**Flexbox: A Powerful Layout System**

**Flexbox** is a CSS layout module that provides a flexible and efficient way to arrange items on a web page. It's particularly useful for creating responsive designs 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 flex container.
* **Flex Properties:** Properties that control the behavior of flex items within the container.

**Common Flexbox Properties:**

* **display: flex;:** Sets the container as a flex container.
* **justify-content:** Controls the alignment of flex items along the main axis.
  + flex-start: Aligns items to the start of the container.
  + flex-end: Aligns items to the end of the container.
  + center: Centers items within the container.
  + space-between: Spaces items evenly, with the first item at the start and the last item at the end.
  + space-around: Spaces items evenly, with half of the available space before the first item, after the last item, and between items.
* **align-items:** Controls the alignment of flex items along the cross axis.
  + flex-start: Aligns items to the start of the cross axis.
  + flex-end: Aligns items to the end of the cross axis.
  + center: Centers items along the cross axis.
  + stretch: Stretches items to fill the cross axis.
* **flex-direction:** Sets the direction of the main axis and the cross axis.
  + row: The main axis is horizontal, and the cross axis is vertical.
  + row-reverse: The main axis is horizontal, but the items are arranged in reverse order.
  + column: The main axis is vertical, and the cross axis is horizontal.
  + column-reverse: The main axis is vertical, but the items are arranged in reverse order.

**Example:**

HTML

<div class="container">

<div class="item">Item 1</div>

<div class="item">Item 2</div>

<div class="item">Item 3</div>

</div>

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

CSS

.container {

display: flex;

justify-content: space-between;

align-items: center;

}

.item

{

width: 100px;

height: 100px;

background-color: lightblue;

text-align: center;

line-height: 100px; /\* Vertically centers text \*/

}

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

This example creates a container with three items arranged horizontally with equal spacing between them. The items are centered both horizontally and vertically.

By experimenting with different flexbox properties, you can create a wide range of responsive layouts that adapt to various screen sizes and devices.