**Alignment and Display in CSS**

In CSS, aligning elements is crucial for creating visually appealing layouts. Different alignment techniques and display properties can be used to achieve the desired effect.

**Key Display Properties:**

1. **Block:**
   * Elements take up the full width available, with a new line before and after.
   * Example: <div>, <h1>, <p>.
2. **Inline:**
   * Elements only take up as much width as necessary, and they do not start on a new line.
   * Example: <span>, <a>.
3. **Inline-block:**
   * Similar to inline, but elements can have width and height set and respect margins and padding.
   * Example: <div style="display: inline-block;">.
4. **Flex:**
   * A layout model that allows responsive design by controlling the alignment, direction, and size of items in a container.
   * Example: A <div> with display: flex;.

**Alignment Techniques:**

* **Text Alignment:**
  + Use the text-align property to set the horizontal alignment of text within an element.
  + Values: left, right, center, justify.
* **Vertical Alignment:**
  + Use the vertical-align property with inline or table-cell elements.
  + Common values: top, middle, bottom.

**Example HTML Document:**

Here's an example HTML document showcasing various alignment techniques.

html

Copy code

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Alignment and Display Example</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 20px;

}

.block {

background-color: #f0f0f0;

padding: 20px;

margin-bottom: 20px;

}

.inline {

display: inline;

background-color: #ffcccb;

padding: 10px;

margin-right: 10px;

}

.inline-block {

display: inline-block;

width: 150px;

height: 100px;

background-color: #add8e6;

text-align: center;

vertical-align: middle;

margin: 10px;

line-height: 100px; /\* Center text vertically \*/

}

.flex-container {

display: flex;

justify-content: space-between; /\* Align items horizontally \*/

align-items: center; /\* Align items vertically \*/

background-color: #90ee90;

padding: 20px;

margin-top: 20px;

}

.flex-item {

background-color: #fff;

padding: 20px;

border: 1px solid #000;

width: 30%;

text-align: center; /\* Center text within flex items \*/

}

</style>

</head>

<body>

<h1>CSS Alignment and Display Example</h1>

<div class="block">

<h2>Text Alignment</h2>

<p style="text-align: left;">Left-aligned text.</p>

<p style="text-align: center;">Center-aligned text.</p>

<p style="text-align: right;">Right-aligned text.</p>

<p style="text-align: justify;">This text is justified, which means it is stretched to align with both the left and right margins.</p>

</div>

<div class="block">

<h2>Inline Elements</h2>

<span class="inline">Inline Element 1</span>

<span class="inline">Inline Element 2</span>

<span class="inline">Inline Element 3</span>

</div>

<div class="block">

<h2>Inline-Block Elements</h2>

<div class="inline-block">Inline-Block 1</div>

<div class="inline-block">Inline-Block 2</div>

<div class="inline-block">Inline-Block 3</div>

</div>

<div class="flex-container">

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

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

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

</div>

</body>

</html>

**Explanation of the Example:**

1. **Text Alignment:**
   * The text alignment is demonstrated through various paragraphs within a block, showing left, center, right, and justified alignments.
2. **Inline Elements:**
   * Three inline elements are displayed next to each other without breaking to a new line. They occupy only as much space as their content.
3. **Inline-Block Elements:**
   * Inline-block elements allow you to set specific dimensions. Here, the text within each box is vertically centered using line-height.
4. **Flex Container:**
   * A flex container demonstrates how items can be aligned both horizontally (justify-content: space-between) and vertically (align-items: center). Each flex item is styled with padding, borders, and centered text.