



Lab 11: CSS Layout and Styling Extravaganza - "The Retro Gaming Blog"

Instructions:

1. Create a folder named "Lab11" and create the appropriate folder structure.
2. While working on the lab, make sure to:
 - Add author comments to your code, the minimum is to have comments recording your name, lab number, and date.
 - Indent your code so that it's easy to read and debug.
 - Test frequently to eliminate bugs.

Part 1: Website Planning and Setup (30 minutes)

1. **Retro Gaming Theme:** Brainstorm ideas for your retro gaming blog. Consider:
 - **Name:** Come up with a catchy name (e.g., "Pixel Paradise," "Retro Rewind," "8-Bit Adventures").
 - **Focus:** What specific era or consoles will you focus on? (e.g., 8-bit era, classic arcade games, specific consoles like NES or Sega Genesis).
 - **Content:** What kind of content will you feature? (e.g., game reviews, nostalgic articles, interviews with retro gamers, high scores).
2. **Plan Your Pages:** Decide on the main sections or pages for your website. Include at least these:
 - **Home (index.html):** A landing page with featured articles, latest news, and a welcome message.
 - **Reviews (reviews.html):** A page with reviews of classic games.
3. **Create HTML Files:**
 - Create a new HTML file for each page.
 - Add the basic HTML structure (`<!DOCTYPE html>`, `<html>`, `<head>`, `<body>`) to each file.
4. **Add Content:**
 - Populate each HTML file with relevant content using headings (`<h1>`-`<h6>`), paragraphs (`<p>`), images (``), and lists (``, ``).
 - Use semantic HTML5 elements to structure the content within each page:
 - `<header>` for the header section (include the blog's name/logo and navigation)
 - `<nav>` for navigation menus (with links to the different pages)
 - `<main>` for the main content area of each page
 - `<article>` for individual articles or content sections
 - `<aside>` for related content (e.g., a sidebar with top-rated games or a "Retro Gamer of the Week" feature)
 - `<footer>` for the footer section (include copyright information and social media links)



Web Development Phase 2

Part 2: CSS Styling (45 minutes)

1. Create External Stylesheets:

- Create three CSS files:
 - `reset.css`: (Download a reset stylesheet or use the one provided)
 - `layout.css`: For layout-specific styles (floats, flexbox).
 - `style.css`: For general styling (colors, fonts, etc.).

2. Link Stylesheets:

- In the `<head>` of each HTML file, link the stylesheets in this order:

```
<link rel="stylesheet" href="reset.css">
<link rel="stylesheet" href="layout.css">
<link rel="stylesheet" href="style.css">
```

3. Apply CSS Styles:

- **`reset.css`**: This will reset default browser styles to create a consistent foundation.
- **`layout.css`**:
 - Use flexbox to create the basic layout of your pages (e.g., a two-column layout with a sidebar).
 - Arrange elements within specific sections (e.g., a horizontal navigation menu, a grid of game reviews).
 - You may use float if needed.
- **`style.css`**:
 - Style the overall appearance of the website (background colors, font families, text styles).
 - Style individual elements (headings, paragraphs, lists, images, links) to create a visually appealing design.
 - Use selectors (element, class, ID) to target specific elements.
 - Apply the box model properties (margins, borders, padding) to create spacing and visual separation.

Part 3: Refinement and Testing (15 minutes)

1. Review and Refine:

- Open each HTML file in a browser and review the design and layout.
- Test the navigation links to ensure they work correctly.
- Make adjustments to the CSS and HTML as needed to improve the visual appeal and usability of the website.

2. Cross-Browser Check:

- Test your website in different browsers (Chrome, Firefox, Edge) to ensure consistent rendering.



Web Development Phase 2

Submission

1. When you finish, show your work to your facilitator.
2. Zip your files and submit the zipped file to Google Classroom.
3. You may upload your work on Github to build your portfolio.