# Column

A widget that displays its children in a vertical array.

To cause a child to expand to fill the available vertical space, wrap the child in an Expanded widget.

If you only have one child, then consider using Align or Center to position the child.

In the following example, the crossAxisAlignment is set to CrossAxisAlignment.start, so that the children are left-aligned. The mainAxisSize is set to MainAxisSize.min, so that the column shrinks to fit the children.

import 'package:flutter/material.dart';

void main() => runApp(MyApp());

/// This Widget is the main application widget.

class MyApp extends StatelessWidget {

static const String \_title = 'Flutter Code Sample';

@override

Widget build(BuildContext context) {

return MaterialApp(

title: \_title,

home: Scaffold(

appBar: AppBar(title: const Text(\_title)),

body: MyStatelessWidget(),

),

);

}

}

/// This is the stateless widget that the main application instantiates.

class MyStatelessWidget extends StatelessWidget {

MyStatelessWidget({Key key}) : super(key: key);

@override

Widget build(BuildContext context) {

return Column(

crossAxisAlignment: CrossAxisAlignment.start,

mainAxisSize: MainAxisSize.min,

children: <Widget>[

Text('Deliver features faster'),

Text('Craft beautiful UIs'),

Expanded(

child: FittedBox(

fit: BoxFit.contain, // otherwise the logo will be tiny

child: const FlutterLogo(),

),

),

],

);

}

}



# Expanded

A widget that expands a child of a Row, Column, or Flex so that the child fills the available space.

Using an Expanded widget makes a child of a Row, Column, or Flex expand to fill the available space along the main axis (e.g., horizontally for a Row or vertically for a Column). If multiple children are expanded, the available space is divided among them according to the flex factor.

import 'package:flutter/material.dart';

void main() => runApp(MyApp());

/// This Widget is the main application widget.

class MyApp extends StatelessWidget {

static const String \_title = 'Flutter Code Sample';

@override

Widget build(BuildContext context) {

return MaterialApp(

title: \_title,

home: MyStatelessWidget(),

);

}

}

/// This is the stateless widget that the main application instantiates.

class MyStatelessWidget extends StatelessWidget {

MyStatelessWidget({Key key}) : super(key: key);

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Expanded Column Sample'),

),

body: Center(

child: Column(

children: <Widget>[

Container(

color: Colors.blue,

height: 100,

width: 100,

),

Expanded(

child: Container(

color: Colors.amber,

width: 100,

),

),

Container(

color: Colors.blue,

height: 100,

width: 100,

),

],

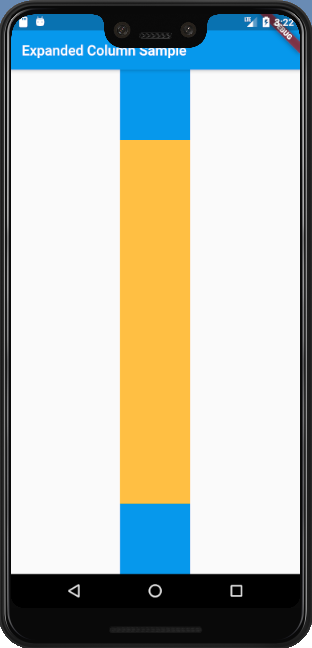
),

),

);

}

}



# Row

A widget that displays its children in a horizontal array.

To cause a child to expand to fill the available horizontal space, wrap the child in an Expanded widget.

import 'package:flutter/material.dart';

void main() => runApp(MyApp());

/// This Widget is the main application widget.

class MyApp extends StatelessWidget {

static const String \_title = 'Flutter Code Sample';

@override

Widget build(BuildContext context) {

return MaterialApp(

title: \_title,

home: MyStatelessWidget(),

);

}

}

/// This is the stateless widget that the main application instantiates.

class MyStatelessWidget extends StatelessWidget {

MyStatelessWidget({Key key}) : super(key: key);

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Expanded Row Sample'),

),

body: Center(

child: Row(

children: <Widget>[

Expanded(

flex: 2,

child: Container(

color: Colors.amber,

height: 100,

),

),

Container(

color: Colors.blue,

height: 100,

width: 50,

),

Expanded(

flex: 1,

child: Container(

color: Colors.amber,

height: 100,

),

),

],

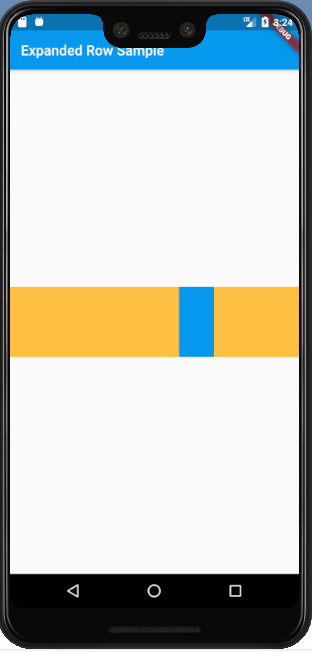
),

),

);

}

}



Flexible, which does not force the child to fill the available space.