

INTERNSHIP REPORT

WEEK 2 DAY 2

Internship department:	Information and Technology/ Media	Submitted to:	alihydershar688@gmail.com
Internship Employer Name:	Ali Hyder	Submission Date:	1 st July, 2025
Program:	Front Development	Semester:	6 TH
Student Name:	Yasal Qamar	Reg. No.	B22F0338CS109

CSS Selectors, Colors, Units, and Typography

Objective of the Day

To explore and practice essential concepts of CSS including various **selectors**, **colors styling**, **CSS units** (like px, %, rem), and **typographic rules** such as font-family, font-size, line-height, and text alignment. These are core to creating visually appealing and responsive web pages.

1. CSS Selectors:

Firstly, I learned about **CSS Selectors**, which are used to select and style specific HTML elements.

CSS selectors help apply consistent styling rules by targeting elements based on their type, class, ID, or other attributes.

I explored the following types of CSS selectors:

- 1) **Element Selector: p** – Targets all paragraph elements.
- 2) **Class Selector: .box** – Targets all elements with a specific class.
- 3) **ID Selector: #unique** – Targets a single element with a specific ID.
- 4) **Universal Selector: *** – Targets **all** elements on the page.
- 5) **Grouping Selector: h1, h2, h3, p** – Targets multiple types of elements with a shared style rule.

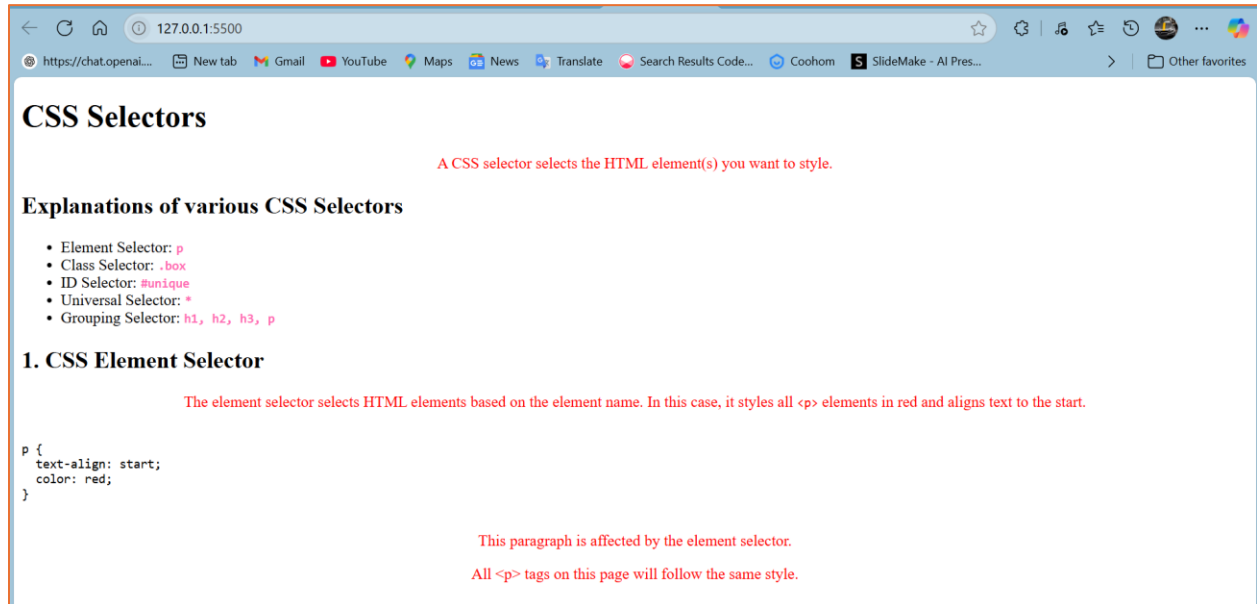
1. Element Selector

Selects HTML elements by name.

Example:

```
• p {  
•   text-align: center;  
•   color: red;  
• }
```

Output:



2. Class Selector

Select elements with a specific class.

Example:

```
• .box {  
•   background-color: lightgreen;  
•   border: 2px dashed green;  
•   padding: 15px;  
•   width: 250px;  
•   margin-bottom: 10px;  
• }
```

Output:

127.0.0.1:5500

https://chat.openai... New tab Gmail YouTube Maps News Translate Search Results Code... Coohom SlideMake - AI Pres...

