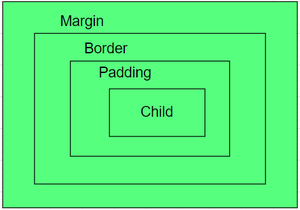
Container class in Flutter:

Container class in flutter is a convenience widget that combines common painting, positioning, and sizing of widgets. A Container class can be used to store one or more widgets and position them on the screen according to our convenience. Basically, a container is like a box to store contents. A basic container element that stores a widget has a margin, which separates the present container from other contents. The total container can be given a border of different shapes, for example, rounded rectangles, etc. A container surrounds its child with padding and then applies additional constraints to the padded extent (incorporating the width and height as constraints, if either is non-null).



**Syntax:**

Container({Key key,

AlignmentGeometry alignment,

EdgeInsetsGeometry padding,

Color color,

Decoration decoration,

Decoration foregroundDecoration,

double width,

double height,

BoxConstraints constraints,

EdgeInsetsGeometry margin,

Matrix4 transform,

Widget child,

Clip clipBehavior: Clip.none});

Properties of Container Class:

1. child: Container widget has a property ‘child:’ which stores its children. The child class can be any widget. Let us take an example, taking a text widget as a child.

import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget {

const MyApp({Key? key}) : super(key: key);

@override

Widget build(BuildContext context) {

return MaterialApp(

home: Scaffold(

appBar: AppBar(

title: const Text("Container example"),

),

body: Container(

child:const Text("Hello! i am inside a container!",

style: TextStyle(fontSize: 20)),

),

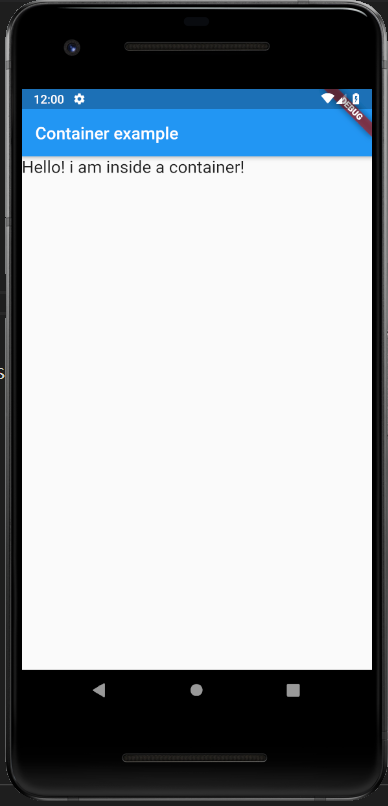
),

);

}

}

OUTPUT:



1. color: The color property sets the background color of the entire container. Now we can visualize the position of the container using a background color.

import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget {

const MyApp({Key? key}) : super(key: key);

@override

Widget build(BuildContext context) {

return MaterialApp(

home

: Scaffold(appBar

: AppBar(title

: const Text("Container example"),

),

body

: Container(color

: Colors.purple,

child

: const Text("Hello! i am inside a container!",

style

: TextStyle(fontSize : 20)),

),

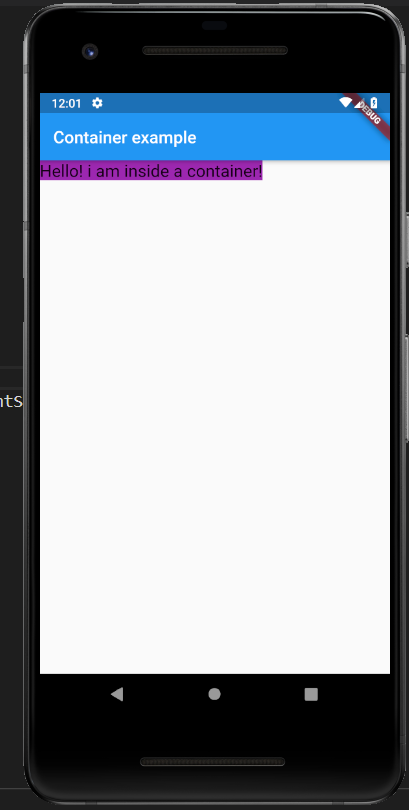
),

);

}

}

OUTPUT:



1. height and width: By default, a container class takes the space that is required by the child. We can also specify the height and width of the container based on our requirements.

import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget {

const MyApp({Key? key}) : super(key: key);

