**UX Instructions Exercise 2c**

1. Add **data-role attribute** to nav links for selector demo.
   * Example: <a href="#" data-role="nav-link">Home</a>
   * This allows selecting them via attribute selectors like [data-role="nav-link"].
2. Add **empty .box** in second flex row.
   * Used to demonstrate :empty pseudo-class and its styling:  
       
     <div class="box"></div>

**Let’s Add CSS Enhancements**

**1. Custom Font Loading**

@font-face {

font-family: 'OpenSans';

src: url(...);

font-display: swap;

}

* Adds OpenSans for performance typography and font-display: swap for faster rendering.

**2. Responsive Typography**

font-size: clamp(1rem, 1vw + 0.5rem, 1.2rem);

* Ensures readable font scaling across devices.

**3. Advanced Selectors and Pseudo-Classes**

* **Attribute Selector**

a[data-role="nav-link"] {

text-transform: uppercase;

}

* Selects only links with data-role="nav-link" and styles them.
* **:first-child and :last-child**

.nav-list li:is(:first-child, :last-child) a {

font-weight: bold;

}

* Highlights first and last nav items for emphasis.
* **:not(:first-child)**

.box:not(:first-child) {

border: 2px dashed #fff;

}

* Adds dashed border to all boxes except the first one.
* **:empty**

.box:empty::before {

content: "This box is empty";

}

* Shows fallback text when a .box is empty.

**4. Visual Effects and UX**

* :hover effect to scale .box:

.box:hover {

transform: scale(1.05);

}

* **Feature query using @supports:**

@supports (display: grid) {

.flex-demo {

border: 3px dashed green;

}

}

* Demonstrates conditional styling if the browser supports CSS Grid.

Optional: Let's add an <img> inside one of the .box elements and use **:has(img)** to give it a special background.

**HTML**

Add an image to one of the boxes:

<div class="box">

<img src="https://via.placeholder.com/100" alt="Placeholder image">

Box with Image

</div>

**CSS**

.box:has(img) {

background-color: #e0f7fa;

border: 2px solid #0076ce;

}

**Important:**

* :has() is supported in most modern desktop browsers (e.g., Chrome, Edge, Safari) but **not in Firefox as of mid-2025**.
* It’s extremely useful for interactive UIs without requiring JavaScript.

**Real-Life Use Cases of :nth-child:**

**Example 1: Zebra-striping table rows**

tr:nth-child(even) {

background-color: #f0f0f0;

}

* Alternates row color in a data table for readability.

**Example 2: Highlight every 3rd item in a list**

li:nth-child(3n) {

color: red;

}

* Useful for marking important steps or promotions (e.g., "every third product is on sale").

**Example 3: Adjust spacing in a grid**

.grid-item:nth-child(4n) {

margin-right: 0;

}

* Removes margin from the last item in a row of 4, avoiding layout gaps.

**Example 4: Target odd cards for background style**

.card:nth-child(odd) {

background: #e6f7ff;

}

* Visually separates alternating cards in a blog or product layout.

**Example 5: First 3 bullets in a tutorial get icons**

li:nth-child(-n+3)::before {

content: "🔥 ";

}

* Adds emoji icons to the first three tips.