

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

الجاهزة عن الواجهات المسؤولة AppBar Scaffold Text Colors

```
Void main(){
```

runApp(MyApp); // لازم نعرفها تحت بـ class هنا راح يطلع خطأ

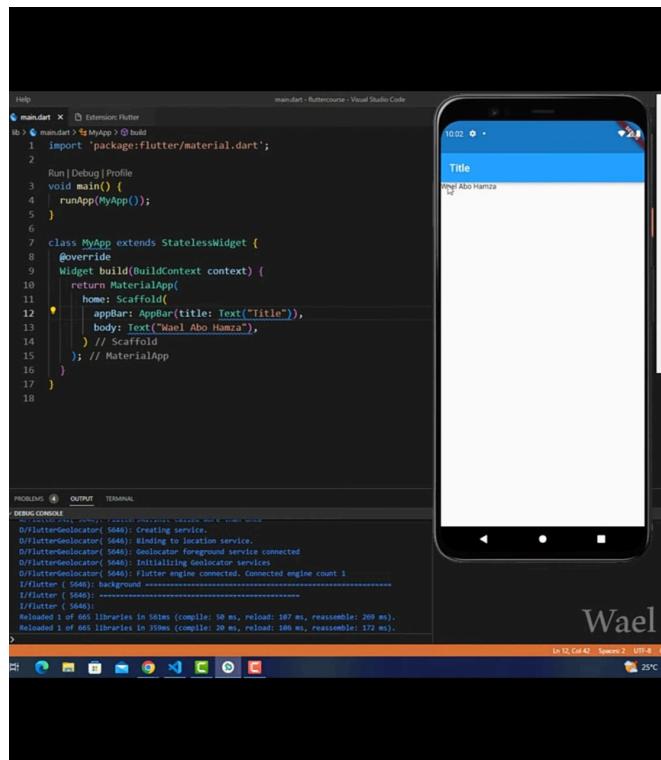
```
}
```

```
runApp(MyApp)
```

نقطة بداية أي تطبيق Flutter

runApp تشغيل الودجت الرئيسية للتطبيق

MyApp هي الودجت الأساسية



و Title يظهر بالأأسفل فوق العنوان وال text راح يظهر

عنوان العنواني scaffold يرتب الصفحة ويضيف عناصر أساسية مثل body وشريط home بدل ما تبني كل من الصفر

The screenshot shows the Android Studio interface. On the left, the code editor displays a Dart file named `main.dart` with the following content:

```

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: Text("Title")),
        body: Text("Wael Abo Hamza", style: TextStyle(
          color: Color(0xff3498db),
          fontSize: 40.2
        )), // TextStyle // Text
      ) // Scaffold
    ); // MaterialApp
}

```

The terminal output window shows the build process:

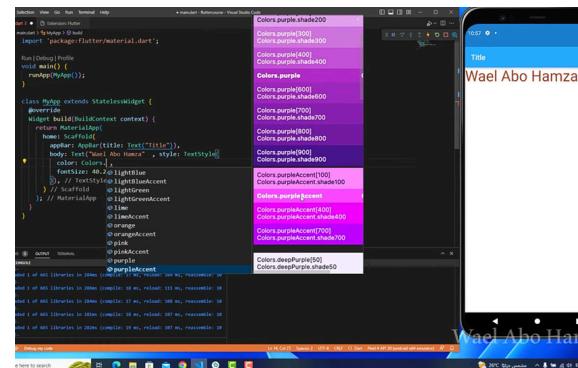
```

Reloaded 1 of 665 libraries in 280ms (compile: 19 ms, reload: 108 ms, reassemble: 10
1 ms).
Reloaded 1 of 665 libraries in 284ms (compile: 17 ms, reload: 109 ms, reassemble: 10
4 ms).
Reloaded 1 of 665 libraries in 286ms (compile: 18 ms, reload: 111 ms, reassemble: 10
8 ms).
Reloaded 1 of 665 libraries in 284ms (compile: 17 ms, reload: 108 ms, reassemble: 10
9 ms).
Reloaded 1 of 665 libraries in 281ms (compile: 18 ms, reload: 107 ms, reassemble: 10
4 ms).

```

The emulator on the right shows the app running with the title "Title" and the text "Wael Abo Hamza" in a purple font.

ونستطيع إجراء التغييرات على الخط واللون والحجم والسمك في الأعلى موضح ونستطيع الحصول على اللون من خلال تطبيق الفوتوشوب



ولو ما بذنا كود بنستطيع إضافة colors ونحصل على كل الألوان

```

File Edit Selection View Go Run Terminal Help
main.dart - Flutter
lib > main.dart > MyApp > build
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   @override
9   Widget build(BuildContext context) {
10    return MaterialApp(
11      home: Scaffold(
12        appBar: AppBar(title: const Text("Title")),
13        body: const Text("Wael Abo Hamza" , style: TextStyle(
14          color: Color.fromRGBO(255, 91, 185, 96) ,
15          fontSize: 40.2 ,
16          fontWeight: FontWeight.w900
17        )), // TextStyle // Text
18      ) // Scaffold
19    ); // MaterialApp
20  }
21 }
22

```

PROBLEMS OUTPUT TERMINAL

DEBUG CONSOLE

```

Reloaded 1 of 665 libraries in 263ms (compile: 13 ms, reload: 100 ms, reassemble: 98 ms).
Reloaded 1 of 665 libraries in 279ms (compile: 12 ms, reload: 109 ms, reassemble: 11 0 ms).
Reloaded 1 of 665 libraries in 282ms (compile: 17 ms, reload: 107 ms, reassemble: 10 7 ms).

