lexbox Layout Model

Flexbox is a CSS layout model that provides a flexible and efficient way to arrange items within a container. It's particularly useful for creating responsive layouts that adapt to different screen sizes.

Key Concepts:

Flex container: The container element that defines the flexbox layout.

Flex items: The child elements within the flex container.

Flex axis: The direction in which the flex items are arranged (horizontal or vertical).

Cross axis: The axis perpendicular to the flex axis.

Flexbox Properties:

display: flex;: Sets the container as a flex container.

flex-direction: Controls the direction of the flex items.

row: Items are arranged horizontally.

row-reverse: Items are arranged horizontally in reverse order.

column: Items are arranged vertically.

column-reverse: Items are arranged vertically in reverse order.

justify-content: Controls the alignment of items along the main axis.

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: Controls the alignment of items along the cross axis.

flex-start: Items are aligned to the start of the cross axis.

flex-end: Items are aligned to the end of the cross axis.

center: Items are centered along the cross axis.

stretch: Items are stretched to fill the cross axis.

Example:

HTML

<!DOCTYPE html>

<html>

<head>

<title>Flexbox Example</title>

<style>

.container {

display: flex;

justify-content: space-between;

align-items: center;

}

.item {

width: 200px;

height: 100px;

background-color: lightblue;

text-align: center;

line-height: 100px;

}

</style>

</head>

<body>

<div class="container">

<div class="item">Item

1</div>

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

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

</div>

</body>

</html>

Use code with caution.

In this example, the three items are arranged horizontally with equal spacing between them and centered vertically within the container. If you adjust the screen size, the items will automatically rearrange to fit the available space.

By using Flexbox and its various properties, you can create flexible and responsive layouts for your web pages.