📘 CSS Fundamentals

1. ⚙️ All About External Styling

The **External Style Sheet** is the preferred method for styling modern websites, separating **structure (HTML)** from **presentation (CSS)**.

Aspect	Description
File Creation	Styles are written in a separate file with a **.css** extension (e.g., styles.css).
Advantage	**Maintainability:** Change one CSS file to update the look of multiple HTML pages.
Implementation	The <link> tag is placed inside the HTML <head> section to connect the files.

Implementation Code

HTML File (**index.html**)

```
HTML
<!DOCTYPE html>
<html lang="en">
<head>
    <title>External Styling Demo</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <h1>My Main Heading</h1>
    <p>This paragraph is styled by the external sheet.</p>
</body>
```

```
</html>
```

CSS File (styles.css)

CSS

```
body {
    font-family: Arial, sans-serif;
}
h1 {
    color: navy;
    border-bottom: 2px solid lightgray;
}
```

---

2. 🎯 All About Selectors

Selectors are patterns used to choose the HTML elements to which CSS rules will be applied.

Simple Selectors

Selector Type	Symbol	Targets	Example Syntax	HTML Element Example
Universal	*	**All** elements on the page.	* { margin: 0; padding: 0; }	Selects all elements.
Element	(None)	All instances of a specific **HTML tag**.	p { font-size: 16px; }	Selects all <p> tags.
Class	.	All elements with a specific **class** attribute value. **(Reusable)**	.alert { color: red; }	...

ID	#	The **single unique element** with a specific **id** attribute value. **(Unique)**	#sidebar { width: 300px; }	...

Combinator & Group Selectors

Selector Type	Operator	Targets	Example Syntax
Descendant	(Space)	Elements that are **inside** another element (any nesting level).	div p { color: green; }
Group	, (Comma)	Groups multiple selectors to apply the **exact same style** to all of them.	h1, h2, .title { font-weight: bold; }

Code Example

CSS

```
/* Universal selector: reset defaults */
* {
    box-sizing: border-box;
}

/* Element selector: applies to all <li> */
li {
    list-style: none;
}

/* Group selector: applies style to two different tags */
h2, h3 {
    text-decoration: underline;
}

/* Descendant selector: only <li> items *inside* a <nav> will be blue */
```

```
nav li {
    color: blue;
}
```

---

3. 📦 Box Model

The **CSS Box Model** defines how the browser renders elements, including the space taken by the **Content**, **Padding**, **Border**, and **Margin**.

Component	Position	Effect (Standard Model)	CSS Properties
Content	Innermost	Defined by width and height.	width, height
Padding	Inside Border	**Adds** to the total element size.	padding
Border	Separator	**Adds** to the total element size.	border
Margin	Outermost	Does **NOT** add to the element's size, but controls separation from other elements.	margin

box-sizing Property

Value	Description

content-box (Default)	width/height = **Content** size. Padding & Border are **added** to this.
border-box (Recommended)	width/height = **Content + Padding + Border** size. Padding & Border are **included inside** the set dimensions.

Code Example

CSS

```
.box-demo {
    /* Content size */
    width: 200px;
    height: 100px;

    /* Padding: space inside the border (10px on all sides) */
    padding: 10px;

    /* Border: 5px line around the padding */
    border: 5px solid darkgreen;

    /* Margin: space outside the border (20px on all sides) */
    margin: 20px;

    /* Set to border-box: The final visible box (200x100) will INCLUDE
the 10px padding and 5px border.
       The Content area shrinks to 170x70 to accommodate them. */
    box-sizing: border-box;
}
```

---

4. 📏 CSS Units

CSS Units specify the size for properties, categorized as **Absolute** (fixed) or **Relative** (scaling).

Absolute Units

- **Fixed** values, not dependent on screen size or other elements.
- **Primary Unit: px (Pixels)**.

Relative Units

- **Scaling** values, dependent on other factors, essential for **responsive design**.

Unit	Name	Relative To	Recommended Use
rem	**Root Em**	The **font-size of the root element** (<html>).	Global typography for easy user scaling.
em	Em	The **font-size of the parent element**.	Component-specific scaling (e.g., buttons padding relative to its own font size).
vw	Viewport Width	**1%** of the **viewport's width**.	Making elements (like a headline) scale with the browser width.

