**Gemini**

**Flexbox: A Powerful Layout System**

**Flexbox** is a CSS layout module designed to provide a more efficient way to arrange items in a container. It's particularly useful for creating flexible and responsive layouts that adapt to different screen sizes.

**Key Concepts**

* **Flex Container:** The parent element that establishes the layout context.
* **Flex Items:** The child elements within the container.

**Properties**

* **display: flex;:** Sets the container as a flex container.
* **flex-direction:** Controls the direction of the items within the container.
  + row: Items are arranged horizontally.
  + column: Items are arranged vertically.
  + row-reverse: Items are arranged horizontally in reverse order.
  + column-reverse: Items are arranged vertically in reverse order.
* **justify-content:** Aligns the items along the main axis (horizontal by default).
  + flex-start: Items are aligned to the start of the container.
  + flex-end: Items are aligned to the end of the container.
  + center: Items are centered within the container.
  + space-between: Items are spaced evenly with the first item at the start and the last item at the end.
  + space-around: Items are spaced evenly with equal gaps on both sides of each item.
* **align-items:** Aligns the items along the cross axis (vertical by default).
  + flex-start: Items are aligned to the start of the container.
  + flex-end: Items are aligned to the end of the container.
  + center: Items are centered within the container.
  + stretch: Items are stretched to fill the full height of the container.
  + baseline: Items are aligned based on their baselines.

**Example: Responsive Grid Layout**

HTML

<!DOCTYPE html>

<html>

<head>

<title>Flexbox Layout</title>

<style>

.container {

display: flex;

flex-wrap: wrap;

}

.box {

width: 200px;

height: 200px;

border: 1px solid black;

margin: 10px;

text-align: center;

}

</style>

</head>

<body>

<div class="container">

<div class="box">Box 1</div>

<div class="box">Box 2</div>

<div class="box">Box 3</div>

<div class="box">Box

4</div>

<div class="box">Box 5</div>

</div>

</body>

</html>

Use code [with caution.](/faq#coding)

**Explanation:**

* The .container class sets the container to use Flexbox layout.
* The flex-wrap: wrap; property allows the items to wrap to the next line if there's not enough space.
* The .box class styles each individual item.

By adjusting the properties like flex-direction, justify-content, and align-items, you can create various responsive layouts that adapt to different screen sizes and orientations. Flexbox provides a powerful and flexible way to design modern web interfaces.