COACH

Introduction to CSS

Lesson: Introduction to CSS - Styling the Web

1. Objectives:

- Understand what CSS stands for and its purpose in web development.
- Explain the relationship between HTML and CSS (structure vs. presentation).
- Identify the basic syntax of a CSS rule (selector, property, value).
- Learn the three ways to apply CSS to HTML (Inline, Internal, External) and understand the pros/cons of each.
- Use basic CSS selectors (element, class, ID) to target HTML elements.
- Apply common CSS properties to change text color, background color, font size, font family, borders, padding, and margins.
- Modify an existing HTML file by adding CSS styles.

2. Target Audience:

- Beginners who have completed the "Introduction to HTML" lesson or have a basic understanding of HTML structure and tags.

3. Prerequisites:

- Familiarity with basic HTML tags (**<h1>**-**<h6>**, **<p>**, **<a>**, **<img>**, ** **, ****).
- Understanding of HTML document structure (**<!DOCTYPE>**, **<html>**, **<head>**, **<body>**).
- Ability to create and save HTML files using a text editor.
- Ability to open HTML files in a web browser.

4. Estimated Duration:

- 60 - 75 minutes

5. Materials Needed:

- A computer (Windows, Mac, or Linux).
- A simple text editor (like Notepad, TextEdit, VS Code, Sublime Text, Atom).
- A web browser (like Chrome, Firefox, Edge, Safari).
- The **mypage.html** file created in the HTML lesson (or a similar simple HTML file to practice on).

6. Lesson Content & Procedure:

Part A: What is CSS? (5-10 minutes)

- **Analogy Revisited:** If HTML is the **skeleton** of the webpage (providing structure), CSS (Cascading Style Sheets) is like the **skin, clothes, and overall appearance**. It controls the colors, fonts, layout, spacing, and visual presentation.
- **Definition:** CSS stands for **C**ascading **S**tyle **S**heets.
  - **Style Sheets:** It's a "sheet" or file containing style rules.
  - **Cascading:** This refers to the rules that govern how styles are applied when multiple rules target the same element. The "cascade" determines which style wins (we'll touch on this later, but the basic idea is that more specific rules often override general ones).
- **Purpose:** To separate the presentation (look and feel) of a webpage from its structure (HTML content). This makes websites easier to maintain, update, and allows for more complex and sophisticated designs. You can change the look of an entire website by editing just one CSS file!

Part B: How CSS Works - The Basic Rule (10 minutes)

- CSS works by defining **rules**. Each rule selects specific HTML elements and declares how they should look.
- A CSS rule has two main parts:
  1. **Selector:** Specifies *which* HTML element(s) the rule applies to (e.g., all **<p>** tags, elements with a specific ID or class).
  2. **Declaration Block:** Contains one or more **declarations**, enclosed in curly braces **{}**. Each declaration includes:
     - A **Property:** The style attribute you want to change (e.g., **color**, **font-size**, **background-color**).
     - A **Value:** The specific setting for that property (e.g., **blue**, **16px**, **#FFFF00**).
     - Declarations end with a semicolon **;**.
- **Syntax:**

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

- **Example:** Make all paragraph (** **) text red.

```
p {
    color: red;
}
```

Part C: Ways to Add CSS to HTML (15 minutes)

There are three main ways to connect your CSS rules to your HTML document:

1. **Inline Styles (Use Sparingly):**
    - **How:** Add the **style** attribute directly to an HTML tag.
    - **Example:** ** This paragraph is blue and small. **
    - **Pros:** Quick for testing or applying a unique style to a single element.
    - **Cons:** Mixes content and presentation, hard to maintain for larger sites, overrides other styles (generally bad practice for overall styling).
2. **Internal Stylesheet (Good for Single Pages):**
    - **How:** Place CSS rules within **<style>** tags, inside the **<head>** section of your HTML file.
    - **Example (inside <head>):**

```
<head>
    <title>My Styled Page</title>
    <style>
        body {
            background-color: lightgray;
        }
        h1 {
            color: navy;
        }
        p {
            color: darkslategray;
        }
    </style>
</head>
```

- **Pros:** Keeps styles for a single page in one place, doesn't require extra files.
- **Cons:** Only applies to that one HTML page; not efficient if you want the same style across multiple pages.

3. **External Stylesheet (Best Practice):**
   - **How:** Write CSS rules in a separate file with a **.css** extension (e.g., **styles.css**). Link this file to your HTML document using the **<link>** tag inside the **<head>** section.
   - **Example (styles.css file):**

```css
/* This is a CSS comment */
body {
    background-color: #f0f0f0; /* A light gray */
    font-family: sans-serif;
}

h1 {
    color: steelblue;
}

p {
    color: #333; /* A dark gray */
    line-height: 1.5; /* Increase space between lines */
}
```

   - **Example (inside <head> of mypage.html):**

```html
<head>
    <title>My Styled Page</title>
    <link rel="stylesheet" href="styles.css">
</head>
```

     - **rel="stylesheet"**: Tells the browser it's a stylesheet.
     - **href="styles.css"**: Specifies the path to the CSS file (must be in the same folder for this example).
   - **Pros: Separation of concerns!** Keeps HTML clean, allows multiple HTML pages to use the same styles, makes site-wide changes easy (edit one file!). This is the standard method.
   - **Cons:** Requires managing an extra file.

Part D: Basic Selectors (10 minutes)

Selectors target the HTML elements you want to style.

