Flutter Layout Cheat Sheet

codingwithflutter.com

Version 1.0

Introduction

Almost everything in Flutter is a widget.
So any layout that you build in your app is expressed as a combination of widgets.

The official Flutter docs already offer a complete list of layout widgets that you can use to arrange your content.

This document is intended as a quick reference about how to use the most common layouts.

Further Reading

- Building Layouts in Flutter
- Layout Widgets

Also check out my Flutter Layout Demo on Github.

Two types of widgets

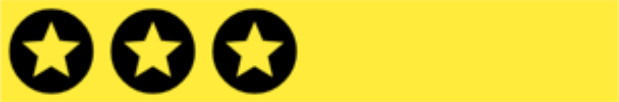
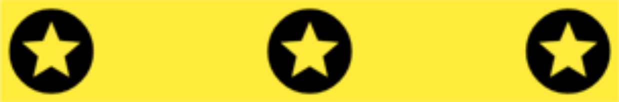
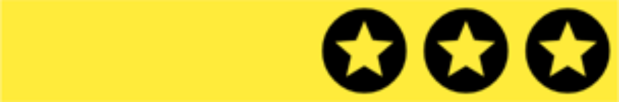
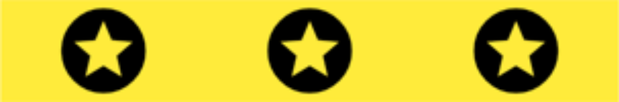
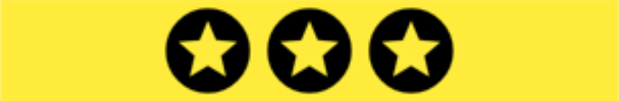
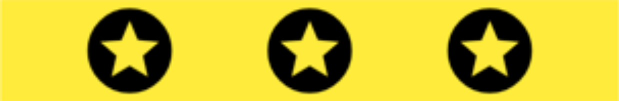
As far as layouts are concerned, there are two types of widgets:

- single-child widgets (e.g. Container, Expanded)
- multi-child widgets (e.g. Row, Column)

Here we explore a few of these and their properties.

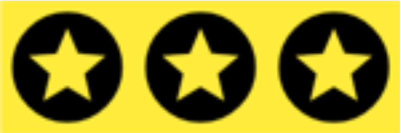
Row.mainAxisAlignment

```
Row(
  mainAxisAlignment: _mainAxisAlignment,
  mainAxisSize: MainAxisSize.max,
  children: [
    Icon(Icons.stars, size: 50.0),
    Icon(Icons.stars, size: 50.0),
    Icon(Icons.stars, size: 50.0),
  ],
)
```

MainAxisAlignment.start 	**MainAxisAlignment.spaceBetween** 
MainAxisAlignment.end 	**MainAxisAlignment.spaceAround** 
MainAxisAlignment.center 	**MainAxisAlignment.spaceEvenly** 

MainAxisSize.min

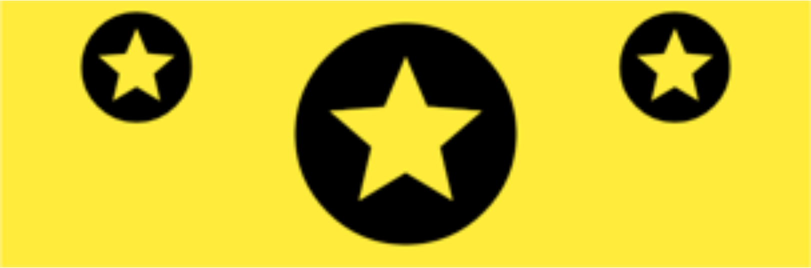
Note: setting MainAxisSize.min minimises the amount of free space along the main axis.



