline CSS. To apply inline CSS, open an HTML tag you want to style.

For example, you can open a <p> (paragraph) tag:

```
<p style="property: value;">  
    This is a paragraph with inline CSS.  
</p>
```

Inside the opening tag of the HTML element, add the style attribute. The style attribute contains one or more CSS property-value pairs separated by semicolons. Each property-value pair defines a specific style for the element.

For example, to change the text color of the paragraph to blue and set the font size to *16 pixels*, you can use the following code:

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

Using inline CSS can be helpful for applying quick and specific styles to individual elements, but it's important to note that this approach can lead to code repetition and might not be as maintainable as using external or internal CSS for larger projects. Consider using external or internal CSS stylesheets for consistent styling across multiple elements.

- 2. Internal CSS:** Using the internal way to link CSS with HTML involves placing your CSS code directly within the HTML document's `<style>` element. This approach is known as internal CSS.

In the **<head>** section of your HTML document, open the **<style>** element. This is where you will write your CSS code.

```
<!DOCTYPE html>  
<html>  
  <head>  
    <style>  
      /* Your CSS code will go here */  
    </style>  
  </head>  
  <body>  
    <!-- Your HTML content goes here -->  
  </body>
```

```
</html>
```

- 3. External CSS:** Using the external way to link CSS with HTML involves creating a separate CSS file and linking it to your HTML document using the **<link>** element. This approach is known as external CSS. Here's how you can use external CSS to style HTML elements:

Create a CSS File: Start by creating a new file with a **.css** extension. You can name it something like **styles.css**. This file will contain your CSS rules.

Write CSS Code: Inside the CSS file, write your CSS code just like you would for internal CSS or any other stylesheet.

```
/* styles.css */  
/* Define CSS rules */  
p {  
    color: blue;  
    font-size: 16px;  
}  
  
h1 {  
    color: green;  
    text-align: center;  
}
```

Link the CSS File in HTML: In your HTML document's **<head>** section, use the **<link>** element to link the external CSS file. Place this link between the opening and closing **<head>** tags.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Link to external CSS file -->
    <link rel="stylesheet" type="text/css" href="styles.css">
  </head>
  <body>
    <!-- Your HTML content goes here -->
    <h1>This is a centered heading with green text.</h1>
    <p>This is a paragraph with blue text and a font size of
16px.</p>
  </body>
</html>
```

Specify the CSS File Path: Provide the path to your CSS file in the href attribute of the <link> element. This path can be absolute or relative to the HTML file.

- Absolute Path Example: **href="https://example.com/css/styles.css"**
- Relative Path Example: **href="styles.css"**

Once the CSS file is linked, the defined styles will be applied to the HTML elements based on the selectors.

Applying Styling on Element using Selectors: You have a variety of selectors, and with the help of these selectors, you can apply styling on your structured webpage:


- Name Selector** - An Element Name Selector in CSS is used to target and style all instances of a specific HTML element on a webpage. It

allows you to apply the same styles to all occurrences of that particular element type.

Syntax: To use the Element Name Selector, simply write the name of the HTML element followed by the desired CSS properties and values within curly braces. The basic syntax for name selector is:

```
element-name {  
    /* CSS properties and values */  
}
```

Example: Let's say you want to style all the **<h2>** elements on your webpage with a red color and a larger font size. You would use the Element Name Selector for **<h2>** like this:



```
h2 {  
    color: red;  
    font-size: 24px;  
}
```

In the example above, both **<h2>** elements will be styled with the CSS's specified color and font size.

- b. ID Selector** - An ID Selector in CSS targets and styles a specific HTML element based on its unique ID attribute. IDs are intended to be unique on a page, so using an ID selector will target only one specific element. To apply an ID selector to an HTML element, ensure that the ID attribute is assigned to the element. IDs are written in the format **id="name-id"**.

Syntax:

To style the element with a specific ID, use the hash (**#**) symbol followed by the ID name in your CSS code. Within curly braces **{}** include the desired CSS properties and values.

```
#header {  
    background-color: gray;  
    color: white;  
}
```

Example:

```
<!DOCTYPE html>  
<html>  
  <head>  
    <link rel="stylesheet" type="text/css" href="styles.css">  
  </head>  
  <body>  
    <div id="header">  
      <!-- Content of the header -->  
    </div>  
    <!-- Other content goes here -->  
  </body>  
</html>
```

In the example above, the **<div>** element with the ID **"header"** will have its background color set to gray and text color set to white, as specified in the CSS.

- c. **Class Selector** – It targets and styles multiple HTML elements that share the same class attribute value. Classes allow you to apply consistent styles to different elements across your webpage. To use the Class Selector, assign the same class attribute value to multiple HTML elements you want to style similarly. Class names are written in the format **class="your-class"**.

Example:

```
<p class="highlight">This is a highlighted paragraph.</p>  
<p class="highlight">Another highlighted paragraph.</p>
```

Syntax: To style elements with a specific class, use a period (.) followed by the class name in your CSS code. Within curly braces {}, include the desired CSS properties and values.

```
.highlight {  
    background-color: yellow;  
    font-weight: bold;  
}
```

In the example above, <p> elements with the class "highlight" will have a yellow background and bold font weight, as specified in the CSS.

- d. Universal Selector** – It targets all HTML elements on a webpage. It allows you to apply a specific style to every element within the document.

Syntax: To use the Universal Selector, simply use an asterisk (*) in your CSS code. Within curly braces {}, include the desired CSS properties and values.

Example:

```
* {  
    margin: 0;  
    padding: 0;  
}
```

In the example above, all HTML elements within the <body> will have their margins and padding set to zero, as specified in the CSS. This creates consistent spacing throughout the page.

- e. Grouping Selector** – It allows you to apply the same styles to multiple selectors at once. This can help you achieve consistency in styling across different elements that share similar characteristics.

Syntax: To use the Grouping Selector, simply list the selectors you want to group, separated by commas, and apply the desired styles within curly braces {}.

```
h1, h2, h3 {  
    color: blue;  
    font-family: Arial, sans-serif;  
}
```

Example:

```
<!DOCTYPE html>  
<html>  
  <head>  
    <link rel="stylesheet" type="text/css" href="styles.css">  
  </head>  
  <body>  
    <h1>This is a blue heading</h1>  
    <h2>Another blue heading</h2>  
    <h3>Yet another blue heading</h3>  
    <!-- Other content goes here -->  
  </body>  
</html>
```

In the example, all <h1>, <h2>, and <h3> elements will have the specified blue color and font family, as defined in the CSS.

Conclusion: In this way, you can apply CSS to your webpage to make them look visually appealing.