Media Queries -Exercise

In this exercise, you will practice using media queries to create a responsive web page. Media queries allow you to apply different styles to a webpage based on the characteristics of the device, such as its width, height, and orientation. You will create a simple layout that adjusts its design at different breakpoints.

Objective

Understand and apply media queries to create responsive designs.

Use flexbox and grid layouts in combination with media queries to achieve a fluid and adaptive design.

You will create a simple web page with a header, main content area, and footer. The layout will change based on the screen size.

- For screens wider than 1200px, the layout should be a three-column grid.
- For screens between 600px and 1200px, the layout should switch to a two-column grid.
- For screens narrower than 600px, the layout should use a single-column flexbox layout.

Starter code

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Responsive Design with Media Queries</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }

        .container {
        }

        header, footer {
            background-color: #333;
            color: white;
            text-align: center;
            padding: 10px 0;
        }
        main {
            background-color: #f4f4f4;
            padding: 10px;
        }

        .box {
            background-color: #ddd;
            padding: 20px;
            border: 1px solid #ccc;
            text-align: center;
        }

        @media (min-width: 600px) and (max-width: 1199px) {
            .container {
            }
        }

        @media (min-width: 1200px) {
            .container {
            }
        }

        @media (max-width: 599px) {
            .container {
            }
        }
    </style>
</head>
<body>
    <header>
        <h1>Responsive Design Example</h1>
    </header>
    <div class="container">
        <div class="box">Box 1</div>
