# CSS3 Exercises - 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 \*/)

Example HTML and CSS:

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8" />  
 <title>Local Community Event Portal</title>  
 <!-- Internal CSS -->  
 <style>  
 /\* Body background style \*/  
 body {  
 background-color: #f9f9f9;  
 font-family: Arial, sans-serif;  
 }  
 </style>  
 <!-- External CSS -->  
 <link rel="stylesheet" href="styles.css" />  
</head>  
<body>  
 <!-- Inline style: makes heading red -->  
 <h1 style="color: red;">Welcome to the Community Event Portal</h1>  
  
 <h2 id="mainHeader">Upcoming Events</h2>  
  
 <div class="eventCard">  
 <h3>Spring Festival</h3>  
 <p>Join us for food and fun!</p>  
 </div>  
</body>  
</html>

Content of external stylesheet (styles.css):

/\* styles.css \*/  
  
/\* Header styles \*/  
#mainHeader {  
 text-align: center;  
 background-color: #004466;  
 color: white;  
 padding: 15px;  
}  
  
/\* Event card styles \*/  
.eventCard {  
 border: 1px solid #ccc;  
 padding: 10px;  
 margin: 10px auto;  
 max-width: 400px;  
 background-color: #e6f2ff;  
}

## 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:

o Universal selector \* to reset margin/padding

o Element selector to style all <h2>

o ID selector #mainHeader for the banner

o Class selector .eventCard for event containers

o Grouping selector for h3, p to style together

/\* Reset margin and padding for all elements \*/  
\* {  
 margin: 0;  
 padding: 0;  
 box-sizing: border-box;  
}  
  
/\* Style all <h2> elements \*/  
h2 {  
 font-size: 2rem;  
 color: #333;  
}  
  
/\* ID selector for banner \*/  
#mainHeader {  
 font-weight: bold;  
}  
  
/\* Class selector for event containers \*/  
.eventCard {  
 border-radius: 5px;  
 box-shadow: 2px 2px 5px rgba(0,0,0,0.1);  
}  
  
/\* Grouping selector for h3 and p \*/  
h3, p {  
 font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
 margin-bottom: 8px;  
}

## 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(...)

/\* Body background with fallback and image \*/  
body {  
 background-color: #cce0ff; /\* fallback color \*/  
 background-image: url('cityscape.jpg');  
 background-size: cover;  
 background-repeat: no-repeat;  
 background-position: center;  
 color: rgba(0, 51, 102, 0.9); /\* text color with opacity \*/  
}  
  
/\* Gradient background for section headers \*/  
h2 {  
 background: linear-gradient(90deg, #004466, #66ccff);  
 -webkit-background-clip: text;  
 -webkit-text-fill-color: transparent;  
}

## 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

<!-- Add in <head> -->  
<link href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&display=swap" rel="stylesheet" />  
  
/\* Typography for body \*/  
body {  
 font-family: 'Roboto', sans-serif;  
 font-size: 16px;  
 line-height: 1.6;  
 color: #222;  
}  
  
/\* Description styling \*/  
p {  
 text-align: justify;  
 text-transform: none;  
 letter-spacing: 0.5px;  
 font-style: normal;  
}  
  
/\* Heading font styles \*/  
h1 {  
 font-weight: 700;  
 font-size: 2.5rem;  
 text-align: center;  
}

## 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

/\* Link styles \*/  
a:link {  
 color: #007BFF;  
 text-decoration: none;  
}  
a:visited {  
 color: #0056b3;  
}  
a:hover {  
 color: #004080;  
 text-decoration: underline;  
}  
a:active {  
 color: #00264d;  
}  
  
/\* Navigation list styling \*/  
nav ul {  
 list-style-type: none;  
 padding-left: 0;  
}  
  
nav li {  
 padding: 8px 12px;  
 margin: 4px 0;  
}

## 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

/\* Table basic style \*/  
table {  
 width: 100%;  
 border-collapse: collapse;  
 text-align: center;  
 margin: 20px 0;  
}  
  
th, td {  
 border: 1px solid #ddd;  
 padding: 10px;  
}  
  
/\* Zebra striping \*/  
tr:nth-child(even) {  
 background-color: #f2f2f2;  
}  
  
/\* Header row \*/  
th {  
 background-color: #004466;  
 color: white;  
}

## 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

/\* Event card spacing and border \*/  
.eventCard {  
 border: 2px solid #007BFF;  
 padding: 15px;  
 margin: 20px auto;  
 max-width: 400px;  
}  
  
/\* Form field outline on focus \*/  
input:focus, select:focus, textarea:focus {  
 outline: 2px solid #007BFF;  
}  
  
/\* Visibility vs Display \*/  
.hidden-element {  
 visibility: hidden; /\* occupies space but invisible \*/  
}  
  
.none-element {  
 display: none; /\* removed from layout \*/  
}

## 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;

/\* News article multi-column layout \*/  
.news-article {  
 column-count: 2;  
 column-gap: 30px;  
 column-rule: 1px solid gray;  
 padding: 20px;  
 max-width: 800px;  
 margin: auto;  
}

## 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

/\* Responsive for screens smaller than 768px \*/  
@media (max-width: 768px) {  
 nav ul {  
 display: flex;  
 flex-direction: column;  
 }  
  
 nav li {  
 padding: 10px 0;  
 }  
  
 img {  
 max-width: 100%;  
 height: auto;  
 }  
  
 body {  
 font-size: 14px;  
 }  
}  
  
/\* Bonus flexbox example \*/  
nav ul {  
 display: flex;  
 justify-content: space-around;  
 flex-wrap: wrap;  
}

## 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