1. **Element Selector (Type Selector):** Selects all elements of a specific type.
    - **Syntax: elementName**
    - **Example: h1 { color: green; }** (Selects *all* **<h1>** elements). **p { font-size: 16px; }** (Selects *all* ** ** elements).
2. **Class Selector:** Selects all elements that have a specific class attribute. Classes allow you to group elements and apply the same style to them, even if they are different types (e.g., some ** ** and some **<li>** items).
    - **Syntax: .className** (Note the leading dot **.**)
    - **HTML: This is important. This too! **
    - **CSS: .highlight { background-color: yellow; font-weight: bold; }** (Selects *any* element with **class="highlight"**).
3. **ID Selector:** Selects a *single* element that has a specific id attribute. IDs **must be unique** within an HTML page. Used for targeting one specific element.
    - **Syntax: #idName** (Note the leading hash **#**)
    - **HTML: Site Title **
    - **CSS: #main-header { border-bottom: 2px solid black; padding-bottom: 10px; }** (Selects *only* the element with **id="main-header"**).

Part E: Common CSS Properties (10 minutes)

Here are a few fundamental properties to get started:

- **color**: Sets the text color (e.g., **color: red**;, **color: #FF0000;**).
- **background-color**: Sets the background color of an element (e.g., **background-color: lightblue;**, **background-color: #f0f0f0;**).
- **font-family**: Sets the typeface (e.g., **font-family: Arial, sans-serif;** - provides fallback fonts).
- **font-size**: Sets the size of the text (e.g., **font-size: 18px;**, **font-size: 1.2em;**).
- **font-weight**: Sets the thickness of the text (e.g., **font-weight: bold;**, **font-weight: normal;**).
- **text-align**: Aligns text horizontally (e.g., **text-align: center;**, **text-align: left;**).
- **border**: A shorthand to set border width, style, and color (e.g., **border: 1px solid black;**).
- **padding**: Adds space *inside* an element's border (between the content and the border) (e.g., **padding: 15px;**).
- **margin**: Adds space *outside* an element's border (between this element and others) (e.g., **margin: 10px;**).

7. Activity: Styling Your Web Page! (15-20 minutes)

1. **Open mypage.html** (or your similar simple HTML file) in your text editor.
2. **Choose a Method:** We recommend using an **External Stylesheet** for best practice.
3. **Create CSS File:** Create a *new* file in your text editor. Save it as **styles.css in the same folder** as your **mypage.html** file.
4. **Link CSS to HTML:** In **mypage.html**, add the **<link>** tag inside the **<head>** section:

```
<head>
    <title>My Awesome Page</title>
    <link rel="stylesheet" href="styles.css">
</head>
```

5. **Add Styles to styles.css:** Open **styles.css** and add some rules. Try these or experiment:

```
/* Style the whole page */
body {
    font-family: Arial, Helvetica, sans-serif; /* Use common sans-serif fonts */
    background-color: #e9f5ff; /* Light blue background */
    margin: 20px; /* Add some space around the edges */
}

/* Style the main heading */
h1 {
    color: #0056b3; /* Dark blue color */
    text-align: center; /* Center the heading */
    border-bottom: 2px solid #0056b3; /* Add an underline */
    padding-bottom: 10px; /* Space between text and underline */
}

/* Style the paragraphs */
p {
    color: #333; /* Dark gray text */
    font-size: 16px;
    line-height: 1.6; /* More space between lines for readability */
}

/* Style the link */
a {
    color: #d9534f; /* Reddish color for link */
    text-decoration: none; /* Remove the default underline */
}

a:hover { /* Style link when mouse hovers over it */
    color: #c9302c; /* Darker red on hover */
    text-decoration: underline; /* Add underline on hover */
}

/* Style the image */
img {
    border: 3px solid #ccc; /* Add a gray border */
    padding: 5px; /* Space between image and border */
    display: block; /* Make it a block element */
    margin: 15px auto; /* Add space above/below, and center it */
    max-width: 100%; /* Ensure image isn't wider than its container */
    height: auto; /* Maintain aspect ratio */
}
```

6. **Save Both Files:** Make sure both **mypage.html** and **styles.css** are saved.
7. **Refresh in Browser:** Open (or refresh) **mypage.html** in your web browser. You should see the styles applied! Experiment by changing values in **styles.css** and refreshing the browser.

8. Review & Recap (5 minutes)

- What does CSS stand for? (Cascading Style Sheets)
- What is its main purpose? (To style the presentation and layout of HTML)
- What are the parts of a CSS rule? (Selector and Declaration Block {property: value;})
- What are the 3 ways to add CSS? (Inline, Internal, External) Which is usually best? (External)
- What does a class selector start with? (**.**)
- What does an ID selector start with? (**#**)
- Name 3 CSS properties you learned. (e.g., **color**, **background-color**, **font-size**, **margin**, **padding**, **border**)

9. Next Steps & Further Learning:

- Explore more CSS selectors (attribute selectors, pseudo-classes like **:hover**, pseudo-elements like **::before**).
- Learn about the CSS Box Model (content, padding, border, margin) in detail.
- Dive into CSS Layout techniques (Flexbox, Grid).
- Understand CSS Specificity and the Cascade.
- Practice using your browser's Developer Tools (usually F12) to inspect elements and experiment with CSS live.
- Continue exploring resources like MDN Web Docs, freeCodeCamp, and W3Schools.