        <div class="box">Box 2</div>
        <div class="box">Box 3</div>
    </div>
    <footer>
        <p>&copy; 2024 Responsive Design</p>
    </footer>
</body>
</html>
```

Tasks

Initial Setup: Examine the starter code. Ensure you understand the basic structure of the HTML and the initial CSS styling.

Media Queries: Implement the provided media queries to achieve the desired layout changes at the specified breakpoints.

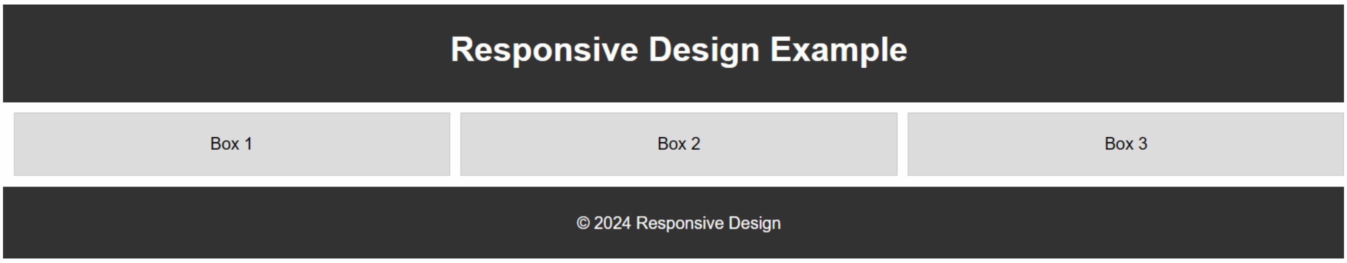
For screens wider than 1200px, apply a three-column grid layout.

For screens between 600px and 1200px, apply a two-column grid layout.

For screens narrower than 600px, switch to a single-column flexbox layout.

Test Responsiveness: Use your browser's developer tools to simulate different screen sizes and verify that the layout changes correctly at each breakpoint.

Enhance Styles: Optionally, enhance the styles for the header, footer, and boxes to improve the visual appeal of the page.



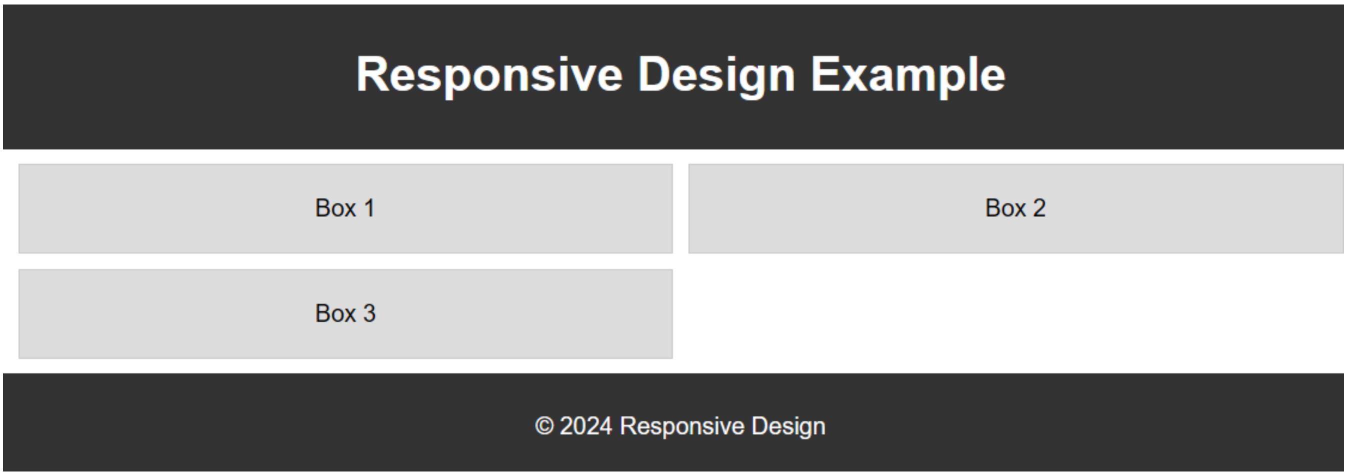
Responsive Design Example

Box 1

Box 2

Box 3

© 2024 Responsive Design



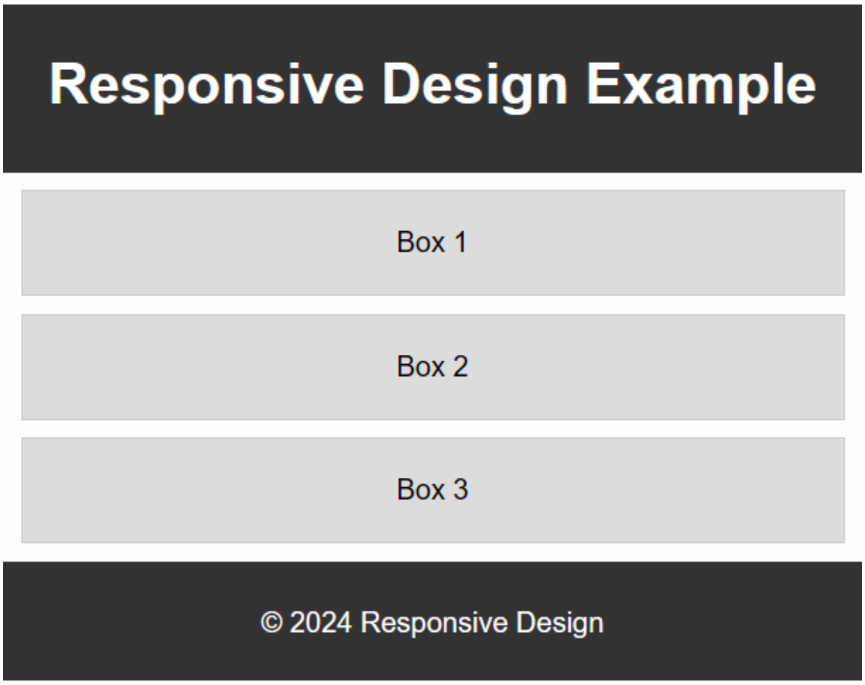
Responsive Design Example

Box 1

Box 2

Box 3

© 2024 Responsive Design



Responsive Design Example

Box 1

Box 2

Box 3

© 2024 Responsive Design

Advanced

In this advanced exercise, you will create a more complex responsive web page using media queries in combination with flexbox and grid layouts. The page will include a header, a navigation bar, a main content area with multiple sections, and a footer. The layout will adapt to various screen sizes.

Objective

- Apply media queries, flexbox, and grid layouts to create a complex responsive web page.
- Ensure that the page layout adapts seamlessly to different screen sizes.

Instructions

You will create a web page with the following sections:

1. **Header**: Contains the site title.
2. **Navigation Bar**: Contains links to different sections of the page.
3. **Main Content**: Divided into four sections.
