**UX Instructions Exercise 2b**

**Build the HTML Structure (index.html)**

1. Add a <header> with an <h1> title
2. Add a <main> element with two sections

Intro text

Flexbox layout section with two flex rows: one normal and one reverse

1. Add a <footer> with copyright

**Make sure this is in styles (styles.css)**

1. Use :root to define CSS variables for consistent color, padding, and max-width.
2. Apply box-sizing: border-box globally.
3. Style body with padding, font-family, and background.
4. Limit content width using .container and max-width.
5. Implement the **skip link** so it’s only visible on focus.
6. Style .main-nav:

Horizontal layout using flex

Show/hide nav toggle based on screen width

1. Use @media to show .nav-toggle below 768px and hide/show .nav-list.
2. Add .flex-row with display: flex, gap, wrap, and justify-content.
3. Create .reverse class to flip the order using flex-direction: row-reverse.
4. Style .box with padding, color, and flexible sizing.

**Test and Explore**

* Resize the browser window to see the nav and layout adapt.
* Press Tab from the top of the page to see the skip link appear.
* Open DevTools and toggle mobile view to test wrapping behavior.
* Try removing the .reverse class and observe how content order changes.
* Change flex-wrap: wrap to nowrap and see how boxes behave on small screens.

**Key Concepts**

|  |  |
| --- | --- |
| **Feature** | **Purpose** |
| flex | Creates a responsive row or column layout |
| flex-wrap: wrap | Allows boxes to stack if space runs out |
| flex-direction: row-reverse | Changes the visual order of boxes |

**Step 1: Navigation Bar**

Create a sticky and responsive navigation bar:

.main-nav {

background: #eee;

padding: var(--padding);

display: flex;

justify-content: space-between;

align-items: center;

}

.nav-toggle {

display: none;

background: none;

border: none;

font-size: 1.2rem;

}

.nav-list {

list-style: none;

display: flex;

gap: 1rem;

padding: 0;

}

.nav-list li a {

text-decoration: none;

color: var(--primary-color);

}

**Step 2: Header and Footer**

Style site-wide branding areas:

.site-header {

text-align: center;

padding: 2rem 0;

background: var(--primary-color);

color: white;

}

.site-footer {

margin-top: 2rem;

text-align: center;

padding: 1rem;

background: #0076ce;

color: white;

}

**Step 3: Flexbox Demo Section**

Build two Flexbox rows: a standard row and a reversed row:

.flex-demo {

margin-top: 2rem;

background: #f0f0f0;

padding: var(--padding);

}

.flex-row {

display: flex;

gap: 1rem;

flex-wrap: wrap;

justify-content: space-between;

}

.flex-row.reverse {

flex-direction: row-reverse;

}

.box {

flex: 1 1 200px;

background: var(--primary-color);

color: white;

text-align: center;

padding: 2rem;

border-radius: 8px;

}

**Step 4: Mobile Navigation Support (Media Queries)**

Ensure the nav behaves well on smaller screens:

@media (max-width: 768px) {

.nav-toggle {

display: block;

}

.nav-list {

display: none;

flex-direction: column;

}

body.menu-open .nav-list {

display: flex;

}

}

**Extra: Add ARIA and Semantic Enhancements**

Improve accessibility by adding ARIA and role attributes:

* <nav>: add role="navigation" and aria-label="Main Navigation"
* Add role="banner" to the <header>
* Add role="contentinfo" to the <footer>