CSS Selectors

A CSS selector selects the HTML element(s) you want to style.

Explanations of Various CSS Selectors

- Element Selector: `p`
- Class Selector: `.box`
- ID Selector: `#unique`
- Universal Selector: `*`
- Grouping Selector: `h1, h2, h3, p`

2. CSS Class Selector

The class selector targets **any element** with a specific `class` attribute. In this case, we apply styles to any element with `class="box"`.

This box is styled using the class selector `.box`.

```
.box {  
  background-color: lightgreen;  
  border: 2px dashed green;  
  padding: 15px;  
  width: 250px;  
}
```

This paragraph is not styled because it doesn't have the class.

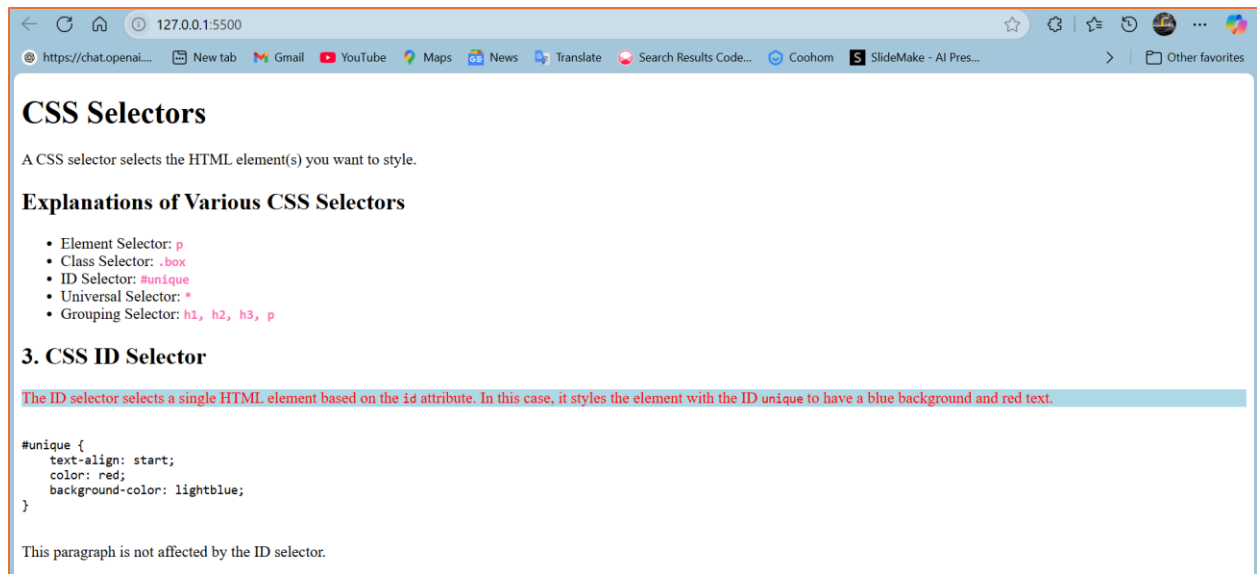
3. ID Selector:

Targets a single unique element using an ID.

Examples

```
#unique {  
  text-align: start;  
  color: red;  
  background-color: lightblue;  
}
```

Output:



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5500'. The page title is 'CSS Selectors'. Below the title, a paragraph states: 'A CSS selector selects the HTML element(s) you want to style.' This is followed by a section titled 'Explanations of Various CSS Selectors' which contains a bulleted list: 'Element Selector: p', 'Class Selector: .box', 'ID Selector: #unique', 'Universal Selector: *', and 'Grouping Selector: h1, h2, h3, p'. Below this list is a section titled '3. CSS ID Selector'. A light blue highlighted box contains the text: 'The ID selector selects a single HTML element based on the id attribute. In this case, it styles the element with the ID unique to have a blue background and red text.' Below the highlight is a code block containing the CSS:

```
#unique {
  text-align: start;
  color: red;
  background-color: lightblue;
}
```

 At the bottom of the browser window, a paragraph reads: 'This paragraph is not affected by the ID selector.'

CSS Selectors

A CSS selector selects the HTML element(s) you want to style.

