Day-2

Task-1 : Introduction to CSS

Introduction to CSS

CSS, or **Cascading Style Sheets**, is a stylesheet language used to describe the presentation of HTML elements. It controls the layout, design, and visual aspects of a web page, allowing developers to create attractive and user-friendly interfaces.

Why Use CSS?

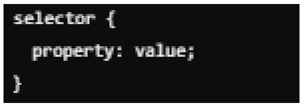
HTML structures content but lacks styling capabilities. CSS enhances web pages by:

- Controlling colors, fonts, and layouts.
- Making websites responsive for different screen sizes.
- Improving consistency and maintainability.

How CSS Works

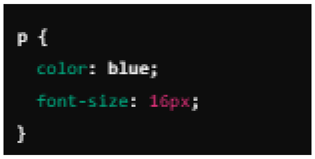
CSS applies styles to HTML elements using **selectors** and **rules**. These rules define the appearance and layout of specific elements.

Basic CSS Syntax:



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

Example:

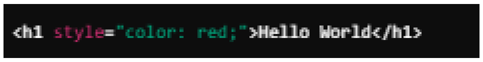


```
p {
  color: blue;
  font-size: 16px;
}
```

- **Selector**: Targets the element (`p`).
- **Property**: Specifies the style to apply (`color`, `font-size`).
- **Value**: Defines the appearance (`blue`, `16px`).

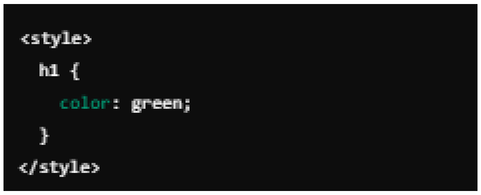
Ways to Apply CSS

1. **Inline CSS**: Directly within HTML elements.



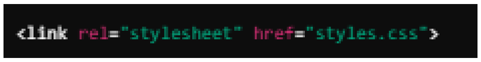
```
<h1 style="color: red;">Hello World</h1>
```

2. **Internal CSS**: In the `<style>` tag inside the `<head>` section.



```
<style>
  h1 {
    color: green;
  }
</style>
```

3. **External CSS**: Using an external stylesheet linked to the HTML file.



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

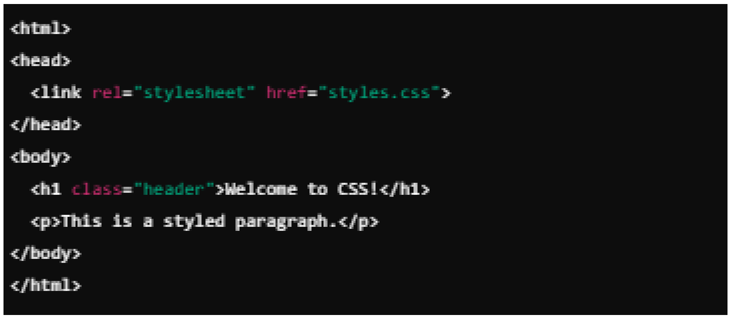
External stylesheets are reusable, keeping the code clean and manageable.

Key Features of CSS

- **Selectors**: Target elements to style.
  - `h1` (element selector), `.class-name` (class selector), `#id-name` (ID selector).
- **Box Model**: Defines layout using `margin`, `border`, `padding`, and `content`.
- **Responsive Design**: Ensures the website looks good on all devices using media queries and flexible layouts.