Code Example

CSS

```
/* <html> element defines the baseline for 'rem' */
html {
    font-size: 16px;
}

.title {
    /* 3rem = 3 * 16px = 48px, scales if user changes browser root font size */
    font-size: 3rem;
}
```

```
.text-area {
    /* 100px is fixed, regardless of the user's font preferences */
    padding: 10px;

    /* 80vw = 80% of the viewport width, making it responsive */
    width: 80vw;
}
```

---

5. 🌊 Flow & Interaction Properties

These properties manage how content behaves when constrained and how the user interacts with elements.

overflow

Controls the content that exceeds an element's dimensions.

Value	Behavior
hidden	Content is clipped, the rest is invisible. **No scrollbar.**
scroll	Scrollbars are **always** added.
auto	Scrollbars are added **only if needed**.

text-overflow

Indicates clipped, inline content (usually text) with an ellipsis (...). Requires overflow: hidden; and white-space: nowrap;.

cursor

Defines the mouse pointer's appearance. **pointer** is commonly used for clickable items.

Code Example

```
CSS
```

```
.card-teaser {
    width: 200px;
    height: 100px;

    /* 1. Handling Overflowing Content */
    overflow: auto; /* Adds scrollbar if content is too tall */
}

.single-title {
    width: 150px;

    /* 2. Enabling text-overflow: ellipsis */
    overflow: hidden;         /* Hides any overflowing text */
    white-space: nowrap;      /* Forces text onto a single line */
    text-overflow: ellipsis;  /* Displays '...' at the clip point */
}

.clickable-icon {
    /* 3. Changing the cursor */
    cursor: pointer;
}
```

---

6. ✨ Pseudo-Classes & Pseudo-Elements

These advanced selectors style elements based on a special state or insert generated content.

Pseudo-Class: :hover

Styles an element when the user **mouses over** it.

Pseudo-Elements: ::before and ::after

Used to insert generated content **before** or **after** an element's actual content. **Requires the content property.**

Code Example

```
CSS
/* 1. Pseudo-Class: :hover for interaction */
```

```
.button {
    background-color: blue;
    transition: background-color 0.3s; /* Smooth transition for visual
feedback */
}

.button:hover {
    background-color: darkblue; /* Applies only when hovering */
    cursor: pointer;
}

/* 2. Pseudo-Elements: ::before for decoration */
.list-item::before {
    content: "➔ "; /* Inserts a small arrow before the content of every
.list-item */
    color: red;
    font-weight: bold;
}
```

📝 Practice Quiz Questions

Here are some quick questions to test your students' understanding of these topics:

1. **Selectors:** How would you write a single CSS rule to select all <h1> elements and all elements with the class note?
2. **External Styling:** What HTML tag and attributes are required to link an external stylesheet named main.css to an HTML document?
3. **Box Model:** If an element has width: 200px; padding: 20px; and box-sizing: content-box;, what is the total horizontal space the element takes up (excluding margin)?
4. **Box Model:** What is the primary benefit of setting box-sizing: border-box; globally with the universal selector (*)?
5. **Units:** Which CSS unit is best suited for responsive typography that scales relative to the browser's default font setting: px or rem? Why?
6. **Overflow:** Explain the difference in behavior between overflow: scroll; and overflow: auto; in a constrained container.
7. **Pseudo-Elements:** What CSS property is absolutely required when using the ::before or ::after pseudo-elements?
8. **External Styling:** To properly connect an external stylesheet to an HTML file, the `<link>` tag must be placed inside of which specific HTML section?
9. **Selectors:** What symbol must you place immediately before a name to target an element using a CSS **ID selector**?
10. **Selectors:** What specific punctuation mark is used to combine multiple selectors (like `h1` and `.warning`) so they all receive the exact same styles?

11. **Box Model:** Which component of the CSS Box Model is the transparent space located **between** the content and the border?
12. **Box Model:** To ensure that an element's padding and border do not increase its total size, you should set the `box-sizing` property to what value?
13. **Units:** Which relative unit, **`rem`** or **`vw`**, is used to set a size relative to the font size of the document's root `<html>` element?
14. **Flow:** If content is too large for a container, which `overflow` value should be used to clip the content but **only** display a scrollbar if the content actually overflows?
15. **Interaction:** To make the mouse cursor change to a hand icon when a user moves over a button, what value should the `cursor` property be set to?
16. **Pseudo-Classes:** Which pseudo-class should you use to change the background color of a link only when the user's mouse pointer is resting on it?
17. **Pseudo-Elements:** When using `::before` or `::after` to insert decorative content, what required CSS property must be included for the content to appear?