Day 2: Styling Text, Colors, and Layout

Lesson Objective:

By the end of this lesson, learners should be able to:

- Style text using various properties.
- Use color systems (named colors, hex, RGB, RGBA) to apply colors to elements.
- Apply and position backgrounds using CSS.
- Add borders and shadows to elements for visual enhancement.
- Understand and use the `display` property to control layout.

Lesson Outline:

1. Text Properties
2. Color Systems
3. Backgrounds
4. Borders and Shadows
5. Display Property
6. Practice Project

---

1. Text Properties

CSS provides several properties for styling text. Today, we'll focus on:

- **font-family**
- **font-size**
- **font-weight**
- **line-height**
- **text-align**

1.1 font-family

The `font-family` property specifies the typeface to use for the text. You can specify multiple fonts as fallbacks, in case a user's device doesn't support a particular font.

The `font-weight` property defines the thickness of the text, ranging from `100` (thin) to `900` (bold).

1.4 line-height

The `line-height` property controls the space between lines of text, improving readability.

1.5 text-align

The `text-align` property aligns the text within its container (left, center, right, justify).

> **Activity:** Create a paragraph with multiple lines of text. Use `font-family`, `font-size`, `font-weight`, `line-height`, and `text-align` to style it. Experiment with different fonts and sizes.

---

2. Color Systems

CSS offers several ways to apply colors to text and elements. We'll explore the main systems:

2.1 Named Colors

CSS has a list of pre-defined color names that you can use directly.

2.2 Hexadecimal Colors

The `#RRGGBB` format represents colors with hexadecimal values. Each pair of characters corresponds to red, green, and blue (e.g., `#ff0000` is red).

2.3 RGB Colors

The `rgb()` function allows you to define colors by specifying the red, green, and blue values (0 to 255)

2.4 RGBA Colors

The `rgba()` function is similar to `rgb()` but includes an alpha channel for opacity (0 = fully transparent, 1 = fully opaque).

> **Activity:** Create a heading and paragraphs. Apply different colors using named colors, hex values, RGB, and RGBA. Try making some text semi-transparent using RGBA.

---

3. Backgrounds

CSS allows you to control backgrounds with various properties. Let's explore:

3.1 Background Color

You can set the background color of an element using the `background-color` property.

3.2 Background Images

You can use an image as the background of an element with `background-image`. The image can be repeated or positioned differently.

```
div {
    background-image: url('image.jpg');
    background-repeat: no-repeat;
    background-position: center;
}
```

3.3 Background Repeat

By default, background images repeat. You can control this behavior using the `background-repeat` property.

```
div {
    background-repeat: no-repeat; /* prevent repeating */
}
```

3.4 Background Position

The `background-position` property allows you to position the background image within the element.

```
div {
    background-position: center center; /* center horizontally and
vertically */
}
```

> **Activity:** Create a `div` with some text inside. Set a background color, and then try using a background image. Experiment with `background-repeat` and `background-position`.

---

4. Borders and Shadows

Borders and shadows can add depth to elements and make them visually appealing.

4.1 Border

The `border` property allows you to add borders around an element. You can control the width, style, and color.

4.2 border-radius

The `border-radius` property is used to round the corners of an element.

4.3 box-shadow

The `box-shadow` property adds a shadow effect to an element.

```
div {
    box-shadow: 5px 5px 10px rgba(0, 0, 0, 0.5);
}
```

> **Activity:** Create a `div` with a background color. Add a border around it and use `border-radius` to round the corners. Then, add a `box-shadow` to give the element depth.

---

5. Display Property

The `display` property controls how an element is displayed on the page. The main values we'll cover are:

- **block:** The element takes up the full width available (e.g., `<div>`, `<p>`).
- **inline:** The element takes up only as much width as necessary (e.g., `<span>`, `<a>`).
- **inline-block:** The element behaves like an inline element but allows width and height to be set.
- **none:** The element is not displayed.

5.1 block

Block-level elements start on a new line and take up the full width.

5.2 inline

Inline elements flow with the text and don’t start on a new line.

5.3 inline-block

An inline-block element behaves like an inline element but can have its width and height set.

5.4 none

This hides the element from the page.

> **Activity:** Create multiple elements (`div`, `span`, `button`) and experiment with changing their `display` properties to `block`, `inline`, `inline-block`, and `none`. Observe how their behavior and layout change.

---

6. Practice Project

Let’s bring all of today’s concepts together into a mini project.

Goal:

Create a styled card with text, background, borders, and shadows.

Steps:
HTML structure:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <link rel="stylesheet" href="styles.css">
    <title>Styled Card</title>
</head>
<body>
    <div class="card">
        <h1>Welcome to CSS</h1>
```

```
        <p class="description">CSS helps you style web pages with
colors, layouts, and text.</p>
        <button class="cta-button">Learn More</button>
    </div>
</body>
</html>
```

CSS (styles.css):

```
/* Style the card */
.card {
    width: 300px;
    padding: 20px;
    margin: 0 auto;
    background-color: #f4f4f4;
    border-radius: 10px;
    box-shadow: 0px 5px 15px rgba(0, 0, 0, 0.1);
    text-align: center;
}

/* Style the heading */
h1 {
    font-family: 'Arial', sans-serif;
    font-size: 1.5rem;
    color: #333;
}

/* Style the description */
.description {
    font-size: 1rem;
    line-height: 1.6;
    color: #555;
}

/* Style the button */
.cta-button {
    display: inline-block;
    padding: 10px 20px;
    background-color: #007BFF;
    color: white;
    border: none;
    border-radius: 5px;
    text-align: center;
```

```
    font-size: 1rem;
}

.cta-button:hover {
    background-color: #0056b3;
}
```