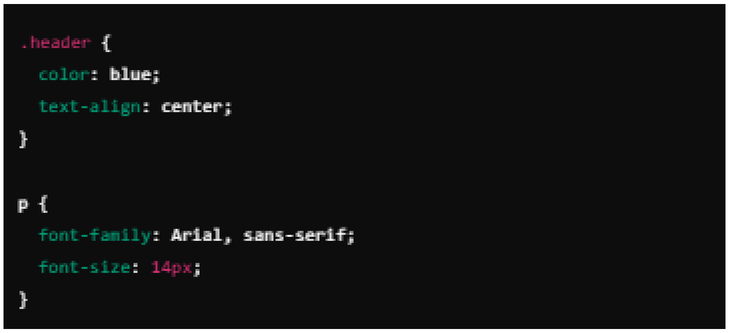
Example CSS Application

HTML:



```
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <h1 class="header">Welcome to CSS!</h1>
  <p>This is a styled paragraph.</p>
</body>
</html>
```

CSS (`styles.css`):



```
.header {
  color: blue;
  text-align: center;
}

p {
  font-family: Arial, sans-serif;
  font-size: 14px;
}
```

Task-2 : Styling Basics

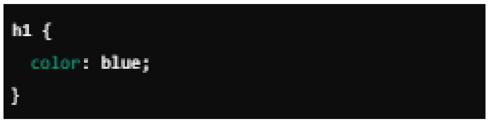
Styling Basics in CSS

Styling is the process of enhancing the visual appearance of HTML elements using CSS. It involves applying various properties to control aspects such as color, typography, spacing, layout, and responsiveness. Here's an overview of the basics:

1. CSS Selectors

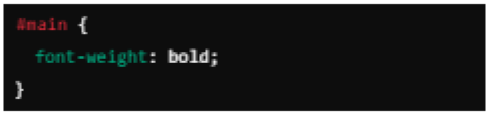
CSS selectors determine which HTML elements a style rule applies to.

- **Type Selector**: Targets all elements of a specific type.



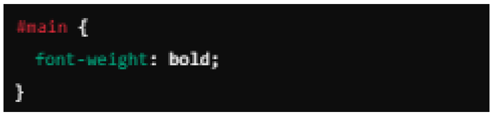
```
h1 {
  color: blue;
}
```

- **Class Selector**: Targets elements with a specific class (prefix with a .).



```
#main {
  font-weight: bold;
}
```

- **ID Selector**: Targets a single element with a specific ID (prefix with #).

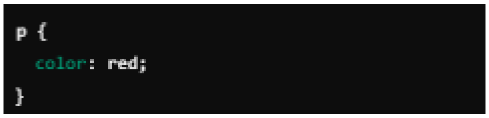


```
#main {
  font-weight: bold;
}
```

2. Text Styling

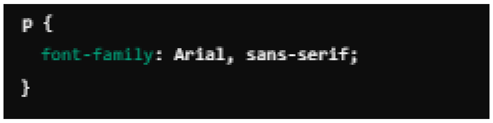
CSS provides properties to style text for better readability and visual appeal.

- **Color**: Sets the text color.



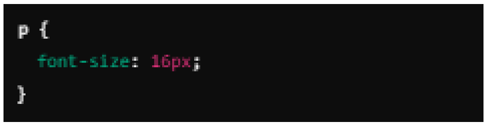
```
p {
  color: red;
}
```

- **Font Family**: Specifies the font style.



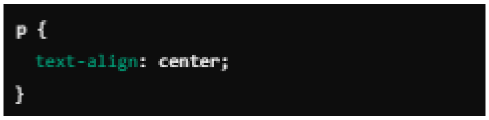
```
p {
  font-family: Arial, sans-serif;
}
```

- **Font Size**: Sets the size of the text.



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

- **Text Alignment**: Aligns text horizontally.

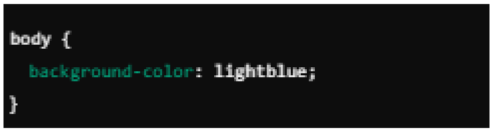


```
p {
  text-align: center;
}
```

3. Background Styling

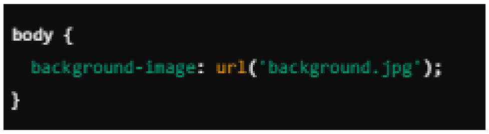
Background properties control the appearance behind elements.

- **Background Color**: Fills the element’s background.



