**Name : Shafi Ur Rehman**

**Reg No : CS120212042**

**Semester : 6th “A”(Open)**

**Assignment : Web development**

**Date : 05/05/2024**

**Submitted to: Sir Noor Mast**

**Topic: CSS FlexBox**

**Introduction**

* CSS flexbox is a layout model that allows you to design flexible and responsive layouts in a single direction—either horizontally or vertically. It enables you to distribute space among items in a container, align them, and reorder them easily, making it great for creating dynamic and adaptive designs.

**CSS Flexbox Layout Module**

* Before the Flexbox Layout module, there were four layout modes:
* Block, for sections in a webpage
* Inline, for text
* Table, for two-dimensional table data
* Positioned, for explicit position of an element

**Flexbox Elements**

* To start using the Flexbox model, you need to first define a flex container.



The element above represents a flex container (the blue area) with three flex items.

**Example**

<!DOCTYPE html>

<html>

<head>

<style>

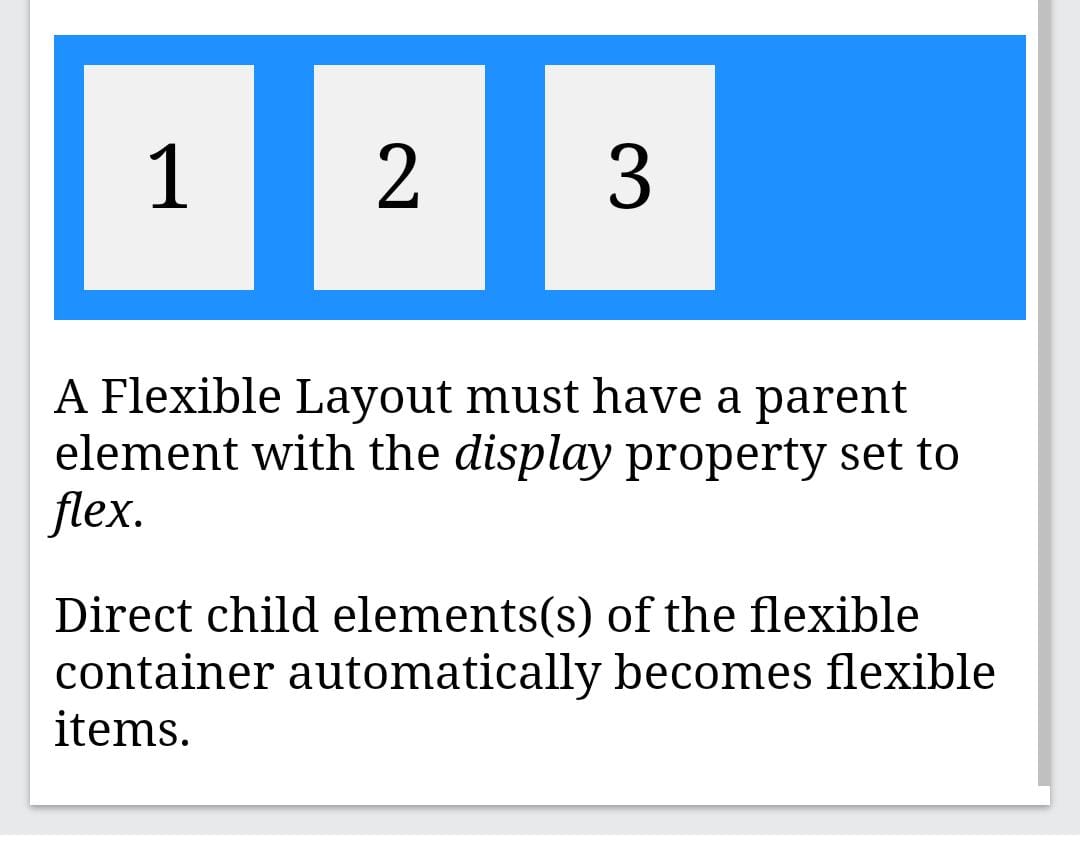
.flex-container {

display: flex;

OutPut

background-color: DodgerBlue; **Create a Flex Container**

}



.flex-container > div {

background-color: #f1f1f1;

margin: 10px;

padding: 20px;

font-size: 30px;

}

</style>

</head>

<body>

<h1>Create a Flex Container</h1>

<div class="flex-container">

<div>1</div>

<div>2</div>

<div>3</div>

</div>

<p>A Flexible Layout must have a parent element with the <em>display</em> property set to <em>flex</em>.</p>

<p>Direct child elements(s) of the flexible container automatically becomes flexible items.</p>

</body>

</html>

**FlexBox Properties**

Flexbox is a layout model in CSS that allows you to design more complex and flexible layouts with ease. Some key properties include:

1. display: Set to flex to enable flexbox layout on a container element.
2. flex-direction: Determines the direction in which flex items are placed in the flex container (row, row-reverse, column, column-reverse).
3. flex-wrap: Defines whether flex items are forced onto a single line or can wrap onto multiple lines.
4. justify-content: Aligns flex items along the main axis of the flex container.
5. align-items: Aligns flex items along the cross axis of the flex container.
6. align-content: Aligns a flex container's lines within the flex container when there is extra space in the cross-axis.
7. flex: A shorthand for flex-grow, flex-shrink, and flex-basis properties of a flex item.
8. order: Specifies the order of a flex item relative to the other flex items in the flex container.

These properties give you a lot of control over the layout and alignment of flex items within a flex container.

**Flexbox Advantage**

Flexbox offers several advantages for creating layouts:

1. \*Flexible Box Sizing\*: Flexbox allows items within a container to grow or shrink to fill available space, making it easier to create responsive designs.

2. \*Direction Control\*: You can control the direction of items within the container, whether it's in rows or columns, and even reverse the order if needed.

3. \*Alignment\*: Flexbox provides powerful alignment options, allowing you to easily align items both horizontally and vertically within the container, as well as distributing space between them.

4. \*Responsive Design\*: With Flexbox, you can create layouts that adapt to different screen sizes and devices, making it ideal for building responsive websites.

5. \*No Floats or Clearfix Hacks\*: Unlike traditional CSS layout techniques like floats, Flexbox eliminates the need for hacks like clearfix to handle alignment and spacing issues.

6. \*Single-Dimensional Layout Control\*: Flexbox simplifies layout control by focusing on either the row or column axis at a time, making it easier to manage complex layouts.

7. \*Ordering\*: You can easily reorder items within the container without altering the source order in the HTML, providing more flexibility in layout design.

Overall, Flexbox offers a more efficient and intuitive way to create complex layouts compared to traditional CSS methods.