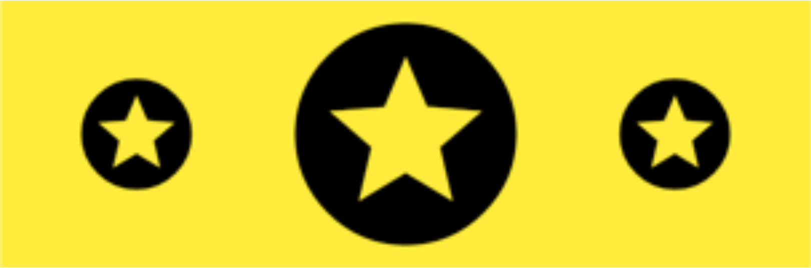
Row.crossAxisAlignment

```
Row(
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  crossAxisAlignment: _crossAxisAlignment,
  children: [
    Icon(Icons.stars, size: 50.0),
    Icon(Icons.stars, size: 100.0),
    Icon(Icons.stars, size: 50.0),
  ],
)
```

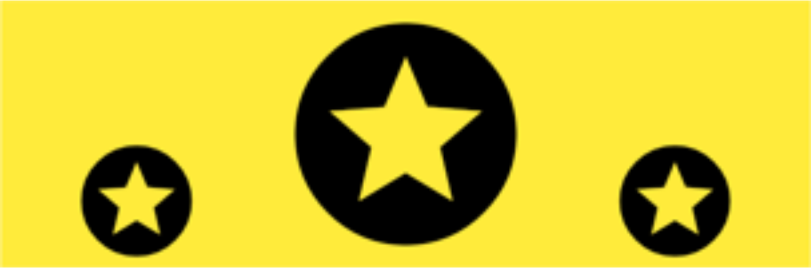
CrossAxisAlignment.start



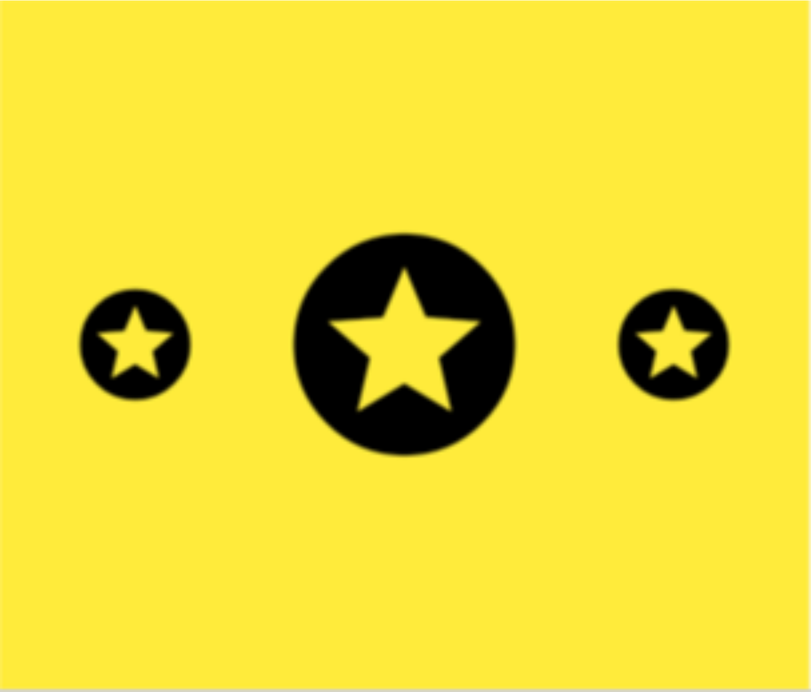
CrossAxisAlignment.center



CrossAxisAlignment.end



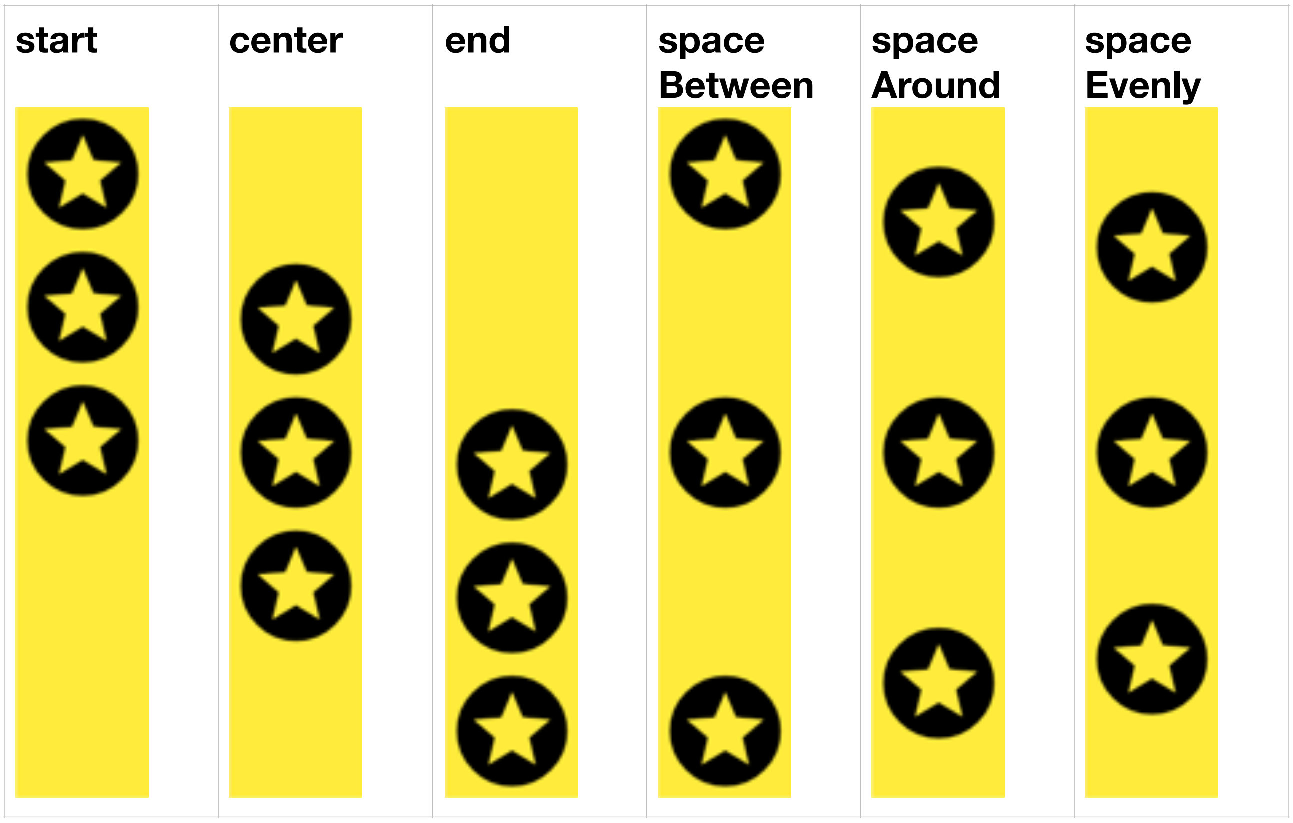
CrossAxisAlignment.stretch
(fills all the space in the cross axis)



