CSS Notes

1. Introduction to CSS

What is CSS?

CSS (Cascading Style Sheets) is a stylesheet language used to describe the presentation of a document written in HTML or XML. It controls the layout, colors, fonts, and overall visual appearance of web pages.

How CSS works with HTML

CSS is used to separate content (HTML) from design (CSS). HTML provides the structure of the webpage, while CSS handles the presentation. CSS can be applied to HTML documents in three ways:

- **Inline CSS:** Styles are applied directly within an HTML element using the `style` attribute.
- **Internal CSS:** Styles are defined within the `<style>` tag in the `<head>` section of an HTML document.
- **External CSS:** Styles are defined in a separate `.css` file and linked to the HTML document using the `<link>` tag.

Inline, Internal, and External CSS

Inline CSS:

```
<p style="color: blue;">This is an inline-styled paragraph.</p>
```

Internal CSS:

```
<head>
    <style>
        p {
            color: blue;
        }
    </style>
</head>
<body>
    <p>This is an internal-styled paragraph.</p>
</body>
```

External CSS:

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

```
<body>
    <p>This is an external-styled paragraph.</p>
</body>
```

2. Basic Syntax

Selectors

Selectors are patterns used to select the elements you want to style.

- **Type Selector:** Targets elements by their type (e.g., `div`, `p`, `h1`).
- **Class Selector:** Targets elements by their class attribute (e.g., `.classname`).
- **ID Selector:** Targets an element by its ID attribute (e.g., `#idname`).

Properties

Properties define the specific style to apply (e.g., `color`, `font-size`, `margin`).

Values

Values are assigned to properties (e.g., `red`, `16px`, `10%`).

Comments in CSS

Comments are used to explain the code. They are not displayed on the webpage.

```
/* This is a single-line comment */
/* This is a
   multi-line comment */
```

3. CSS Selectors

Basic Selectors

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

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

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

```
.example {
    color: blue;
}
```

- **ID Selector:** Targets a single element with a specific ID.

```
#unique {
    color: green;
}
```

Grouping Selectors

Combine multiple selectors to apply the same styles to several elements.

```
h1, h2, h3 {
    color: purple;
}
```

Combinator Selectors

- **Descendant Selector:** Selects elements nested within other elements.

```
.container p {
    color: orange;
}
```

- **Child Selector:** Selects direct children of an element.

```
.parent > .child {
    color: pink;
}
```

- **Adjacent Sibling Selector:** Selects an element immediately following a specified element.

```
h1 + p {
    color: grey;
}
```

- **General Sibling Selector:** Selects all elements that are siblings of a specified element.

```
h1 ~ p {
    color: brown;
}
```

Attribute Selectors

Targets elements based on the presence or value of attributes.

- **Presence:** `[type]`
- **Value:** `[type="text"]`
- **Partial Match:** `[type^="text"]` (starts with), `[type$="text"]` (ends with), `[type*="text"]` (contains)

Pseudo-class Selectors

Apply styles to elements based on their state or position.

- `:hover` (when hovering over an element)
- `:nth-child(n)` (targeting specific children)
- `:first-child` (the first child of its parent)

Pseudo-element Selectors

Target parts of elements.

- `::before` (insert content before an element's content)
- `::after` (insert content after an element's content)
- `::first-line` (style the first line of a block-level element)
- `::first-letter` (style the first letter of a block-level element)

4. CSS Box Model

Content

The actual content of the box, where text and images appear.

Padding

Space between the content and the border.

```
padding: 10px;
```

Border

Surrounds the padding (if any) and content.

```
border: 1px solid black;
```

Margin

Space outside the border. It clears space around the element.

```
margin: 20px;
```

Box Sizing

Defines how the width and height of an element are calculated.

- `content-box` (default, width and height include only the content area)
- `border-box` (width and height include content, padding, and border)

5. Text and Font Styling

Font Family

Specifies the typeface to use.

```
font-family: Arial, sans-serif;
```

Font Size

Defines the size of the font.

```
font-size: 16px;
```

Font Weight

Specifies the thickness of the font.

```
font-weight: bold;
```

Line Height

Controls the space between lines of text.

```
line-height: 1.5;
```

Text Alignment

Aligns text within an element.

```
text-align: center;
```

Text Decoration

Adds decoration to text, such as underlines.

```
text-decoration: underline;
```

Text Transformation

Controls the capitalization of text.

```
text-transform: uppercase;
```

Letter Spacing

Adjusts the space between letters.

```
letter-spacing: 2px;
```

6. Colors and Backgrounds

Color Values

- **Hex:** `#RRGGBB` (e.g., `#ff0000` for red)
- **RGB:** `rgb(r, g, b)` (e.g., `rgb(255, 0, 0)`)
- **RGBA:** `rgba(r, g, b, a)` (e.g., `rgba(255, 0, 0, 0.5)`)
- **HSL:** `hsl(h, s, l)` (e.g., `hsl(0, 100%, 50%)`)
- **HSLA:** `hsla(h, s, l, a)` (e.g., `hsla(0, 100%, 50%, 0.5)`)

Background Color

Sets the background color of an element.

```
background-color: lightblue;
```

Background Image

Sets an image as the background of an element.

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

Background Size

Controls the size of the background image.

```
background-size: cover;
```

Background Repeat

Controls if and how the background image repeats.

```
background-repeat: no-repeat;
```

Background Position

Specifies the position of the background image.

```
background-position: center;
```

Gradient Backgrounds

Creates a gradient as the background image.

```
background: linear-gradient(to right, red, yellow);
```

7. Layout Techniques

Display Property

Defines how elements are displayed.

- `block` (occupies the full width, starts on a new line)
- `inline` (occupies only the space it needs, does not start on a new line)
- `inline-block` (like inline but can set width and height)
- `none` (the element is not displayed)

Positioning

- `static` (default, normal flow)
- `relative` (relative to its normal position)

- `absolute` (relative to the nearest positioned ancestor)
- `fixed` (relative to the viewport)
- `sticky` (sticky positioning, switches between relative and fixed)

Float and Clear

- `float` (positions an element to the left or right, allowing text to wrap around it)
- `clear` (specifies which sides of an element's box are not allowed to float)

Flexbox

Container Properties

- `display: flex` (enables flexbox on the container)
- `flex-direction` (defines the direction of the flex items)
- `justify-content` (aligns items on the main axis)
- `align-items` (aligns items on the cross axis)
- `flex-wrap` (defines whether items should wrap or not)

Item Properties

- `flex-grow` (defines the ability of a flex item to grow)
- `flex-shrink` (defines the ability of a flex item to shrink)
- `flex-basis` (defines the default size of a flex item)
- `align-self` (overrides `align-items` for individualS