



Al-Mustaqbal University
College of Science
Intelligent Medical System Department



College of Sciences
Intelligent Medical System
Department



Lecture 1

Container class in Flutter

part (1)

Subject: Mobile Applications

Level: Third

Semester: second

Lecturer: Asst. Lect. Ali Saleem Haleem

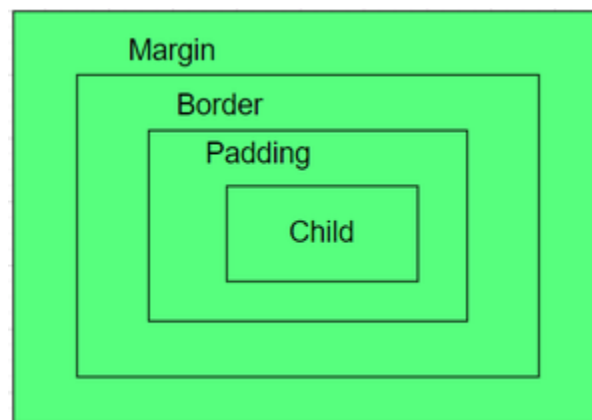


Google Class Room



Introduction

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).



Properties of Container Class

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

Example:

```
import 'package:flutter/material.dart';

// Entry point of the application
void main() => runApp(const MyApp());

// Main application widget
class MyApp extends StatelessWidget
{
```



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

@override
Widget build(BuildContext context) {
  return MaterialApp(

    // Disable the debug banner
    debugShowCheckedModeBanner: false,
    home: Scaffold(

      // App bar with a title
      appBar: AppBar(
        backgroundColor: Colors.blue,
        title: const Text(
          "Container example",
          style: TextStyle(color: Colors.white),
        ),
      ),

      // Body of the scaffold with a container
      body: Container(
        child: const Text(
          "Hello! I am inside a container!",
          style: TextStyle(fontSize: 20),
        ),
      ),
    ),
  );
}
```



2. 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.

Example:

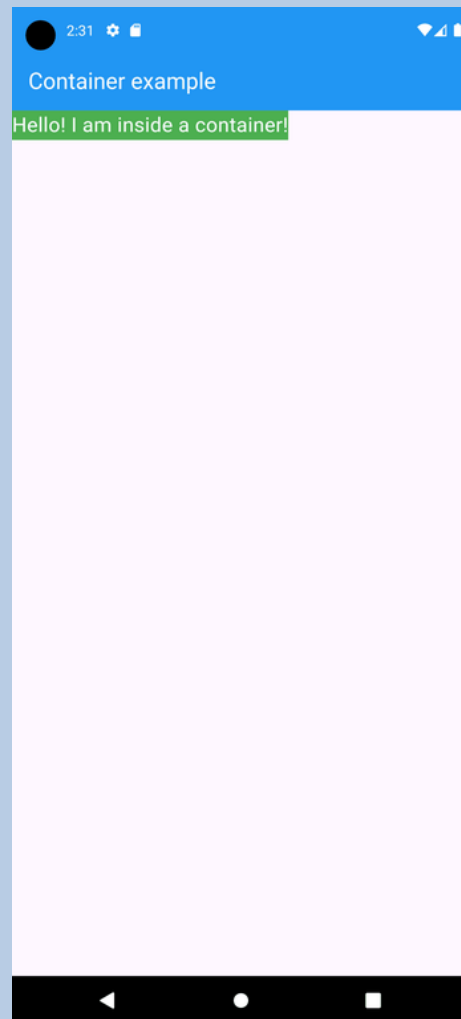
```
import 'package:flutter/material.dart';

// Entry point of the application
void main() => runApp(const MyApp());

// Main application widget
```



```
class MyApp extends StatelessWidget {  
  const MyApp({Key? key}) : super(key: key);  
  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
  
      // Disable the debug banner  
      debugShowCheckedModeBanner: false,  
      home: Scaffold(  
  
        // App bar with a title  
        appBar: AppBar(  
          backgroundColor: Colors.blue,  
          title: const Text(  
            "Container example",  
            style: TextStyle(color: Colors.white),  
          ),  
        ),  
  
        // Body of the scaffold with a container  
        body: Container(  
  
          // Background Color of the Container  
          color: Colors.green,      child: const Text(  
            "Hello! I am inside a container!",  
            style: TextStyle(fontSize: 20,  
              color: Colors.white  
            ),  
          ),  
        ),  
      ),  
    );  
  }  
}
```



3. 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.



Example:

```
import 'package:flutter/material.dart';

// Entry point of the application
void main() => runApp(const MyApp());

// Main application widget
class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

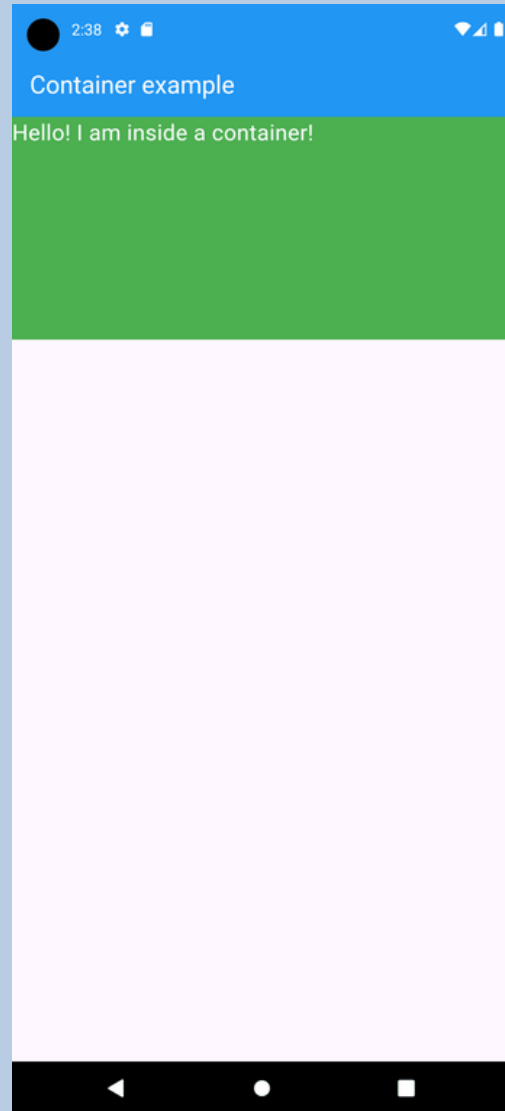
  @override
  Widget build(BuildContext context) {
    return MaterialApp(

      // Disable the debug banner
      debugShowCheckedModeBanner: false,
      home: Scaffold(

        // App bar with a title
        appBar: AppBar(
          backgroundColor: Colors.blue,
          title: const Text(
            "Container example",
            style: TextStyle(color: Colors.white),
          ),
        ),

        // Body of the scaffold with a container
        body: Container(
          // Height of the container
          height: 200,

          // Width of the container
          width: double.infinity,
          color: Colors.green,
          child: const Text(
            "Hello! I am inside a container!",
            style: TextStyle(fontSize: 20,
              color: Colors.white
            ),
          ),
        ),
      ),
    );
  }
}
```



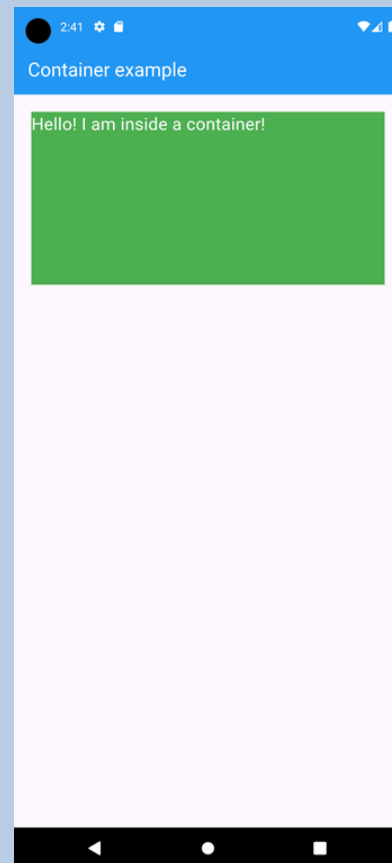


```
),  
);  
}  
}
```

4. 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.

Example:

```
import 'package:flutter/material.dart';  
  
// Entry point of the application  
void main() => runApp(const MyApp());  
  
// Main application widget  
class MyApp extends StatelessWidget {  
  const MyApp({Key? key}) : super(key: key);  
  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
  
      // Disable the debug banner  
      debugShowCheckedModeBanner: false,  
      home: Scaffold(  
  
        // App bar with a title  
        appBar: AppBar(  
          backgroundColor: Colors.blue,  
          title: const Text(  
            "Container example",  
            style: TextStyle(color: Colors.white),  
          ),  
        ),  
  
        // Body of the scaffold with a container  
        body: Container(  
          height: 200, // Height of the container
```





Al-Mustaqbal University
College of Science
Intelligent Medical System Department

```
width: double.infinity, // Width of the container
margin: const EdgeInsets.all(20), // Margin of the container
color: Colors.green,
child: const Text(
  "Hello! I am inside a container!",
  style: TextStyle(fontSize: 20,
    color: Colors.white
  ),
),
),
),
),
);
}
```



Homework

Practical Tasks

1. Create a new Flutter project named: lec1_container_basics.
2. Build **one** screen that contains **3 Containers** arranged vertically (using Column):
 - **Container (A):** Display a clear text inside the container using child.
 - **Container (B):** Use the color property to change the container background + choose a readable text color.
 - **Container (C):** Set height and width (example: height = 200, width = double.infinity).
3. Add margin to **Container (C)** using EdgeInsets.all(20), and write **one sentence** (as a code comment) explaining why margin is used.

Theory Tasks

- Write **3 lines only** explaining the difference between **margin** and **padding** (no code).

Submission

Upload:

1. One screenshot of the final UI.
2. main.dart.

Note: Submit the homework via Google Classroom under the lecture's Homework assignment. Upload the required files before the deadline.