4. **Footer**: Contains footer content.

The layout should adapt based on the screen size:

1. For screens wider than 1400px, the navigation bar should be horizontal, and the main content should be a four-column grid.
2. For screens between 900px and 1400px, the navigation bar should be vertical, and the main content should be a two-column grid.

3. For screens between 600px and 900px, the navigation bar should be hidden, and the main content should be a single-column flexbox layout.

4. For screens narrower than 600px, the navigation bar should be hidden, and the main content should be a single-column flexbox layout with adjusted padding.

Starter code

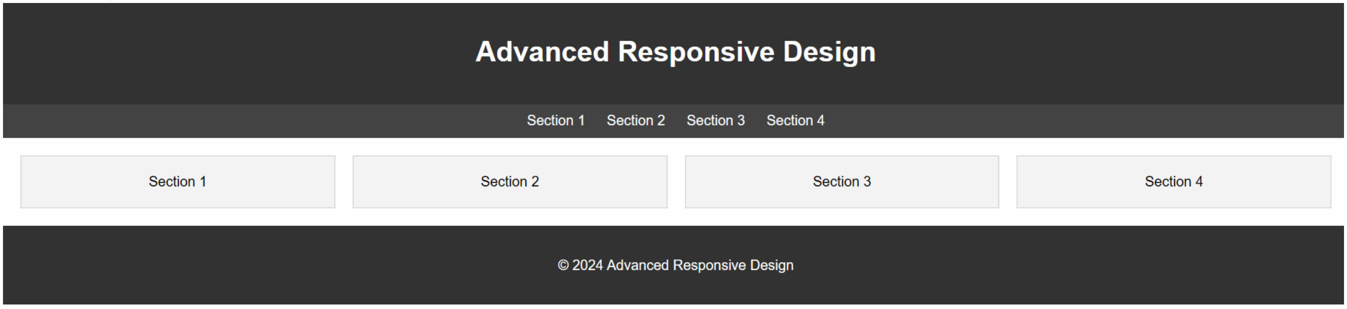
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Advanced Responsive Design</title>
    <style>
        body {
        }

        header, footer {
        }

        nav {
        }

        nav a {
        }

        .container {
        }

        .section {

        }

        @media (min-width: 600px) and (max-width: 899px) {
            .container {
            }

            nav {
            }
        }

        @media (min-width: 900px) and (max-width: 1399px) {
            nav {
            }

            .container {
            }
        }

        @media (min-width: 1400px) {
            nav {
      }

      .container {
      }
    }
  </style>
</head>
<body>
  <header>
    <h1>Advanced Responsive Design</h1>
  </header>
  <nav>
    <a href="#section1">Section 1</a>
    <a href="#section2">Section 2</a>
    <a href="#section3">Section 3</a>
    <a href="#section4">Section 4</a>
  </nav>
  <div class="container">
    <div id="section1" class="section">Section 1</div>
    <div id="section2" class="section">Section 2</div>
    <div id="section3" class="section">Section 3</div>
    <div id="section4" class="section">Section 4</div>
  </div>
  <footer>
    <p>&copy; 2024 Advanced Responsive Design</p>
  </footer>
</body>
</html>
```

