Layouts

# Program 1:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Flexbox Layout</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            background-color: #f2f2f2;

            display: flex;

            justify-content: center;

            align-items: center;

            height: 100vh;

            margin: 0;

        }

        .flex-container {

            display: flex;

            gap: 20px;

            padding: 20px;

            background-color: #fff;

            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

            border-radius: 10px;

        }

        .card {

            background-color: #f0f8ff;

            border-radius: 10px;

            width: 200px;

            padding: 20px;

            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

            text-align: center;

        }

        .card h3 {

            margin: 10px 0;

            color: #333;

        }

        .card p {

            color: #666;

        }

    </style>

</head>

<body>

    <div class="flex-container">

        <div class="card">

            <h3>Card 1</h3>

            <p>This is an example of a Flexbox layout.</p>

        </div>

        <div class="card">

            <h3>Card 2</h3>

            <p>Each card has equal width and spacing.</p>

        </div>

        <div class="card">

            <h3>Card 3</h3>

            <p>Flexbox makes it easy to align items.</p>

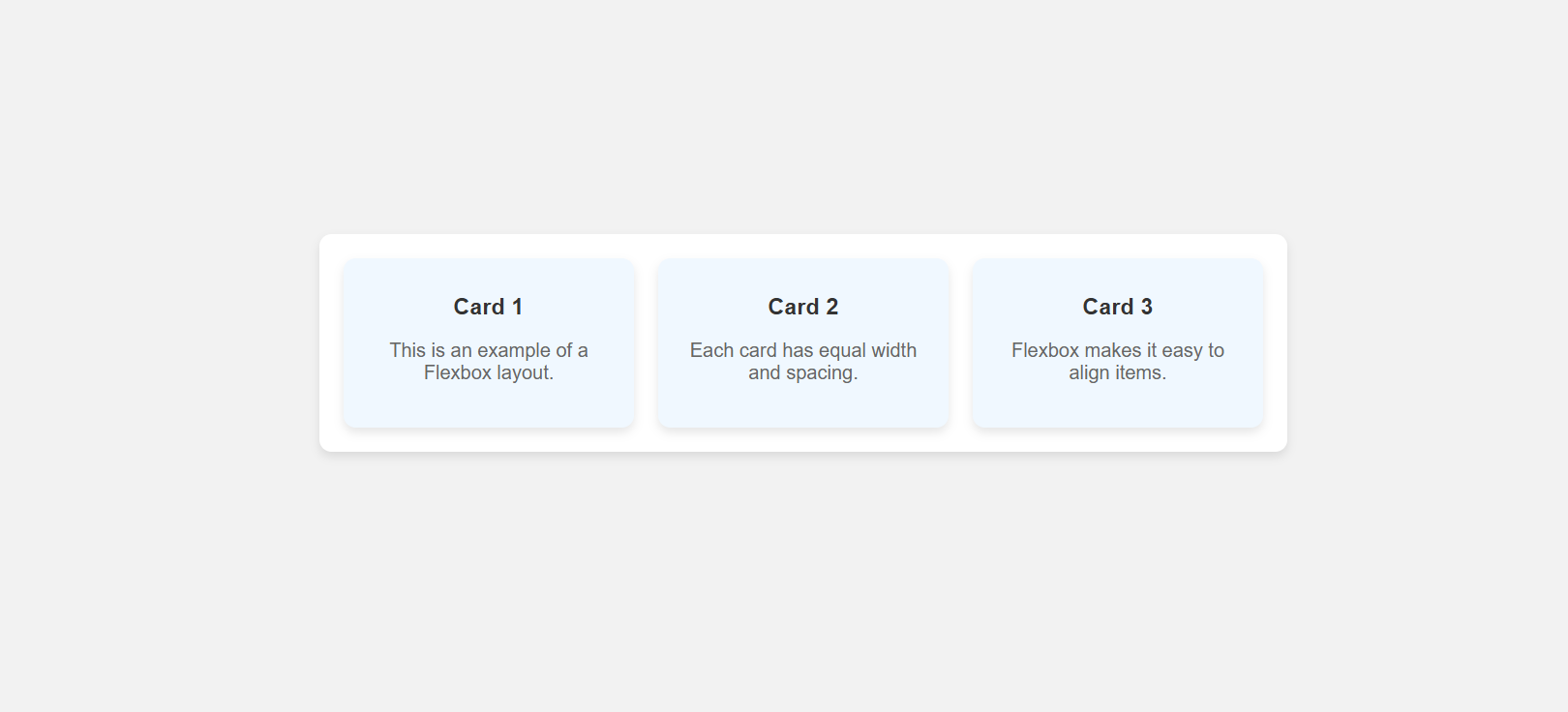
        </div>

    </div>

</body>

</html>

# Output :



# Program 2 :

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>CSS Grid Layout</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            background-color: #e6e6fa;

            display: flex;

            justify-content: center;

            align-items: center;

            height: 100vh;

            margin: 0;

        }

        .grid-container {

            display: grid;

            grid-template-columns: repeat(3, 1fr);

            gap: 15px;

            padding: 20px;

            background-color: #fff;

            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

            border-radius: 10px;

        }

        .grid-item {

            background-color: #fafad2;

            border-radius: 10px;

            padding: 30px;

            text-align: center;

            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

        }

        .grid-item h3 {

            color: #333;

        }

    </style>

</head>

<body>

    <div class="grid-container">

        <div class="grid-item"><h3>Item 1</h3></div>

        <div class="grid-item"><h3>Item 2</h3></div>

        <div class="grid-item"><h3>Item 3</h3></div>

        <div class="grid-item"><h3>Item 4</h3></div>

        <div class="grid-item"><h3>Item 5</h3></div>

        <div class="grid-item"><h3>Item 6</h3></div>

    </div>

</body>

</html>

# Output :



# Program 3 :

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Multi-Column Layout</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            line-height: 1.6;

            background-color: #f5f5f5;

            display: flex;

            justify-content: center;

            padding: 40px;

            margin: 0;

        }

        .content {

            background-color: #fff;

            padding: 20px;

            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

            border-radius: 10px;

            max-width: 800px;

        }

        .text-columns {

            column-count: 3;

            column-gap: 20px;

        }

        h2 {

            text-align: center;

            color: #333;

        }

    </style>

</head>

<body>

    <div class="content">

        <h2>Multi-Column Layout Example</h2>

        <div class="text-columns">

            <p>This layout splits text into multiple columns for better readability. Each column automatically adjusts based on screen size, providing a responsive layout.</p>

            <p>Responsive layouts ensure that content looks good on all devices, from mobile to desktop. The `column-count` CSS property controls the number of columns, and the `column-gap` property manages spacing between them.</p>

            <p>With this design, longer text content can be displayed in a more visually appealing way, mimicking the style of magazines or newspapers.</p>

            <p>Multi-column layouts are helpful for presenting articles, blogs, or any text-heavy content where readability is key. This layout gracefully adapts as you resize the browser window.</p>

        </div>

    </div>

</body>

</html>

# OutPut :

