**Day 6: Mastering Flexbox**

**Introduction**

Welcome to Day 6 of our web development journey! Today, we'll delve into the world of Flexbox, a powerful layout model that makes it easier to design complex, responsive, and flexible layouts.

**Understanding Flexbox**

Flexbox is a layout model in CSS that allows you to design complex layouts more efficiently. It involves two key components:

**Flex Container**: The container where you apply the display: flex; property. It defines a flex context for its direct children.

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

Flex Items: Elements inside a flex container become flex items. Flex items can be arranged horizontally or vertically based on the container's properties.

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

**Creating Flexible Layouts**

Flexbox provides properties to control the layout, alignment, and distribution of elements within a flex container. Let's explore some common properties:

**1. justify-content**

The justify-content property aligns flex items along the main axis. The space-between value evenly distributes items, leaving equal space between them.

**.flex-container {**

**display: flex;**

**justify-content: space-between;**

**}**

**2. align-items**

The align-items property aligns flex items along the cross axis. The center value vertically centers the items.

**.flex-container {**

**display: flex;**

**align-items: center;**

**}**

**Task for Students**

Now it's your turn! Create an HTML document with a flex container and multiple flex items. Experiment with different values for justify-content and align-items to see how they affect the layout. Share your observations and any challenges you faced.

**Reference Code**

**<!DOCTYPE html>**

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

**<head>**

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

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

**<title>Flexbox Example</title>**

**<style>**

**body {**

**font-family: 'Arial', sans-serif;**

**margin: 0;**

**padding: 0;**

**background-color: #f8f8f8;**

**color: #333;**

**}**

**/\* Flexbox Styles \*/**

**.flex-container {**

**display: flex;**

**justify-content: space-between;**

**align-items: center;**

**background-color: #333;**

**color: #fff;**

**padding: 10px;**

**}**

**.nav-bar {**

**display: flex;**

**list-style: none;**

**padding: 0;**

**margin: 0;**

**}**

**.nav-item {**

**margin-right: 20px;**

**}**

**.content-container {**

**display: flex;**

**justify-content: space-evenly;**

**padding: 20px;**

**}**

**.content-section {**

**flex: 1;**

**padding: 20px;**

**background-color: #fff;**

**box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);**

**border-radius: 8px;**

**margin: 10px;**

**}**

**</style>**

**</head>**

**<body>**

**<!-- Header Section -->**

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

**<h1>Flexbox Example</h1>**

**<nav class="nav-bar">**

**<li class="nav-item">Home</li>**

**<li class="nav-item">About</li>**

**<li class="nav-item">Contact</li>**

**</nav>**

**</header>**

**<!-- Content Section -->**

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

**<!-- Content 1 -->**

**<div class="content-section">**

**<h2>Section 1</h2>**

**<p>This is the first content section.</p>**

**</div>**

**<!-- Content 2 -->**

**<div class="content-section">**

**<h2>Section 2</h2>**

**<p>This is the second content section.</p>**

**</div>**

**<!-- Content 3 -->**

**<div class="content-section">**

**<h2>Section 3</h2>**

**<p>This is the third content section.</p>**

**</div>**

**</div>**

**<!-- Footer Section -->**

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

**<p>&copy; 2023 Flexbox Example. All rights reserved.</p>**

**</footer>**

**</body>**

**</html>**

**Day 6 document in code**

**<!DOCTYPE html>**

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

**<head>**

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

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

**<title>Day 6: Mastering Flexbox</title>**

**<style>**

**body {**

**font-family: 'Arial', sans-serif;**

**margin: 0;**

**padding: 0;**

**background-color: #f8f8f8;**

**color: #333;**

**}**

**.container {**

**max-width: 800px;**

**margin: 0 auto;**

**padding: 20px;**

**background-color: #fff;**

**box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);**

**border-radius: 8px;**

**margin-bottom: 20px;**

**}**

**h1, h2, h3 {**

**color: #4CAF50;**

**}**

**p {**

**line-height: 1.6;**

**}**

**.example-container {**

**margin-bottom: 40px;**

**}**

**.code-container {**

**background-color: #f0f0f0;**

**padding: 15px;**

**border-radius: 5px;**

**margin-top: 20px;**

**overflow-x: auto;**

**}**

**/\* Flexbox Styles \*/**

**.flex-container {**

**display: flex;**

**justify-content: space-between;**

**align-items: center;**

**}**

**.flex-item {**

**flex: 1;**

**padding: 20px;**

**background-color: #4CAF50;**

**color: #fff;**

**margin: 10px;**

**text-align: center;**

**}**

**</style>**

**</head>**

**<body>**

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

**<h1>Day 6: Mastering Flexbox</h1>**

**<p>Welcome to Day 6! Today, we'll delve into the world of Flexbox, a powerful layout model that makes it easier to design complex, responsive, and flexible layouts.</p>**

**</div>**

**<!-- Flexbox Explanation -->**

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

**<h2>Understanding Flexbox</h2>**

**<p>Flexbox is a layout model in CSS that allows you to design complex layouts more efficiently. It involves a flex container and flex items.</p>**

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

**<h3>Flex Container</h3>**

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

**<p>&lt;div class="flex-container"&gt;&lt;/div&gt;</p>**

**</div>**

**<p>The <code>display: flex;</code> property is applied to create a flex container. It defines a flex context for its direct children.</p>**

**</div>**

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

**<h3>Flex Items</h3>**

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

**<p>&lt;div class="flex-item"&gt;&lt;/div&gt;</p>**

**</div>**

**<p>Elements inside a flex container become flex items. Flex items can be arranged horizontally or vertically based on the container's properties.</p>**

**</div>**

**</div>**

**<!-- Creating Flexible Layouts -->**

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

**<h2>Creating Flexible Layouts</h2>**

**<p>Flexbox provides properties to control the layout, alignment, and distribution of elements within a flex container. Let's explore some common properties:</p>**

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

**<h3>justify-content</h3>**

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

**<p>.flex-container {</p>**

**<p> display: flex;</p>**

**<p> justify-content: space-between;</p>**

**<p>}</p>**

**</div>**

**<p>The <code>justify-content</code> property aligns flex items along the main axis. The <code>space-between</code> value evenly distributes items, leaving equal space between them.</p>**

**</div>**

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

**<h3>align-items</h3>**

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

**<p>.flex-container {</p>**

**<p> display: flex;</p>**

**<p> align-items: center;</p>**

**<p>}</p>**

**</div>**

**<p>The <code>align-items</code> property aligns flex items along the cross axis. The <code>center</code> value vertically centers the items.</p>**

**</div>**

**</div>**

**<!-- Task for Students -->**

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

**<h2>Task for Students</h2>**

**<p>Now it's your turn! Create an HTML document with a flex container and multiple flex items. Experiment with different values for <code>justify-content</code> and <code>align-items</code> to see how they affect the layout. Share your observations and any challenges you faced.</p>**

**</div>**

**<!-- Footer Section -->**

**<footer>**

**<p>&copy; 2023 Web Development Learning. All rights reserved.</p>**

**</footer>**

**</body>**

**</html>**