Tasks

1. **Initial Setup**: Review the starter code to understand the structure and initial styles.
2. **Implement Media Queries**: Complete the provided media queries to achieve the desired layout adjustments at the specified breakpoints.
    - For screens wider than 1400px, ensure the navigation bar is horizontal, and the main content uses a four-column grid layout.
    - For screens between 900px and 1400px, the navigation bar should be vertical and fixed on the left, and the main content should use a two-column grid layout.
    - For screens between 600px and 900px, hide the navigation bar and switch to a single-column flexbox layout for the main content.
    - For screens narrower than 600px, ensure the navigation bar remains hidden and the main content uses a single-column flexbox layout with adjusted padding.

3. **Test Responsiveness**: Use the browser's developer tools to test the layout at different screen sizes and ensure the design adapts correctly.
4. **Enhance Styles**: Optionally, enhance the visual styles for the header, navigation bar, sections, and footer to improve the overall look and feel.

Desktop



Advanced Responsive Design

Section 1 Section 2 Section 3 Section 4

Section 1

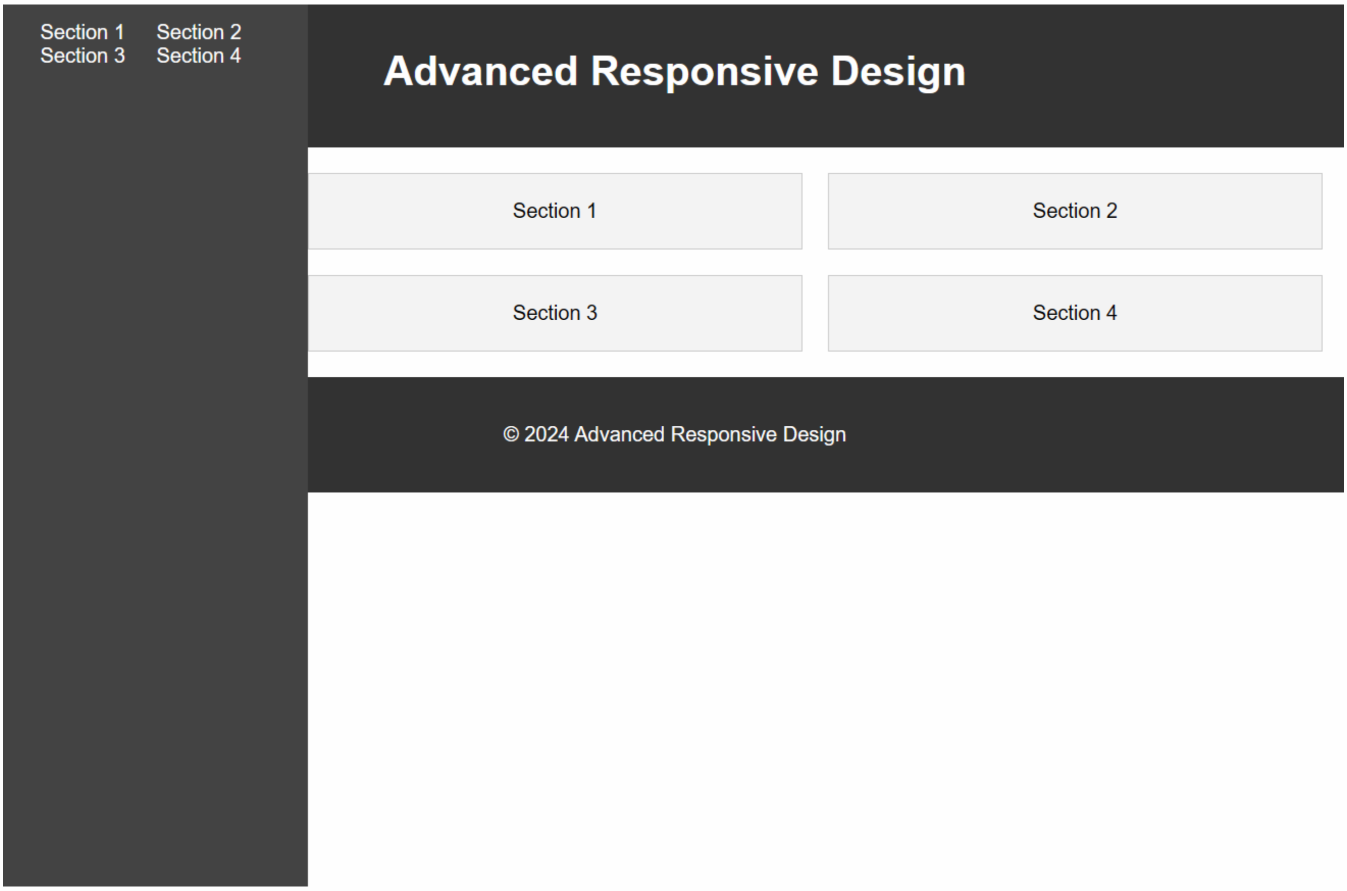
Section 2

Section 3

Section 4

© 2024 Advanced Responsive Design

Tablet



Section 1 Section 2
Section 3 Section 4

Advanced Responsive Design

Section 1

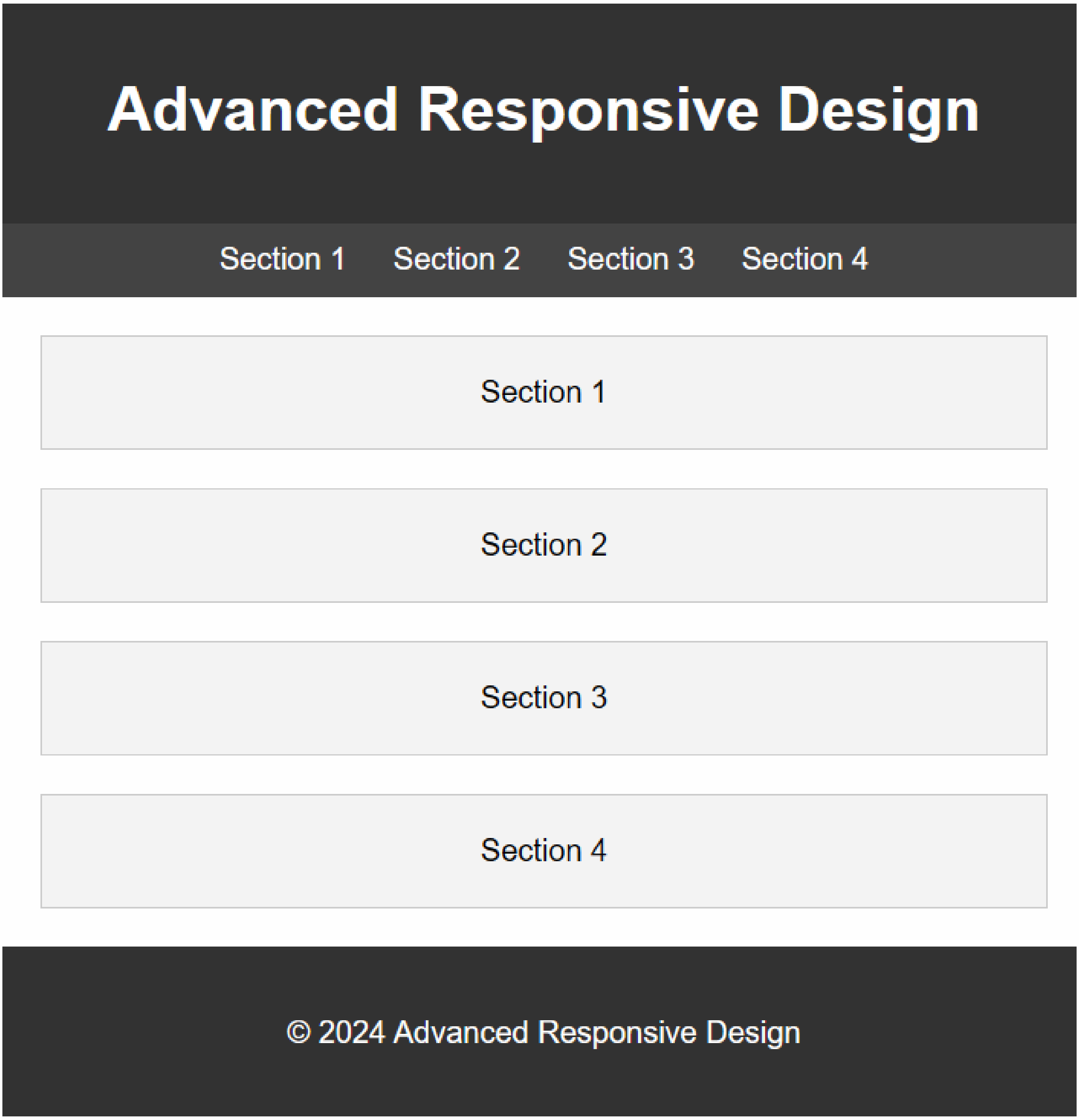
Section 2

Section 3

Section 4

© 2024 Advanced Responsive Design

Mobile:



Advanced Responsive Design

Section 1 Section 2 Section 3 Section 4

Section 1

Section 2

Section 3

Section 4

© 2024 Advanced Responsive Design