Column.mainAxisAlignment

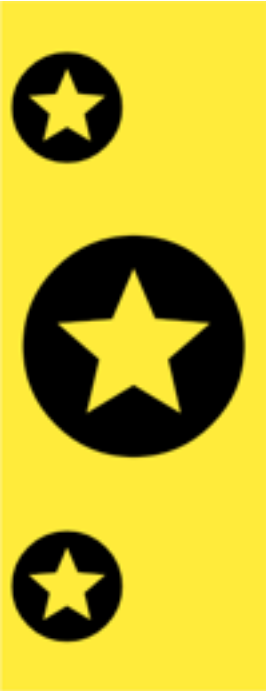
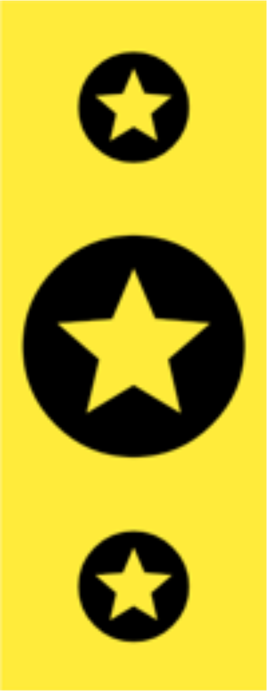
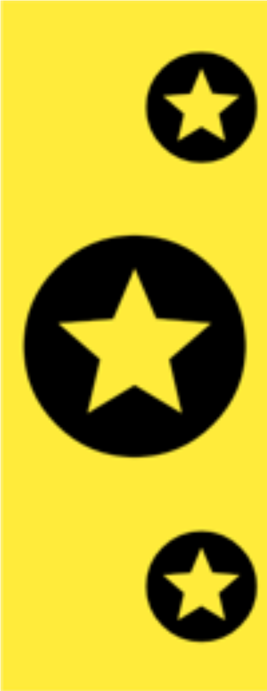
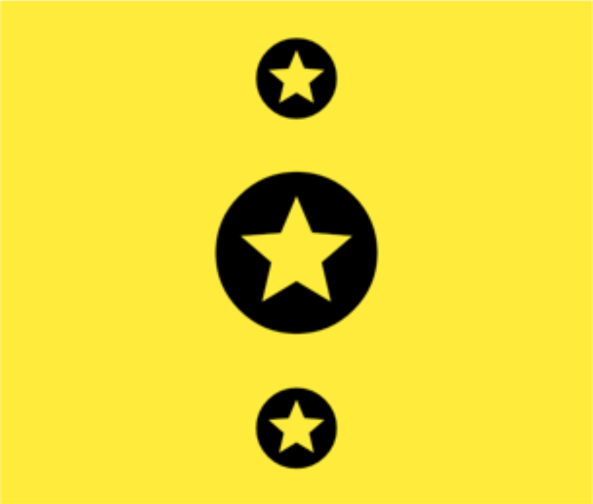
Columns behave in the same way as rows.
Main Axis: **Vertical**. Cross Axis: **Horizontal.**

```
Column(
  mainAxisAlignment: _mainAxisAlignment,
  mainAxisSize: MainAxisSize.max,
  children: [
    Icon(Icons.stars, size: 50.0),
    Icon(Icons.stars, size: 50.0),
    Icon(Icons.stars, size: 50.0),
  ],
)
```



Column.crossAxisAlignment

```
Column(
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  crossAxisAlignment: _crossAxisAlignment,
  children: [
    Icon(Icons.stars, size: 50.0),
    Icon(Icons.stars, size: 100.0),
    Icon(Icons.stars, size: 50.0),
  ],
)
```

CrossAxisAlignment. start 	**CrossAxisAlignment. center** 	**CrossAxisAlignment. end** 
CrossAxisAlignment.stretch 		

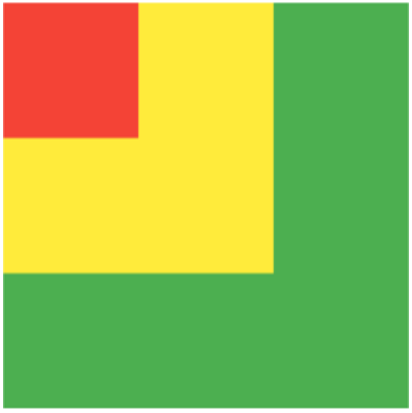
Stack

A stack positions its children relative to the edges of its box.

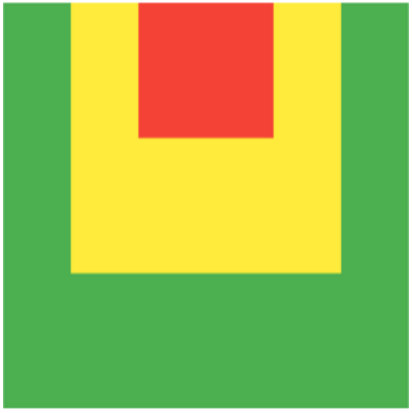
Stack.alignment

```
Stack(
  alignment: _alignmentDirectional,
  children:[
    SizedBox(
      width: 300.0,
      height: 300.0,
      child: Container(color: Colors.green),
    ),
    SizedBox(
      width: 200.0,
      height: 200.0,
      child: Container(color: Colors.yellow),
    ),
    SizedBox(
      width: 100.0,
      height: 100.0,
      child: Container(color: Colors.red),
    ),
  ],
)
```

