The display property in CSS is used to define how elements are displayed on the webpage.

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

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

**<title>CSS Display Property Example</title>**

**<style>**

**/\* Block-level element example \*/**

**.block-element {**

**display: block;**

**width: 100%;**

**background-color: lightcoral;**

**padding: 10px;**

**margin-bottom: 10px;**

**text-align: center;**

**}**

**/\* Inline element example \*/**

**.inline-element {**

**display: inline;**

**background-color: lightseagreen;**

**padding: 10px;**

**margin-right: 10px;**

**color: white;**

**}**

**/\* Inline-block element example \*/**

**.inline-block-element {**

**display: inline-block;**

**width: 100px;**

**height: 100px;**

**margin-right: 10px;**

**background-color: lightblue;**

**text-align: center;**

**vertical-align: top;**

**}**

**/\* Flexbox container example \*/**

**.flex-container {**

**display: flex;**

**justify-content: space-around;**

**margin-bottom: 20px;**

**}**

**.flex-item {**

**background-color: lightgoldenrodyellow;**

**width: 100px;**

**height: 100px;**

**text-align: center;**

**line-height: 100px;**

**}**

**/\* Grid container example \*/**

**.grid-container {**

**display: grid;**

**grid-template-columns: repeat(3, 100px);**

**grid-gap: 10px;**

**}**

**.grid-item {**

**background-color: lightpink;**

**height: 100px;**

**text-align: center;**

**line-height: 100px;**

**}**

**</style>**

**</head>**

**<body>**

**<h1>CSS Display Property Examples</h1>**

**<!-- Block-level element -->**

**<div class="block-element">This is a block element</div>**

**<!-- Inline element -->**

**<span class="inline-element">Inline Element 1</span>**

**<span class="inline-element">Inline Element 2</span>**

**<span class="inline-element">Inline Element 3</span>**

**<!-- Inline-block element -->**

**<div class="inline-block-element">Item 1</div>**

**<div class="inline-block-element">Item 2</div>**

**<div class="inline-block-element">Item 3</div>**

**<!-- Flexbox example -->**

**<div class="flex-container">**

**<div class="flex-item">Flex 1</div>**

**<div class="flex-item">Flex 2</div>**

**<div class="flex-item">Flex 3</div>**

**</div>**

**<!-- Grid example -->**

**<div class="grid-container">**

**<div class="grid-item">Grid 1</div>**

**<div class="grid-item">Grid 2</div>**

**<div class="grid-item">Grid 3</div>**

**<div class="grid-item">Grid 4</div>**

**<div class="grid-item">Grid 5</div>**

**<div class="grid-item">Grid 6</div>**

**</div>**

**</body>**

**</html>**

**Explanation:**

* **Block element (display: block;):** The .block-element div takes up the entire width and starts on a new line.
* **Inline element (display: inline;):** The .inline-element spans are inline, meaning they don't start a new line and are displayed next to each other.
* **Inline-block element (display: inline-block;):** The .inline-block-element divs behave like inline elements but allow you to set width and height, aligning them next to each other.
* **Flexbox container (display: flex;):** The .flex-container uses Flexbox to arrange the child .flex-item divs in a row and distributes space between them.
* **Grid container (display: grid;):** The .grid-container uses CSS Grid to create a layout with three equal-width columns and 10px gaps between items. The .grid-item divs are placed into the grid cells.

o/p

