

Web development

lesson 7

Oct 13, 2024

Introduction to CSS

- **What is CSS?**

- CSS stands for Cascading Style Sheets.
- It's a stylesheet language used to describe the presentation of HTML documents.
- CSS controls the layout, colors, fonts, and overall visual appearance of web pages.

- **Purpose of CSS in Web Design**

- CSS allows for the separation of content (HTML) from presentation (CSS).
- Enhances user experience by enabling responsive design, animations, and visual effects.
- Simplifies the process of updating styles, as changes in one CSS file can update the entire site.

Introduction to CSS 2

- **Benefits of CSS: Separation of Content and Presentation**

- **Maintainability:** Changes to the design can be made in one place without altering the HTML structure.
- **Reusability:** CSS rules can be reused across multiple pages or entire websites, promoting consistency.
- **Accessibility:** CSS can improve web accessibility by making it easier to adapt content for screen readers and other assistive technologies.
- **Performance:** By reducing inline styles, CSS helps decrease the size of HTML files, which can improve page loading times.

How CSS Works

- **How Browsers Render CSS**

- The browser first parses the HTML document, then the linked or embedded CSS.
- CSS is applied in a cascade, with rules being applied based on specificity and source order.
- The browser's rendering engine constructs a “render tree,” combining HTML content with applied CSS styles, then paints the final output on the screen.

- **Relationship between HTML and CSS**

- HTML defines the structure and content of a webpage, while CSS styles and positions that content.
- CSS selectors target HTML elements and apply styles based on class names, IDs, types, and other attributes.
- CSS can be linked externally, embedded within `<style>` tags, or added inline within HTML tags, each having different scopes and impacts.

How CSS Works 2

- **Cascading and Inheritance Concepts**

- **Cascading:**

- The term “Cascading” in CSS means that styles are applied based on a hierarchy, with rules “cascading” from one style to another.
 - This hierarchy is governed by specificity (more specific selectors override less specific ones) and importance (e.g., `!important` declarations).

- **Inheritance:**

- Certain CSS properties (e.g., color, font-family) are inherited by child elements, while others (e.g., padding, margin) are not.
 - Inherited properties help to maintain a consistent look throughout the document without needing to redefine styles for every element.

How CSS Works 3

- **Specificity Calculation:**

- Specificity determines which CSS rule applies when multiple rules could apply to the same element.
- Calculated based on the types of selectors used (ID, class, element), with inline styles and `!important` taking precedence.

CSS Syntax Review

- Anatomy of a CSS rule: selector, property, value
- Importance of semicolons and curly braces

CSS Syntax Structure

- **Basic CSS Rule Structure**

- A CSS rule consists of a **selector** and a **declaration block**.

- Syntax: `selector { property: value; }`

- Example:

```
p {  
  color: blue;  
  font-size: 16px;  
}
```

- In this example:

- The selector `p` targets all `<p>` elements.
- The declarations within `{ }` specify the styles to apply, each as a `property: value;` pair.

Applying CSS: Inline Styles

- **What is Inline CSS?**

- Inline CSS involves adding styles directly to individual HTML elements using the `style` attribute.
- Only affects the specific element, making it the highest specificity.
- Typically used for quick, one-off style changes or testing.

- **Syntax and Example:**

```
<p style="color: red; font-size: 20px;">This is an inline-styled paragraph.</p>
```

Inline Styles

Advantages

- Easy to apply for single-use cases.
- Good for dynamically generated content or JavaScript-controlled styles.

Disadvantages

- Not ideal for maintaining large projects, as it mixes content and presentation. *Harder to manage and update across multiple elements.
- Results in larger HTML files, reducing separation of concerns.

Applying CSS: Internal Styles

What is Internal CSS?

- Internal CSS is applied using the `<style>` tag within the `<head>` section of an HTML document.
- Applies styles to the entire page but only that specific page.

Syntax and Example

```
<head>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f0f0f0;
    }
    h1 {
      color: darkblue;
      text-align: center;
    }
  </style>
</head>
```

Internal Styles

Advantages

- Keeps styles organized within the HTML file.
- Useful for single-page applications or standalone documents.
- Reduces HTTP requests compared to external stylesheets.

Disadvantages

- Limited scope to a single page, so not reusable across multiple pages.
- Can lead to cluttered `<head>` sections if there are numerous styles.
- Not as efficient for large projects requiring a consistent look across pages

Applying CSS: External Stylesheets

What is External CSS?

- External CSS involves linking an external .css file to an HTML document using the `<link>` tag.
- Styles can be applied across multiple pages by linking the same stylesheet, promoting consistency.

Syntax and Example

```
<head>
  <link rel="stylesheet" href="styles.css">
</head>
```

Example of an External CSS File (styles.css)

```
body {
  margin: 0;
  padding: 0;
  font-family: Helvetica, sans-serif;
}
.container {
  width: 80%;
  margin: auto;
}
```

External CSS File

Advantages

- Allows for a consistent design across multiple pages.
- Easy to update styles site-wide by editing a single CSS file.
- Keeps HTML cleaner and more focused on structure, improving separation of concerns.

Disadvantages

- Requires an additional HTTP request to load the CSS file (can be mitigated with caching).
- Styles won't apply if the CSS file is unavailable or not linked properly.
- Less direct than inline or internal styles for quick changes during development