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

- **Background Image**: Adds an image as the background.



```
body {
  background-image: url('background.jpg');
}
```

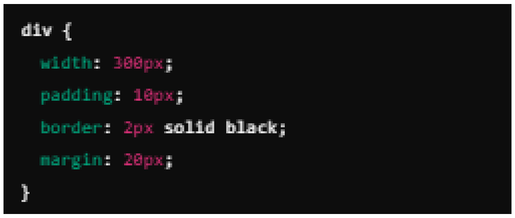
4. The Box Model

The **CSS Box Model** is a framework for layout and spacing. It consists of:

1. **Content**: The main content of the element.
2. **Padding**: Space between the content and the border.
3. **Border**: The edge around the padding.

4. **Margin**: Space outside the border.

Example:

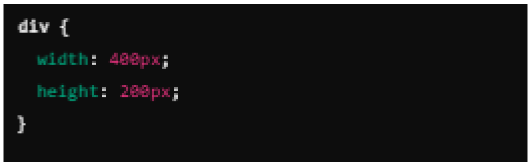


```
div {
  width: 300px;
  padding: 10px;
  border: 2px solid black;
  margin: 20px;
}
```

5. Layout Styling

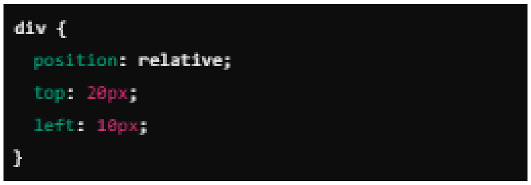
CSS layout properties help position and align elements on a page.

- **Width and Height**: Sets the size of an element.



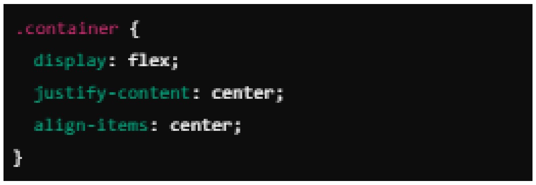
```
div {
  width: 400px;
  height: 200px;
}
```

- **Positioning**: Positions elements using `static`, `relative`, `absolute`, or `fixed`.



```
div {
  position: relative;
  top: 20px;
  left: 10px;
}
```

- **Flexbox**: A modern layout system for aligning items.



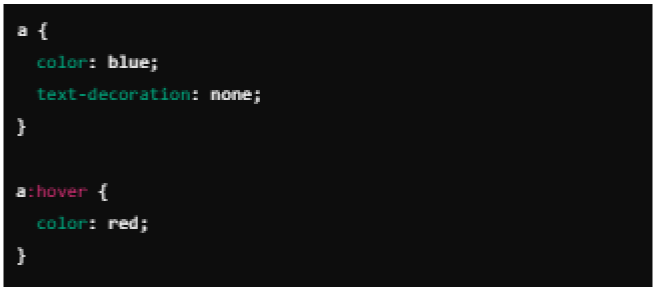
```
.container {
  display: flex;
  justify-content: center;
  align-items: center;
}
```

6. Styling Links

You can style links differently based on their state:

- `a:link`: Normal state.
- `a:visited`: After the link has been clicked.
- `a:hover`: When the mouse is over the link.
- `a:active`: When the link is clicked.

Example:



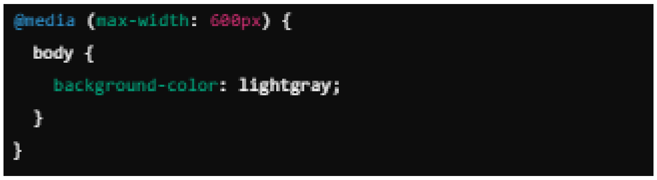
```
a {
  color: blue;
  text-decoration: none;
}

a:hover {
  color: red;
}
```

7. Responsive Styling

Responsive design ensures the webpage looks good on different devices.

- **Media Queries**: Apply styles based on screen size.



```
@media (max-width: 600px) {
  body {
    background-color: lightgray;
  }
}
```