**DAY-14 TASK**

**1. What is a Pseudo-Class?**

A **pseudo-class** is used to style an element based on its **state, action, or position** — even if that state isn’t directly written in HTML.

**Hover effect**

<button>Click Me</button>

button {

background: purple;

color: white;

padding: 10px 20px;

border: none;

}

button:hover {

background: black;

transform: scale(1.1);

}

When you move your mouse over the button . it turns black and slightly enlarges.

**Focus effect**

<input type="text" placeholder="Enter your name">

input:focus {

border: 2px solid purple;

outline: none;

box-shadow: 0 0 5px purple;

}

When you click inside the input box .border glows purple.

**Table highlighting**

tr:nth-child(even) {

background: #f3e5f5;

}

Every even row of your table gets a light purple color automatically.

**Use in Real Websites**

* Highlight navigation menus on hover
* Change form field color when active
* Style first or last items in lists
* Animate buttons when clicked

**2. What is a Pseudo-Element?**

A **pseudo-element** lets you style **specific parts** of an element or **insert virtual content** without actually adding it to HTML.  
Think of it as adding “imaginary elements” inside your tag using CSS.

**Add icons using ::before**

<h2>Dashboard</h2>

h2::before {

content: "📋 ";

}

➡ Adds an icon before the text "Dashboard" without changing HTML.

**Add decoration line using ::after**

h2::after {

content: "";

display: block;

width: 100px;

height: 3px;

background: purple;

margin-top: 5px;

}

Adds a stylish underline below the heading.

**Style first letter**

p::first-letter {

font-size: 35px;

color: purple;

font-weight: bold;

}

Makes the first letter of a paragraph stand out .

**Use in Real Websites**

* Decorative effects (lines, icons)
* Adding quotes automatically
* Highlighting first letter or line of articles
* Styling user-selected text

**3. Difference Between Pseudo-Class and Pseudo-Element**

| **Feature** | **Pseudo-Class** | **Pseudo-Element** |
| --- | --- | --- |
| Controls | State or condition | Specific part of element |
| Symbol | : single colon | :: double colon |
| Example | a:hover, input:focus | p::first-letter, div::after |
| Real or virtual | Acts on real element | Creates virtual content |
| Used For | Hover, Focus, Active, nth-child | Before, After, First-letter, Selection |

**4. What is Position Property in CSS?**

The **position property** defines **how an element is placed** on a webpage — whether it stays in flow, moves freely, or sticks to the screen.

**🔹 Syntax**

position: value;

**🔸 1. Static (Default)**

All elements are **static by default** — they appear in the normal document flow.

div {

position: static;

}

You can’t move it using top/left/bottom/right.  
Example: Normal paragraph or div without special positioning.

**🔸 2. Relative**

Moves the element **relative to its original place**.

div {

position: relative;

top: 20px;

left: 30px;

}

It shifts slightly but still keeps its original space reserved.

**Use:** Small adjustments (like moving text or icon a few pixels).

**🔸 3. Absolute**

Moves element **out of normal flow** and positions it relative to the **nearest ancestor** that has position set (not static).

.parent {

position: relative;

}

.child {

position: absolute;

top: 20px;

right: 30px;

}

**Use:** Tooltips, dropdown menus, popups, badges on icons.

**🔸 4. Fixed**

Stays **fixed to the viewport** — does not move even when you scroll.

.chatbox {

position: fixed;

bottom: 20px;

right: 20px;

}

**Use:** “Back to top” buttons, sticky chat icons, floating menus.

**🔸 5. Sticky**

Acts **relative until scrolling**, then becomes **fixed** when reaching a certain position.

header {

position: sticky;

top: 0;

background: white;

}

**Use:** Sticky navbar that stays at top while scrolling.

**5. Z-Index Property**

z-index decides which element appears **on top or behind** another (like layers in Photoshop).

**🔹 Syntax**

z-index: number;

Higher value = element appears in front.

Works only when position is **relative, absolute, fixed, or sticky**.

**Overlapping Boxes**

**Index.html**

<div class="box box1">Box 1</div>

<div class="box box2">Box 2</div>

<div class="box box3">Box 3</div>

**Style.css**

.box {

width: 150px;

height: 150px;

position: absolute;

text-align: center;

line-height: 150px;

font-weight: bold;

color: white;

}

.box1 { background: purple; top: 40px; left: 40px; z-index: 1; }

.box2 { background: violet; top: 70px; left: 70px; z-index: 2; }

.box3 { background: black; top: 100px; left: 100px; z-index: 3; }

“Box 3” appears at the top, because it has the highest z-index.

**Real Website Use Cases**

* Popup windows over background
* Navigation bars above images
* Sticky chat icons above content
* Modal dialogs or alert boxes