**Assignment 2: Styling My First Page (CSS Basics & Selectors)**

* **Objective:** Apply basic CSS styling using different selectors.
* **Tasks:**
  1. Create an external CSS file named about\_me\_styles.css and link it to your about\_me.html.
  2. **Element Selectors:**
     + Change the font-family of the entire body to a common sans-serif font (e.g., Arial, Helvetica).
     + Set a background-color for the body.
  3. **Class Selectors:**
     + Add a class (e.g., introduction) to your introduction paragraph and give it a different font-size and color.
     + Add a class (e.g., hobby-list) to your <ul> and change its list-style-type (e.g., square, circle).
  4. **ID Selector (Optional, for practice):**
     + If you have a unique element, give it an ID and style it (e.g., make the <h1> a different color using an ID).
  5. Style the link (<a>) to remove the underline by default (text-decoration: none;) and add it back on hover. Change its color.
* **Bonus:** Add some basic styling to the image (e.g., a simple border).

**Assignment 3: The Boxy Look (CSS Box Model)**

* **Objective:** Understand and apply padding, border, and margin.
* **Tasks:**
  1. Create a new HTML file product\_card.html and a linked product\_card\_styles.css.
  2. In HTML, create a div with a class product-card.
  3. Inside the product-card, add:
     + An image (<img>).
     + A heading (<h2>) for the product name.
     + A paragraph (<p>) for the product description.
     + A div or <p> with class price for the price.
     + A button (<button>) "Add to Cart".
  4. In CSS, for the .product-card:
     + Set a width (e.g., 300px).
     + Add a border (e.g., 1px solid #ccc).
     + Add padding (e.g., 15px) to create space between the content and the border.
     + Add margin (e.g., 20px) to create space around the card.
     + Set box-sizing: border-box; and observe its effect.
  5. Style the inner elements (image, heading, paragraph, price, button) with some basic margins and paddings to arrange them neatly within the card.
* **Bonus:** Create two or three product cards and arrange them side-by-side (you might need display: inline-block; for now, or this can lead into Flexbox).

**Assignment 4: Interactive Elements (Advanced CSS Styling)**

* **Objective:** Practice hover effects, transitions, shadows, and advanced text styling.
* **Tasks:**
  1. Continue with your product\_card.html or create a new page with a few buttons and text sections.
  2. **Buttons:**
     + For the "Add to Cart" button (and any other buttons):
       - Add a :hover state that changes its background-color and maybe color.
       - Add an :active state that makes it look pressed (e.g., slightly darker, transform: translateY(1px);).
       - Apply a transition to make these changes smooth.
       - Add a subtle box-shadow to the button and make it more pronounced on hover.
  3. **Text Styling:**
     + Style the product name (<h2>) with a specific font-weight, letter-spacing, and maybe text-transform: uppercase;.
     + If you have a sale price, use a <span> to style it differently (e.g., color red, text-decoration: line-through; for an old price).
     + Add a text-shadow to a prominent heading.
* **Bonus:** Create a "tooltip" (a simple div with text) that appears when you hover over an icon or a small piece of text (you'll need basic positioning for this, which is covered next).

**Assignment 5: Flexible Rows (CSS Flexbox)**

* **Objective:** Learn to create 1-dimensional layouts using Flexbox.
* **Tasks:**
  1. Create an HTML file navbar.html and navbar\_styles.css.
  2. Create a <nav> element that will be your flex container.
  3. Inside the <nav>, add:
     + A div for a logo (can be text or an <img>).
     + A ul for navigation links (<li><a>...</a></li>).
     + A div for a search bar (<input type="text">) or action buttons.
  4. Using Flexbox (display: flex; on the <nav>):
     + Arrange the logo, navigation links (ul), and search/actions div in a single row (flex-direction: row;).
     + Use justify-content to distribute space (e.g., space-between to push logo left and actions right, with nav in between or also pushed).
     + Use align-items: center; to vertically align all items in the navbar.
     + Make the ul for navigation links also a flex container to arrange the <li> items in a row.
* **Bonus:** Make the navigation links (ul) take up remaining space using flex-grow: 1;.

**Assignment 6: Page Structure with Grid (CSS Grid)**

* **Objective:** Learn to create 2-dimensional page layouts using CSS Grid.
* **Tasks:**
  1. Create holy\_grail\_layout.html and holy\_grail\_styles.css.
  2. The "Holy Grail" layout typically has a header, a footer, a main content area, and two sidebars (or one).
  3. Create a main div container for your grid.
  4. Inside, create divs for: header, left-sidebar, main-content, right-sidebar (optional), footer.
  5. Using CSS Grid (display: grid; on the main container):
     + Define grid-template-columns (e.g., 200px 1fr 200px for left-sidebar, main, right-sidebar or 200px 1fr if one sidebar).
     + Define grid-template-rows (e.g., auto 1fr auto for header, main area (stretching), footer).
     + Use grid-template-areas to visually define the layout and assign grid items to these areas using grid-area.
     + Add gap for spacing between grid areas.
* **Bonus:** Make one of the grid items (e.g., main-content) span multiple rows if you have a more complex row setup.

**Assignment 7: Overlays and Fixed Elements (CSS Positioning)**

* **Objective:** Practice using position: relative, absolute, fixed, and z-index.
* **Tasks:**
  1. Create image\_gallery.html and image\_gallery\_styles.css.
  2. Create a section that will act as a fixed header for this page.
  3. Create a grid (or flexbox) of several "image cards" (a div containing an <img>).
  4. For each image card (div):
     + Set its position: relative;.
     + Inside it, add a div for a caption or an icon.
     + Style this inner div with position: absolute; and use top, right, bottom, or left to place it as an overlay on the image (e.g., bottom-left corner).
     + Give it a semi-transparent background-color so the image behind is partially visible.
  5. Add a "modal" or "popup" div that is initially hidden (display: none;). Style this modal with position: fixed;, centered on the screen (using top, left, transform: translate(-50%, -50%);), and give it a high z-index so it appears above everything. (You won't make it appear/disappear with CSS alone yet, just style its positioned state).
* **Bonus:** Add a "SALE" badge to one of the image cards, absolutely positioned to its top-right corner, and use z-index to ensure it's above any caption overlay.