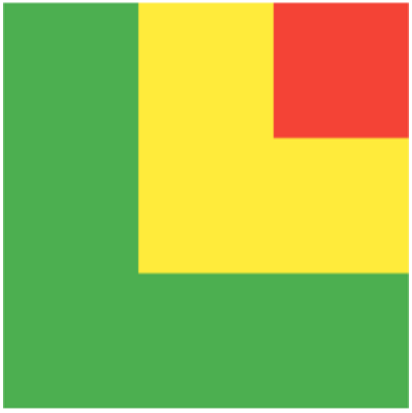
topStart



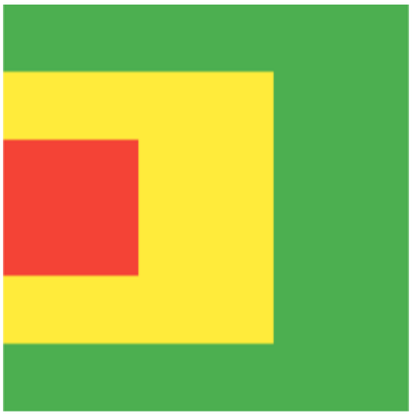
topCenter



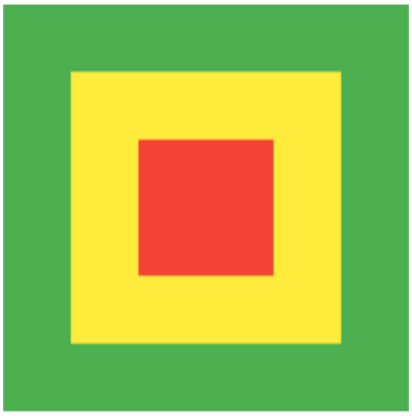
topEnd



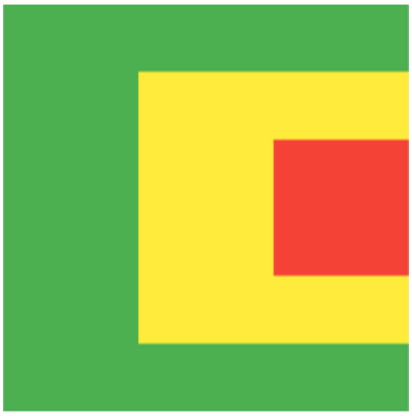
centerStart



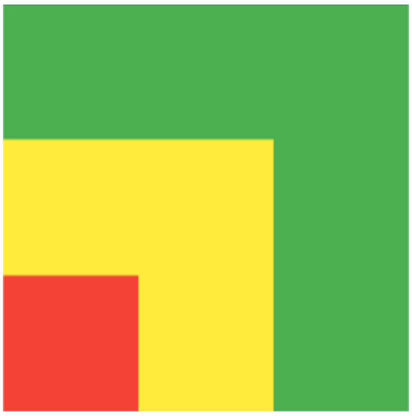
center



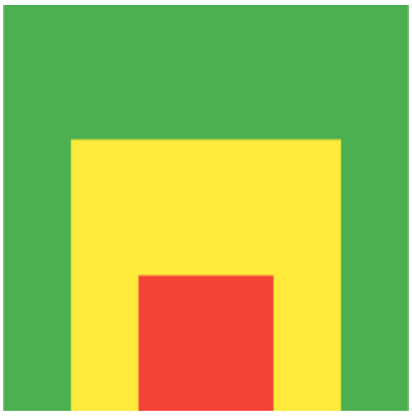
centerEnd



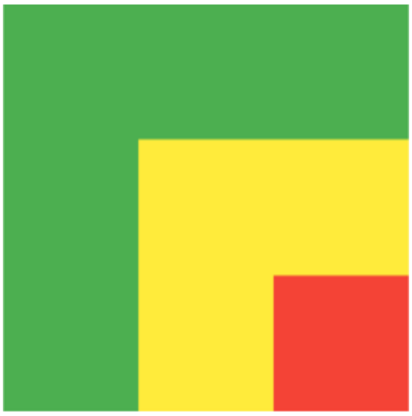
bottomStart



bottomCenter

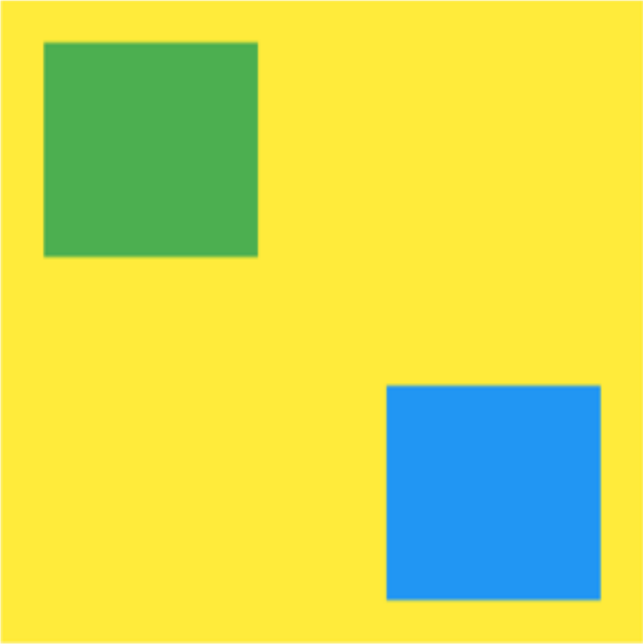


bottomEnd



Positioned