Explanations of Various CSS Selectors

- Element Selector: `p`
- Class Selector: `.box`
- ID Selector: `#unique`
- Universal Selector: `*`
- Grouping Selector: `h1, h2, h3, p`

3. CSS ID Selector

The ID selector selects a single HTML element based on the `id` attribute. In this case, it styles the element with the ID `unique` to have a blue background and red text.

```
#unique {
  text-align: start;
  color: red;
  background-color: lightblue;
}
```

This paragraph is not affected by the ID selector.

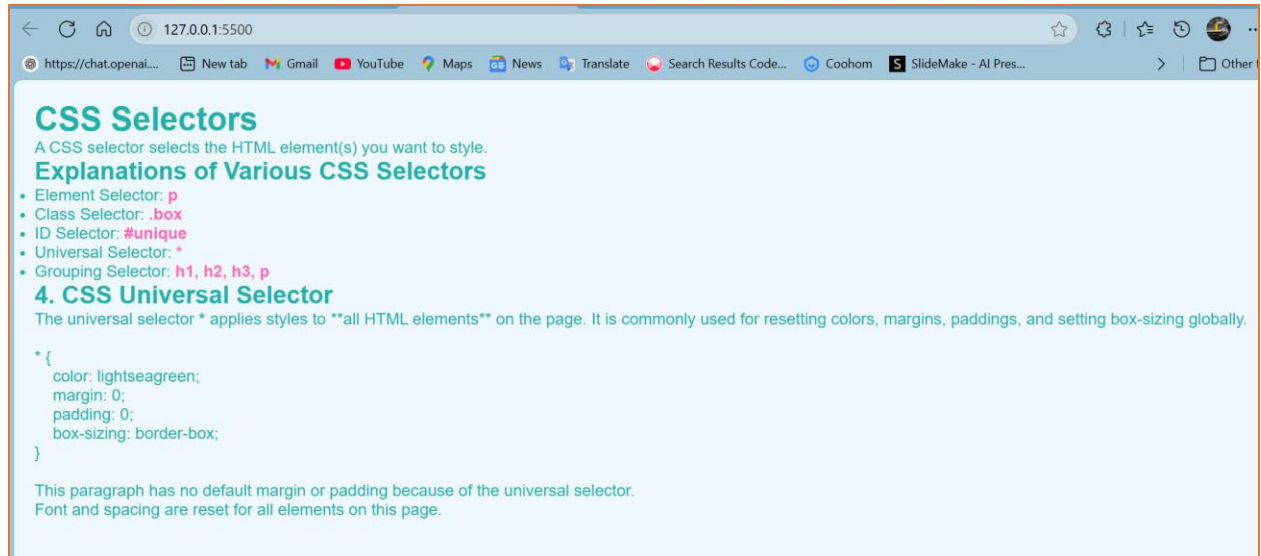
4. Universal Selector

Apply styles to **all HTML elements** on the page.

Example:

```
* {
  color: lightseagreen;
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: Arial, sans-serif;
}
```

Output:



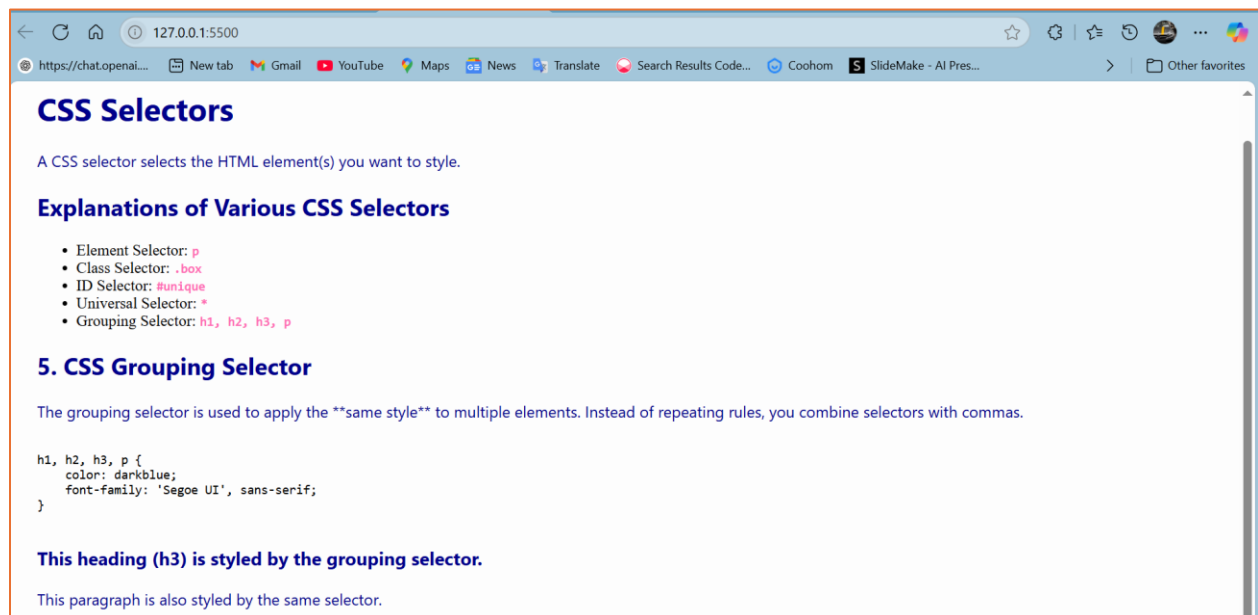
5. Grouping Selector

Applies the same style to multiple elements at once.

Example:

```
h1, h2, h3, p {
  color: darkblue;
  font-family: 'Segoe UI', sans-serif;
}
```

Output:



2. CSS Colors:

In CSS, colors are used to style the look and feel of elements like backgrounds, text, and borders. You can apply colors using names, HEX codes, RGB, RGBA, HSL, and HSLA formats.

- a. **CSS COLOR NAMES**
- b. **CSS COLOR VALUES**

CSS COLOR NAMES:

In CSS, a color can be specified by using a predefined color name:

The following are some of the color names you can use in CSS:

- a. **CSS Backgrounds Color**
 - b. **CSS Text Color**
 - c. **CSS Broder Color**
- **CSS Backgrounds Color:**

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

CSS COLOR background

The background color of an element can be set using the 'background-color' property in CSS. For example:

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

- **CSS Text Color:**

```
body {  
    color: hotpink;  
}
```

CSS COLOR text

The text color of an element can be set using the `color` property in CSS . For example:

```
{
  color: hotpink;
}
```

- **CSS Broder Color:**

```
body {
  border:5px solid Violet
}
```

CSS COLOR border

The border color of an element can be set using the `border-color` property in CSS. For example:

```
body {
  border:5px solid Violet;
}
```

- **CSS COLOR VALUES:**

CSS supports multiple types of color values:

Type	Example
Named	red, blue, orange
HEX	#FF5733, #000000
RGB	rgb(255, 0, 0)
RGBA	rgba(255, 0, 0, 0.5)
HSL	hsl(120, 100%, 50%)
HSLA	hsla(120, 100%, 50%, 0.3)

```
/* CSS Color Values */
.named-color
{
  color: blue;
}
```

```
.hex-color
{
  color: #FF5733;
}
.rgb-color
{
  color: rgb(0, 128, 0);
}
.rgba-color
{
  color: rgba(255, 0, 0, 0.5);
}
.hsl-color
{
  color: hsl(210, 100%, 50%);
}
.hsla-color
{
  color: hsla(270, 100%, 50%, 0.5);
}
```


CSS COLOR VALUES

CSS supports several color value formats:

- RGB: `rgb(255, 0, 0)`
- HEX: `#FF0000`
- HSL: `hsl(0, 100%, 50%)`
- RGBA: `rgba(255, 0, 0, 1)`
- HSLA: `hsla(0, 100%, 50%, 1)`

CSS also supports several color value formats:

This uses a named color (blue).

This uses a HEX color (`#FF5733`).

This uses an RGB color (green).

This uses an RGBA color (semi-transparent red).

This uses an HSL color (sky blue).

This uses an HSLA color (semi-transparent purple).

3. CSS Units

- CSS has several different units for expressing a length.
- Many CSS properties take "length" values, such as `width`, `margin`, `padding`, `font-size`, etc.
- **Length** is a number followed by a length unit, such as `10px`, `2em`, etc.
- For some CSS properties, negative lengths are allowed.

Types of length:

- There are two types of length units: **absolute** and **relative**.

1. Absolute Units

These are **fixed** and always represent the same size, regardless of screen/device.