@override

Widget build(BuildContext context) {

return MaterialApp(

home: Scaffold(

appBar: AppBar(

title: const Text("Container example"),

),

body: Container(

height: 200,

width: double.infinity,

color: Colors.purple,

child: const Text("Hello! i am inside a container!",

style: TextStyle(fontSize: 20)),

),

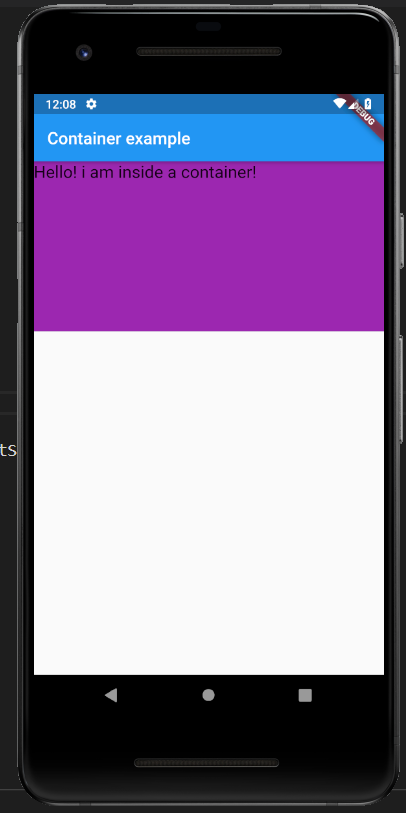
),

);

}

}

OUTPUT:



1. margin: The margin is used to create an empty space around the container. Observe the white space around the container. Here EdgeInsets.geometry is used to set the margin .all() indicates that the margin is present in all four directions equally.

import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget {

const MyApp({Key? key}) : super(key: key);

@override

Widget build(BuildContext context) {

return MaterialApp(

home: Scaffold(

appBar: AppBar(

title: const Text("Container example"),

),

body: Container(

height: 200,

width: double.infinity,

color: Colors.purple,

margin: const EdgeInsets.all(20),

child: const Text("Hello! i am inside a container!",

style: TextStyle(fontSize: 20)),

),

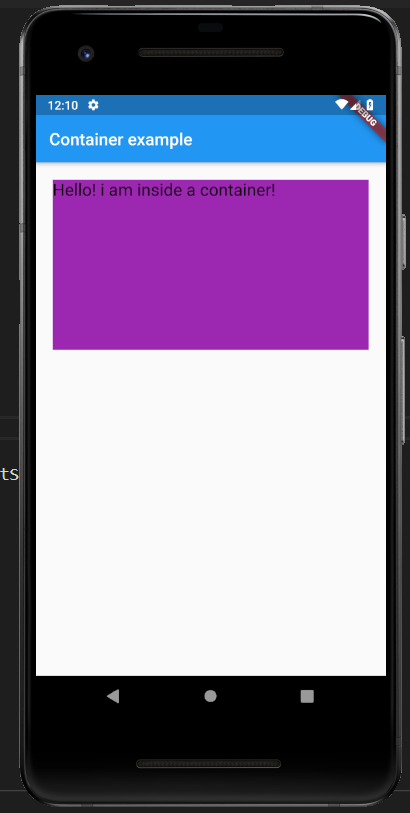
),

);

}

}

OUTPUT:



1. padding: The padding is used to give space from the border of the container from its children. Observe the space between the border and the text.

import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget {

const MyApp({Key? key}) : super(key: key);

@override

Widget build(BuildContext context) {

return MaterialApp(

home: Scaffold(

appBar: AppBar(

title: const Text("Container example"),

),

body: Container(

height: 200,

width: double.infinity,

color: Colors.purple,

margin: const EdgeInsets.all(20),

padding: const EdgeInsets.all(30),

child: const Text("Hello! i am inside a container!",

style: TextStyle(fontSize: 20)),

),

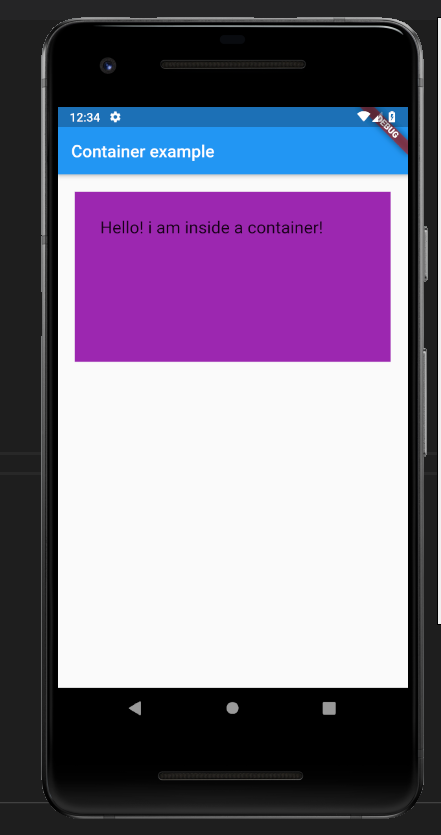
),

);

