CHAPTER 3

CASCADING STYLE SHEET

Introduction to CSS

Cascading Style Sheets (CSS) is a style sheet language used to control the presentation of web pages, including layout, colors, fonts, and spacing. It is one of the core technologies of the web, alongside HTML and JavaScript. While **HTML** provides the structure of a web page, **CSS** is used to define the visual style of that structure.

1. What is CSS?

CSS is used to style HTML elements by applying rules to them. These rules define how the elements should be displayed, including properties like:

- **Colors**
- **Fonts**
- **Spacing**
- **Positioning**
- **Layout** of the web page

2. CSS Syntax

A CSS rule consists of two main parts:

- **Selector**: The HTML element(s) that the style will be applied to.
- **Declaration Block**: One or more declarations enclosed in curly braces `{}`. Each declaration contains a **property** and a **value**.

Example:

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

Explanation:

- **Selector**: `p` (applies to all <p> elements)
- **Declaration Block**: `{ color: blue; font-size: 16px; }`
    - **Property**: `color`, `font-size`
    - **Value**: `blue`, `16px`

3. Ways to Apply CSS

There are three main ways to apply CSS to a webpage:

a. Inline CSS

This is applied directly within an HTML element using the `style` attribute.

```
html
CopyEdit
<p style="color: red; font-size: 18px;">This is an inline CSS example.</p>
```

- **Advantages**: Quick and easy for small changes.
- **Disadvantages**: Difficult to maintain for large websites.

b. Internal CSS

This is written inside the `<style>` tag in the `<head>` section of the HTML document.

```
html
CopyEdit
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Internal CSS Example</title>
  <style>
    body {
      background-color: lightblue;
      font-family: Arial, sans-serif;
    }
    h1 {
      color: green;
    }
  </style>
</head>
<body>
  <h1>Welcome to the Webpage</h1>
  <p>This page uses internal CSS.</p>
</body>
</html>
```

- **Advantages**: Better for styling multiple elements on a single page.
- **Disadvantages**: Only applies to the current document.

c. External CSS

This method uses an external CSS file linked to the HTML document. The external CSS file has a `.css` extension.

```
html
CopyEdit
<!-- Link to external CSS file -->
<link rel="stylesheet" href="styles.css">

<!-- HTML content -->
<h1>Hello, World!</h1>
<p>This page is styled using external CSS.</p>
```

The **styles.css** file:

```
css
CopyEdit
h1 {
  color: blue;
  font-size: 24px;
}
```

- **Advantages**: Reusable across multiple HTML pages, making maintenance easier.
- **Disadvantages**: Requires an additional HTTP request to load the CSS file.

4. CSS Selectors

CSS selectors define which HTML elements a rule will be applied to. There are different types of selectors:

a. Universal Selector (`*`)

The universal selector applies to all elements on the page.

```
css
CopyEdit
* {
  color: black;
}
```

b. Type Selector

A type selector targets all elements of a specific type (e.g., all `<h1>` elements).

```
css
CopyEdit
h1 {
  color: red;
}
```

c. Class Selector (`.`)

A class selector applies styles to any element with a specific class attribute. It is prefixed by a period `.`.

```
css
CopyEdit
.myclass {
  font-size: 18px;
}
```

In HTML:

```
html
CopyEdit
<p class="myclass">This is a paragraph with class "myclass".</p>
```

d. ID Selector (`#`)

An ID selector targets a unique element with a specific `id` attribute. It is prefixed by a hash `#`.

```
css
CopyEdit
#myid {
  color: blue;
}
```

In HTML:

```
html
CopyEdit
<p id="myid">This is a paragraph with id "myid".</p>
```

e. Attribute Selector

You can also select elements based on their attributes.

```
css
CopyEdit
input[type="text"] {
  background-color: lightgray;
}
```

5. CSS Properties

CSS has a wide variety of properties to control the appearance of elements. Here are some common ones:

a. Text Properties

- `color`: Sets the text color.
- `font-size`: Sets the font size.
- `font-family`: Defines the font.
- `text-align`: Aligns text (left, center, right).

```
css
CopyEdit
p {
  color: black;
  font-size: 16px;
  text-align: center;
}
```

b. Box Model Properties

The CSS box model consists of content, padding, border, and margin.

- **`padding`**: Space inside the element, around the content.
- **`border`**: The border surrounding the element.
- **`margin`**: Space outside the element.

```
css
CopyEdit
div {
  padding: 20px;
  border: 1px solid black;
  margin: 10px;
}
```

c. Background Properties

You can change the background color, image, and position of elements.

```
css
CopyEdit
body {
  background-color: lightblue;
}

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

d. Layout Properties

- `width` and `height`: Set the width and height of elements.
- `display`: Defines the display behavior of an element (`block`, `inline`, `flex`, etc.).

```
css
CopyEdit
```

```
div {
  width: 50%;
  height: 300px;
  background-color: lightgreen;
  display: block;
}
```

6. CSS Box Model

The **CSS Box Model** represents the structure of a web element. It consists of:

1. **Content**: The actual content of the element (e.g., text or images).
2. **Padding**: Space between the content and the border.
3. **Border**: Surrounds the padding (if defined).
4. **Margin**: The outermost space around the element.

```
css
CopyEdit
div {
  width: 200px;
  padding: 20px;
  border: 5px solid black;
  margin: 10px;
}
```

This will create a box with content inside, 20px of padding, a 5px border, and a 10px margin around it.

7. CSS Colors

CSS allows you to specify colors in several formats:

- **Named colors**: `red`, `blue`, `green`
- **Hexadecimal**: `#FF5733` (RGB values in base 16)
- **RGB**: `rgb(255, 87, 51)`
- **RGBA** (RGB with alpha transparency): `rgba(255, 87, 51, 0.5)`
- **HSL** (Hue, Saturation, Lightness): `hsl(9, 100%, 60%)`

```
css
CopyEdit
h1 {
  color: #FF5733; /* Hexadecimal color */
}
```

8. CSS Positioning

Positioning defines how elements are placed on a web page. There are several positioning types:

a. Static (default):

The element is positioned according to the normal document flow.

```
css
CopyEdit
div {
  position: static;
}
```

b. Relative:

The element is positioned relative to its normal position.

```
css
CopyEdit
div {
  position: relative;
  top: 20px;
  left: 30px;
}
```

c. Absolute:

The element is positioned relative to the nearest positioned ancestor.

```
css
CopyEdit
div {
  position: absolute;
  top: 50px;
  right: 30px;
}
```

d. Fixed:

The element is positioned relative to the browser window and remains fixed when scrolling.

```
css
CopyEdit
div {
  position: fixed;
  top: 0;
  left: 0;
}
```

9. Responsive Web Design

CSS is a key part of making web pages responsive (adjusting to different screen sizes). This is done using **media queries**.

Example:

```
css
```

```
CopyEdit
@media screen and (max-width: 600px) {
  body {
    background-color: lightblue;
  }
  h1 {
    font-size: 24px;
  }
}
```

This CSS rule applies when the screen width is 600px or less, changing the background color and font size.