Lab 12: CSS Layout and Styling - "Indigenous Artists Showcase"

Instructions:

1. Create a folder named “Lab12” 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. **Choose a Focus:** Select a specific area of Indigenous art to showcase on your website (e.g., painting, sculpture, beadwork, carving, weaving, digital art).
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 a welcome message, featured artists, and an introduction to the art form.
   - **Artists (artists.html):** A page showcasing individual artists with their bios and artwork.
   - **Gallery (gallery.html):** A page with a gallery of different artworks.
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 (<img>), and lists (<ul>, <ol>).
   - Use semantic HTML5 elements to structure the content within each page:
     - <header> for the header section (include the website title and navigation)
     - <nav> for navigation menus (with links to the different pages)
     - <main> for the main content area of each page
     - <article> for individual artist profiles or artwork descriptions
     - <aside> for related content (e.g., a sidebar with links to Indigenous art resources or organizations)
     - <footer> for the footer section (include copyright information and acknowledgements)

Part 2: CSS Styling (45 minutes)

1. **Create External Stylesheets:**

- Create three CSS files:
  - *reset.css*
  - *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).
     - Use Flexbox to arrange elements within specific sections (e.g., a horizontal navigation menu, a grid of artist profiles, an image gallery).
     - Use Floats for smaller elements 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.
     - **Add this rule to the top of your style.css to set the font to Arial:**

```
body {
  font-family: Arial, sans-serif;
}
```

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.

Part 4: Advanced Styling and Enhancements (30 minutes)

1. **Advanced Styling:**
    - Experiment with more advanced CSS properties (e.g., gradients, shadows, transitions, transforms) to enhance the visual design.
    - Consider adding hover effects, animations, or other interactive elements.

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.