Al-Mustaqbal University
College of Science
Intelligent Medical System Department

College of Sciences
Intelligent Medical System Department

Lecture 3

SizedBox Widget in Flutter

Subject: Mobile Applications

Level: Third

Semester: second

Lecturer: Asst. Lect. Ali Saleem Haleem

Google Class Room

Study Year: 2025-2026

Introduction

SizedBox is a built-in widget in flutter SDK. It is a simple box with a specified size. It can be used to set size constraints to the child widget, put an empty SizedBox between the two widgets to get some space in between, or something else. It is somewhat similar to a Container widget with fewer properties.

Constructor of SizedBox Class

It draws a simple box with the mentioned height and width or a child widget inside.

```
const SizedBox(
{
    Key key,
    double width,
    double height,
    Widget child}
)
```

Constructor of SizedBox.expand

This implementation of the SizedBox widget allows it to be as big as the parent widget allows it to be.

```
const SizedBox.expand(
{
    Key key,
    Widget child}
)
```

Constructor of SizedBox.fromSize

This allows the creation of a **SizedBox** with a specified size.

```
SizedBox.fromSize(
{
    Key key,
    Widget child,
    Size size}
)
```

Constructor of SizedBox.shrink

This implementation of the widget allows it to be as small as the child widget allows it to be.

```
const SizedBox.shrink(
{
    Key key,
    Widget child}
)
```

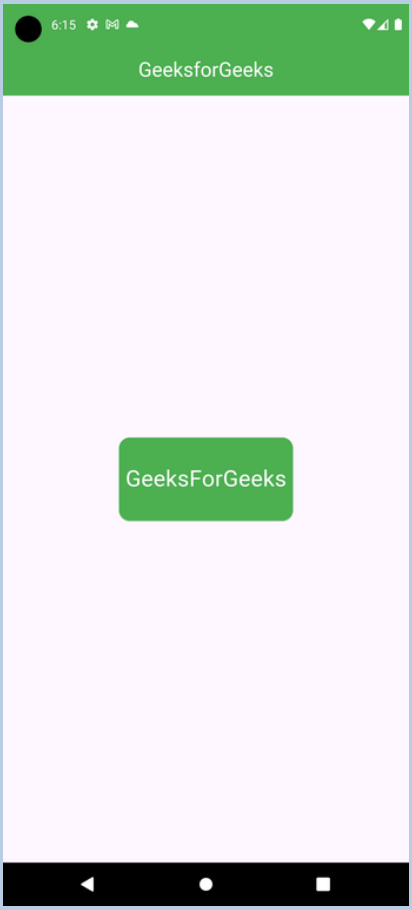
Properties of SizedBox Widget

Property	Description
child	This property takes in a child widget as the object to display it below the SizedBox in the widget tree or inside the SizedBox on the screen.
height	This property specifies the height of the SizedBox in pixels. It is a double value as the object.
width	This property is also a double value as the object to give width to the *SizedBox*.

Example 1:

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

//Importing material design library
void main() {
 runApp(
   //Our app widget tree starts from here
   MaterialApp(
     debugShowCheckedModeBanner: false,
     home: Scaffold(
       appBar: AppBar(
         title: Text('GeeksforGeeks'),
         centerTitle: true,
         backgroundColor: Colors.green,
         foregroundColor: Colors.white,
       ), //AppBar
       body: Center(
         //SizedBox Widget
         child: SizedBox(
           width: 200.0,
           height: 100.0,
           child: Card(
             color: Colors.green,
             child: Center(
               child: Text(
                 'GeeksForGeeks',
                 style: TextStyle(color: Colors.white, fontSize: 25),
               ), //Text
             ), //Center
           ), //Card
         ), //SizedBox
       ), //Center
     ), //Scaffold
   ), //MaterialApp
 );
}
```

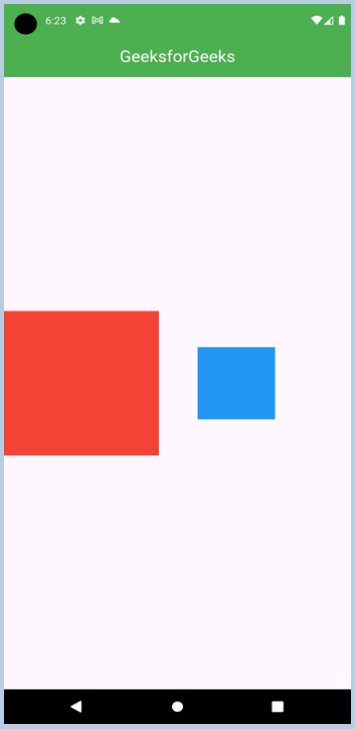


Using SizedBox.expand

```
SizedBox.expand(
   child: Card(
     color: Colors.green,
     child: Center(
       child: Text(
         'GeeksForGeeks',
         style: TextStyle(color: Colors.white,
         fontSize: 25
         ),
       ), //Text
     ), //Center
   ), //Card
 ), //SizedBox.expand
```

Example 2:

```
Center(
  child: Row(
    children: <Widget>[
      Container(
        width: 200,
        height: 200,
        color: Colors.red,
      ), //Container
      //SizedBox Widget
      SizedBox(
        width: 50,
      ),
      Container(
        width: 100,
        height: 100,
        color: Colors.blue,
      ) //Container
    ], //<Widget>[]
  ), //Row
), //center
```



Homework

Practical Tasks

1. Create a new Flutter project named: lec3_sizedbox.
2. Implement **3 examples** on the same screen (you may use Tabs or a Column with headings):

- **Example 1:** Use SizedBox to control width and height of a widget (Card or Container).
- **Example 2:** In a Row, place a red Container and a blue Container with space between them using SizedBox(width: 50).
- **Example 3:** Use SizedBox.expand to fill the available space inside the parent widget.

Theory Tasks (1 line only)

- When would you choose SizedBox instead of Container?

Submission

Upload:

1. One screenshot showing all examples (or 3 screenshots).
2. main.dart

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