Unit	Description	Example
px	Pixels	font-size: 16px;
cm	Centimeters	width: 10cm;
mm	Millimeters	padding: 5mm;
in	Inches	width: 2in;
pt	Points (1pt = 1/72 inch)	font-size: 12pt;
pc	Picas (1pc = 12pt)	margin: 1pc;

Examples:

```
.absolute {  
  width: 200px;  
  height: 100px;  
  background-color: lightcoral;  
}
```

Examples of absolute units:

These units are not affected by other elements on the page:
px, cm, mm, in, pt, pc

```
.absolute {  
  width: 200px;  
  height: 100px;  
  background-color: lightcoral;  
}
```

2. Relative Units

These depend on the context (such as the size of the parent element or viewport).

Unit	Description	Example
------	-------------	---------

%	Relative to parent element	width: 50%;
em	Relative to the font-size of the element	font-size: 2em;
rem	Relative to the root (<html>) font-size	margin: 1rem;
vw	Relative to 1% of the viewport width	width: 80vw;
vh	Relative to 1% of the viewport height	height: 50vh;
vmin	Smaller of vw and vh	font-size: 5vmin;
vmax	Larger of vw and vh	padding: 3vmax;

Examples:

```
.relative {
  width: 50%;
  font-size: 1.5em;
  background-color: lightgreen;
}
```

Examples of relative units:

These units are affected by other elements on the page: **%, em, rem, vw, vh**

```
.relative {
  width: 50%;
  font-size: 1.5em;
  background-color: lightgreen;
}
```

4. CSS TYPOGRAPHY:

Typography in CSS refers to the way **text** is styled and displayed on a webpage. It helps improve **readability**, **visual hierarchy**, and **user experience**. Good typography makes websites look clean and professional.

Property	Purpose	Example
font-family	Defines the font type	font-family: Arial, sans-serif;
font-size	Sets the size of the text	font-size: 18px;
font-weight	Controls boldness (normal, bold, 100–900)	font-weight: bold;
font-style	Sets style like italic or normal	font-style: italic;

text-align	Aligns text (left, center, right, justify)	text-align: center;
line-height	Controls space between lines	line-height: 1.5;
letter-spacing	Adds space between letters	letter-spacing: 1px;
text-decoration	Adds underline, overline, or line-through	text-decoration: underline;
text-transform	Changes case (uppercase, lowercase, capitalize)	text-transform: uppercase;

Examples:

```
.typography-example {
  font-family: 'Segoe UI', sans-serif;
  font-size: 20px;
  font-weight: 500;
  font-style: italic;
  text-align: justify;
  line-height: 1.6;
  letter-spacing: 1px;
  text-transform: capitalize;
  color: #333;
}
```

OUTPUT:

CSS Typography Properties

Example Text

This Is A Demonstration Of Css Typography. Proper Typography Improves Readability And Design Aesthetics. It Includes Properties Like Font Family, Size, Weight, Style, Alignment, Line Height, Letter Spacing, And Text Transformation. Using These Properties Effectively Can Enhance The User Experience On Web Pages.

```
.typography-example {
  font-family: 'Segoe UI', sans-serif;
  font-size: 20px;
  font-weight: 500;
  font-style: italic;
  text-align: justify;
  line-height: 1.6;
  letter-spacing: 1px;
  text-transform: capitalize;
  color: #333;
}
```

Conclusion:

- On Day 6 of the internship, I gained a solid understanding of key CSS fundamentals that are essential for front-end development.
- I explored various **CSS Selectors** including element, class, ID, universal, and grouping selectors and practiced how each one targets HTML elements differently for styling purposes.

- In addition, I learned about **CSS Colors**, including how to apply background colors, text colors, and border colors using various color values like named colors, HEX, RGB, RGBA, and HSL formats.
 - Moving forward, I explored **CSS Units**, both **absolute** (like px, cm, in) and **relative** (like %, em, rem, vw, vh), and how they help control layout, spacing, and responsiveness.
 - Finally, I studied **CSS Typography**, which covered properties like font-family, font-size, font-weight, line-height, text-align, and more — all of which enhance readability and design consistency.
 - Overall, today's session helped me understand how to use CSS to build clean, well-structured, and visually appealing web pages. These skills are fundamental for creating responsive and professional front-end designs, and I'm excited to apply them in future tasks during the internship.
-