}

}

OUTPUT:



1. alignment: The alignment is used to position the child within the container. We can align in different ways: bottom, bottom center, left, right, etc. here the child is aligned to the bottom center.

import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget {

const MyApp({Key? key}) : super(key: key);

@override

Widget build(BuildContext context) {

return MaterialApp(

home: Scaffold(

appBar: AppBar(

title: const Text("Container example"),

),

body: Container(

height: 200,

width: double.infinity,

color: Colors.purple,

alignment: Alignment.bottomCenter,

margin: const EdgeInsets.all(20),

padding: const EdgeInsets.all(30),

child: const Text("Hello! i am inside a container!",

style: TextStyle(fontSize: 20)),

),

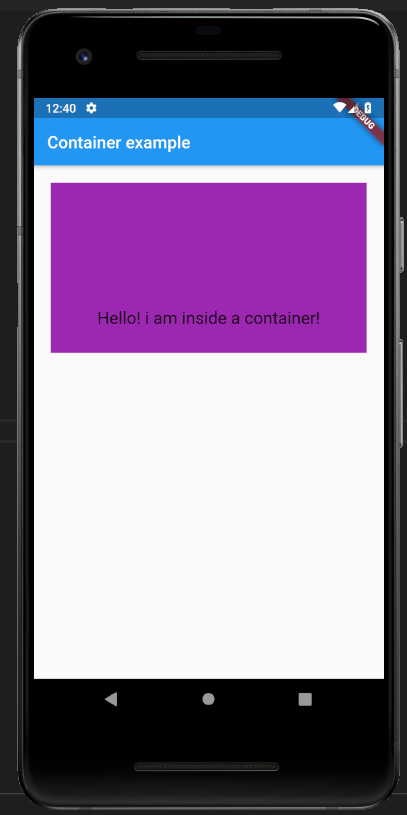
),

);

}

}

OUTPUT:



1. Decoration: The decoration property is used to decorate the box(e.g. give a border). This paints behind the child. Whereas foreground Decoration paints in front of a child. Let us give a border to the container. But, both color and border color cannot be given.

import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget {

const MyApp({Key? key}) : super(key: key);

@override

Widget build(BuildContext context) {

return MaterialApp(

home: Scaffold(

appBar: AppBar(

title: const Text("Container example"),

),

body: Container(

height: 200,

width: double.infinity,

//color: Colors.purple,

alignment: Alignment.center,

margin: const EdgeInsets.all(20),

padding: const EdgeInsets.all(30),

decoration: BoxDecoration(

border: Border.all(color: Colors.black, width: 3),

),

child: const Text("Hello! i am inside a container!",

style: TextStyle(fontSize: 20)),

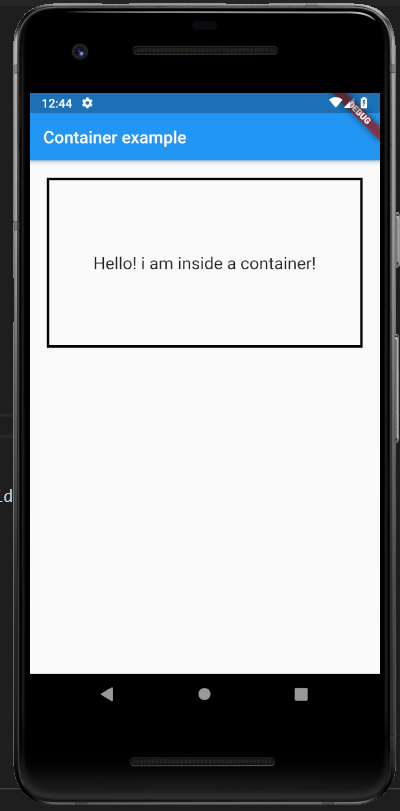
),

),

);

}

}



1. Transform: This property of the container helps us to rotate the container. We can rotate the container in any axis, here we are rotating in the z-axis.

import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget {

const MyApp({Key? key}) : super(key: key);

@override

Widget build(BuildContext context) {

return MaterialApp(

home: Scaffold(

appBar: AppBar(

title: const Text("Container example"),

),

body: Container(

height: 200,

width: double.infinity,

color: Colors.purple,

alignment: Alignment.center,

margin: const EdgeInsets.all(20),

padding: const EdgeInsets.all(30),

transform: Matrix4.rotationZ(0.1),

child: const Text("Hello! i am inside a container!",

style: TextStyle(fontSize: 20)),

),

),

);

}

}

OUTPUT:

