



Al-Mustaqbal University
College of Science
Intelligent Medical System Department



College of Sciences
Intelligent Medical System
Department



Lecture 2

Container class in Flutter *part (2)*

Subject: Mobile Applications

Level: Third

Semester: second

Lecturer: Asst. Lect. Ali Saleem Haleem



Google Class Room



5. 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.

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(

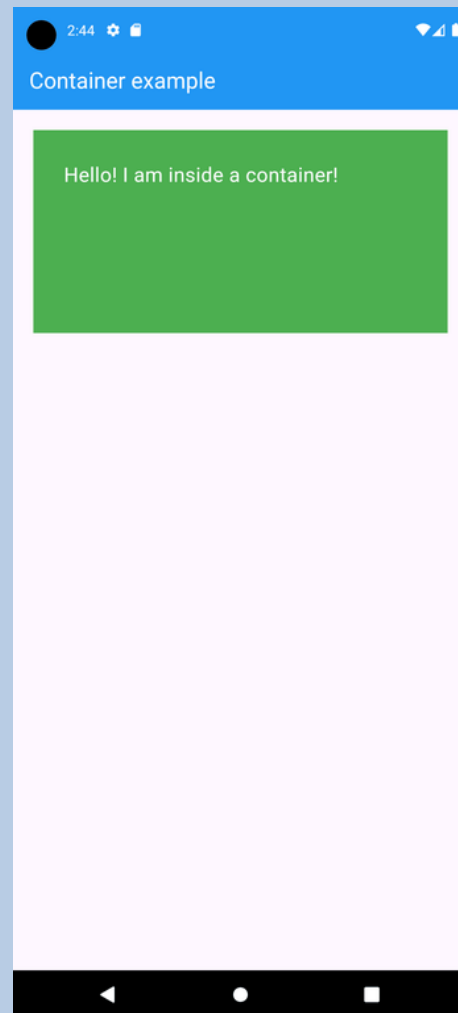
        // App bar with a title
        appBar: AppBar(
          backgroundColor: Colors.blue,
          title: const Text(
            "Container example",
            style: TextStyle(color: Colors.white),
          ),
        ),

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

          // Width of the container
          width: double.infinity,

          // Margin of the container
          margin: const EdgeInsets.all(20),

          // Padding of the container
```





```
padding: const EdgeInsets.all(30),
color: Colors.green,
child: const Text(
  "Hello! I am inside a container!",
  style: TextStyle(fontSize: 20,
    color: Colors.white
  ),
),
),
),
),
);
}
```

6. 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.

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,  

```

```
  // Alignment of the container  
  alignment: Alignment.bottomCenter,  

```

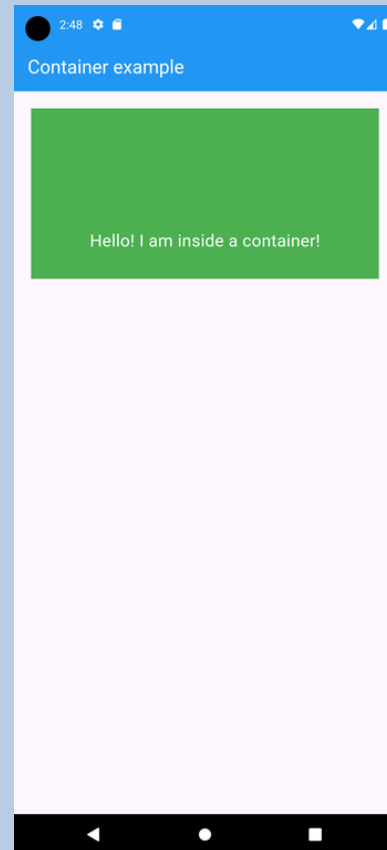
```
  // Margin of the container  
  margin: const EdgeInsets.all(20),  

```

```
  // Padding of the container  
  padding: const EdgeInsets.all(30),  
  color: Colors.green,  
  child: const Text(  
    "Hello! I am inside a container!",  
    style: TextStyle(fontSize: 20,  
      color: Colors.white  

```

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



7. 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. if you use the color outside the decoration property.



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),
          ),
        ),

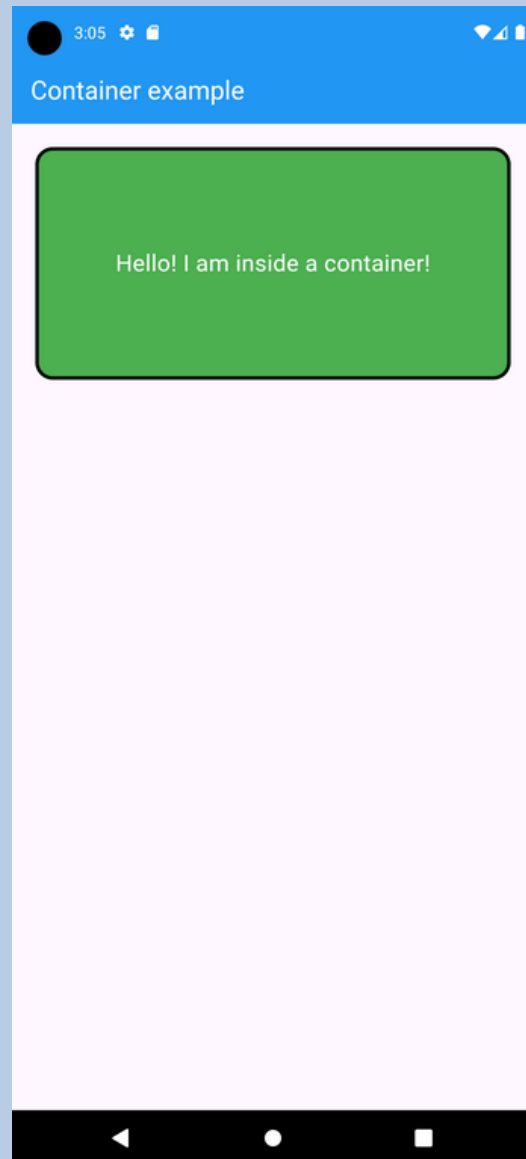
        // Body of the scaffold with a container
        body: Container(

          // Height of the container
          height: 200,

          // Width of the container
          width: double.infinity,

          // Alignment of the container
          alignment: Alignment.center,

          // Margin of the container
          margin: const EdgeInsets.all(20),
```





```
// Padding of the container
padding: const EdgeInsets.all(30),

// Removed the color property from here to avoid assertion error
// color: Colors.green,
decoration: BoxDecoration(

  // Use color here when using the decoration property
  color: Colors.green,

  // Add rounded corners
  borderRadius: BorderRadius.circular(15),
  border: Border.all(color: Colors.black, width: 3),
),
child: const Text(
  "Hello! I am inside a container!",
  style: TextStyle(fontSize: 20,
    color: Colors.white
  ),
),
),
),
);
}
```

8. 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.

Example:

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

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

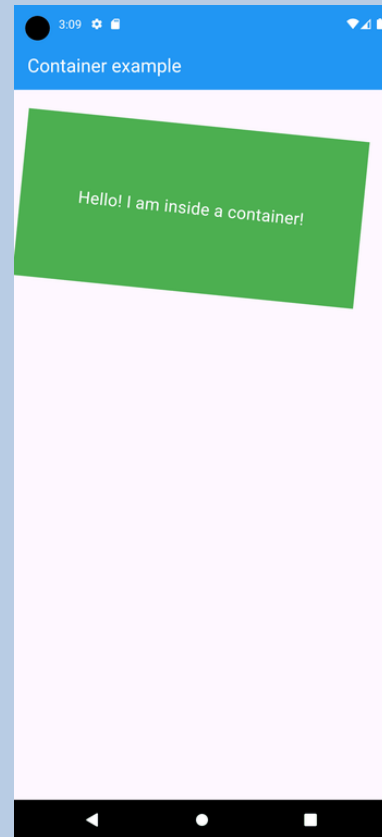
// Main application widget
class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
```



Al-Mustaqbal University
College of Science
Intelligent Medical System Department

```
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,  
  
      // Alignment of the container  
      alignment: Alignment.center,  
  
      // Margin of the container  
      margin: const EdgeInsets.all(20),  
  
      // Padding of the container  
      padding: const EdgeInsets.all(30),  
  
      // Removed the color property from here to avoid assertion error  
      color: Colors.green,  
  
      // Rotation of the container  
      transform: Matrix4.rotationZ(0.1),  
      child: const Text(  
        "Hello! I am inside a container!",  
        style: TextStyle(fontSize: 20,  
          color: Colors.white  
        ),  
      ),  
    ),  
  ),  
);
```





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

9. Constraints: When we want to give additional constraints to the child, we can use this property.

10. ClipBehaviour: This property takes in *Clip* Enum as the object. This decides whether the content inside the container will be clipped or not.

11. Foreground Decoration: This parameter holds *Decoration* class as the object. It controls the decoration in front of the Container widget.



Homework

Practical Tasks

1. Create a new Flutter project named: lec2_container_advanced.
2. Build one main Container that includes:
 - padding: EdgeInsets.all(30)
 - alignment: Alignment.bottomCenter (or make a second version using Alignment.center)
 - decoration: BoxDecoration(...) that must include:
 - background color **inside decoration**
 - borderRadius
 - border
3. Create a second version of the same Container using:
 - transform: Matrix4.rotationZ(0.1)Then provide **two screenshots**: before and after rotation.

Theory Tasks (2 lines only)

- Why is it better not to use color outside decoration when using border?

Submission

Upload:

1. Two screenshots (before/after rotation)
2. main.dart

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