

CSS3 Exercises

Theme: Styling the "Local Community Event Portal"

1. Why CSS? Inline vs. Internal vs. External

Scenario: The designer wants you to experiment with different ways to apply styles.

Objective: Understand various CSS inclusion methods and their impact.

Task:

- Apply an inline style to make one heading red.
- Use an embedded `<style>` tag in the `<head>` to define body background.
- Link an external stylesheet `styles.css` and move all reusable styles there.
- Add comments in your CSS to label each section (`/* Header styles */`)

2. CSS Syntax and Comments

Scenario: You've joined a team and need to understand and maintain a large stylesheet.

Objective: Write clean, readable CSS with proper structure and comments.

Task:

- Create a section in `styles.css` with formatted rules and consistent indentation.
- Add descriptive comments above selectors.
- Example:

```
/* Style for main CTA button */  
.cta-button {  
  background-color: #007BFF;  
  color: white;  
}
```

3. Selectors Playground

Scenario: You need to style various elements based on IDs, classes, and element types.

Objective: Master different selector types.

Task:

- Use:
 - Universal selector `*` to reset margin/padding
 - Element selector to style all `<h2>`
 - ID selector `#mainHeader` for the banner

- Class selector *.eventCard* for event containers
- Grouping selector for *h3, p* to style together

4. Color & Background Styling

Scenario: You're theming the portal based on a city council's branding.

Objective: Apply consistent colors and background visuals.

Task:

- Use HEX and RGBA for setting text and background colors
- Add a background image to the body with fallback color
- Apply gradients to section headers using *background: linear-gradient(...)*

5. Typography: Fonts and Text

Scenario: The marketing team wants more appealing fonts and better readability.

Objective: Enhance textual appearance using CSS properties.

Task:

- Use *@import* or *<link>* to include a Google Font
- Set *font-family*, *font-size*, *font-style*, *font-weight* in different sections
- Use *text-align*, *text-transform*, *letter-spacing*, *line-height* on descriptions

6. Link and List Styling

Scenario: The default blue links and bullet lists don't match the design.

Objective: Customize links and lists.

Task:

- Style links with *:link*, *:hover*, *:active*, and *:visited* pseudo-classes
- Use *list-style-type*, *list-style-position*, and remove bullets from nav menus
- Add padding and margin to list items for spacing

7. Table Styling

Scenario: The events admin table needs a cleaner look.

Objective: Format tables using CSS.

Task:

- Style *table*, *th*, and *td* with borders, padding, and background color

- Add zebra striping to rows using *nth-child(even)*
- Use *border-collapse: collapse* and *text-align: center*

8. Box Model & Layout Control

Scenario: Sections are cramped and need spacing.

Objective: Control element spacing with margin, padding, border, and outline.

Task:

- Use developer tools to inspect and tweak box model properties
- Add border, padding, and margin to *.eventCard*
- Add *outline* to highlight selected fields in a form
- Compare *visibility: hidden* vs. *display: none*

9. Multiple Columns in Text

Scenario: The community bulletin needs to be displayed like a newspaper.

Objective: Use CSS3 multi-column layout.

Task:

- Create a news article section and apply:

```
column-count: 2;
column-gap: 30px;
column-rule: 1px solid gray;
```

10. Responsive Web Design with Media Queries

Scenario: Users will access the portal on phones, tablets, and desktops.

Objective: Apply media queries for responsiveness.

Task:

- Add a media query for screens smaller than 768px
- Stack navigation links vertically instead of horizontally
- Reduce image sizes and font sizes
- Use *%*, *vw*, *vh* for flexible layouts
- Bonus: Try Flexbox or Grid for responsive layouts

11. Debug and Test with Dev Tools and VS Code

Scenario: Some users report layout breaking on smaller screens.

Objective: Use DevTools to test styles and debug.

Task:

- Use Chrome's device toolbar to simulate different screen sizes
- Inspect applied styles and test different values live
- Use the Network tab to check if the external CSS is loaded