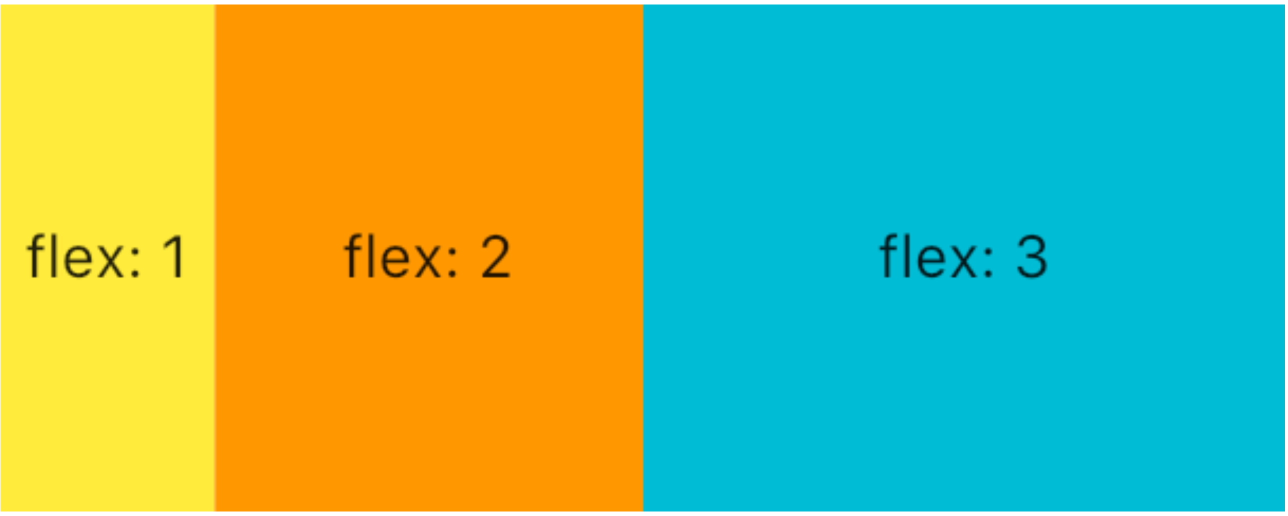
```
Stack(
  children: [
    SizedBox(
      width: 300.0,
      height: 300.0,
      child: Container(color: Colors.yellow),
    ),
    Positioned(
      left: 20.0,
      top: 20.0,
      width: 100.0,
      height: 100.0,
      child: Container(color: Colors.indigo),
    ),
    Positioned(
      bottom: 20.0,
      right: 20.0,
      width: 100.0,
      height: 100.0,
      child: Container(color: Colors.blue),
    ),
  ],
)
```



Expanded

A widget that expands a child of a Row or Column to fill the available space in the main axis.

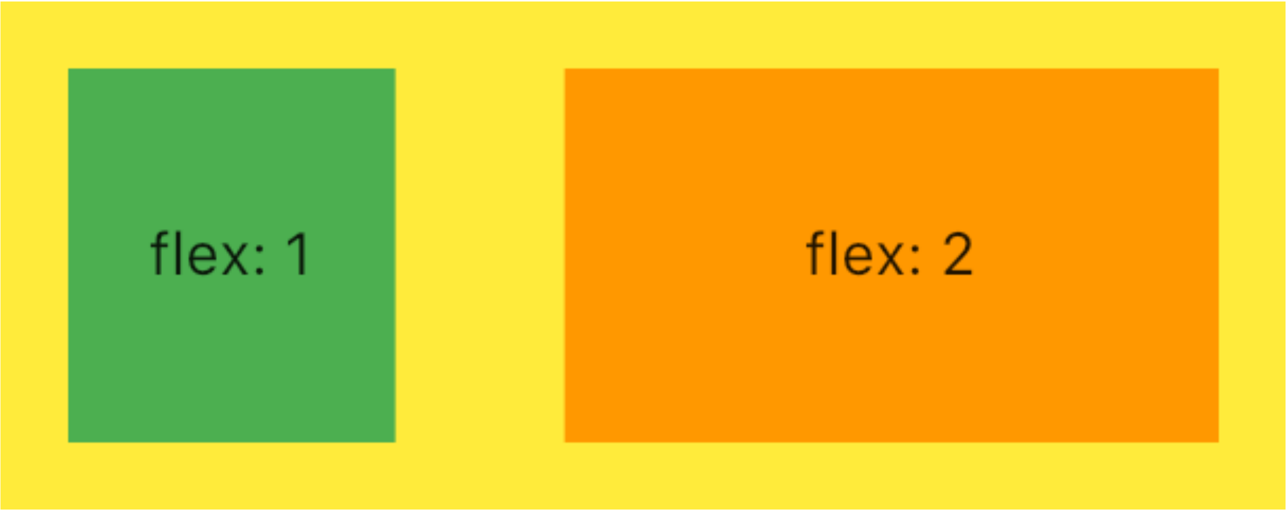
```
Row(
  crossAxisAlignment: CrossAxisAlignment.stretch,
  children: [
    Expanded(
      flex: 1,
      child: Container(
        color: Colors.yellow,
        child: Center(child: Text('flex: 1')),
      ),
    ),
    Expanded(
      flex: 2,
      child: Container(
        color: Colors.orange,
        child: Center(child: Text('flex: 2')),
      ),
    ),
    Expanded(
      flex: 3,
      child: Container(
        color: Colors.cyan,
        child: Center(child: Text('flex: 3')),
      ),
    ),
  ],
)
```



Expanded + Padding + SizedBox

This is how to use padding and a SizedBox to add space around and between content.

```
Container(
  color: Colors.yellow,
  padding: EdgeInsets.all(16.0),
  child: Row(
    children: [
      Expanded(
        flex: 1,
        child: Container(
          color: Colors.green,
          child: Center(child: Text('flex: 1')),
        ),
      ),
      SizedBox(width: 40.0),
      Expanded(
        flex: 2,
        child: Container(
          color: Colors.cyan,
          child: Center(child: Text('flex: 2')),
        ),
      ),
    ],
  ),
)
```


flex: 1
flex: 2