```

Wael Abo Hamza

إضافة const يزيد من أداء التطبيق

```

File Edit Selection View Go Run Terminal Help
main.dart - Flutter
lib > main.dart > MyApp > build
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   @override
9   Widget build(BuildContext context) {
10    return MaterialApp(
11      home: Scaffold(
12        appBar: AppBar(title: const
13          EdgeInsets.all(double value)
Creates insets where all the offsets are value .
Typical eight-pixel margin on all sides:
const EdgeInsets.all(8.0)
14        color: Colors.red,
15        padding: EdgeInsets.all(10),
16        child: const Text("Wael Abo Hamza",
17          style: TextStyle(
18            color: Color.fromRGBO(255, 91, 185, 96),
19            fontSize: 40.2,
20            fontWeight: FontWeight.w900)), // TextStyle // Container
21      ); // Scaffold // MaterialApp
22  }
23 }

```

PROBLEMS OUTPUT TERMINAL

DEBUG CONSOLE

```

Reloaded 1 of 665 libraries in 283ms (compile: 17 ms, reload: 100 ms, reassemble: 10 8 ms).
Reloaded 1 of 665 libraries in 294ms (compile: 17 ms, reload: 111 ms, reassemble: 11 1 ms).
Reloaded 1 of 665 libraries in 298ms (compile: 17 ms, reload: 107 ms, reassemble: 10 6 ms).
Reloaded 1 of 665 libraries in 291ms (compile: 18 ms, reload: 107 ms, reassemble: 11 6 ms).
Reloaded 1 of 665 libraries in 292ms (compile: 17 ms, reload: 100 ms, reassemble: 11 6 ms).

```

Wael Abo Hamza

:padding مسافة داخلية من كل الجهات

:color لون النص أخضر

:fontSize حجم الخط

:fontWeight سمك الخط ثقيل جدا

The screenshot shows the Android Studio interface. On the left is the code editor with Dart code for a Flutter application. The code defines a `MyApp` class that extends `StatelessWidget`. It sets up a `Scaffold` with a red `AppBar` titled "Title". The `body` contains a `Container` with a red background color, alignment set to `Alignment.bottomLeft`, and padding of `EdgeInsets.all(20)`. Inside the container is a green `Text` widget with the text "Wael Abo Hamza". The font style is `TextStyle` with a color of `Color.fromARGB(255, 91, 185, 96)`, a font size of `40.2`, and a font weight of `FontWeight.w900`. On the right, a smartphone screen displays the app with the title "Title" at the top and the text "Wael Abo Hamza" in green at the bottom-left corner of the orange background.

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   @override
9   Widget build(BuildContext context) {
10     return MaterialApp(
11       home: Scaffold(
12         appBar: AppBar(title: const Text("Title")),
13         body: Container(
14           alignment: Alignment.bottomLeft,
15           color: Colors.red,
16           margin: EdgeInsets.all(10),
17           padding: EdgeInsets.all(20),
18           child: const Text("Wael Abo Hamza",
19             style: TextStyle(
20               color: Color.fromARGB(255, 91, 185, 96),
21               fontSize: 40.2,
22               fontWeight: FontWeight.w900)),
23         )));
24 // Scaffold // MaterialApp
25 }
```

مسافات خارجية

يبيخلي النص في اليسار أَسفل

و هنا جعل background على طول الصفحة و نستطيع تقليله بتغيير حجم margin

The screenshot shows the Android Studio interface. The code is identical to the previous one, except the `alignment` in the `Container` is now `Alignment.topRight`. This changes the text position from bottom-left to top-right. The text "Wael Abo Hamza" is now located at the top-right corner of the orange background. The rest of the UI elements remain the same.

```
1 void main() {
2   runApp(MyApp());
3 }
4
5 class MyApp extends StatelessWidget {
6   @override
7   Widget build(BuildContext context) {
8     return MaterialApp(
9       home: Scaffold(
10         appBar: AppBar(title: const Text("Title")),
11         body: Container(
12           alignment: Alignment.topRight,
13           color: Colors.red,
14           margin: EdgeInsets.all(10),
15           padding: EdgeInsets.all(20),
16           child: const Text("Wael Abo Hamza",
17             style: TextStyle(
18               color: Color.fromARGB(255, 91, 185, 96),
19               fontSize: 40.2,
20               fontWeight: FontWeight.w900)),
21         )));
22 // Scaffold // MaterialApp
23 }
```

قمنا بتغييره لفوق على اليمين

```

main.dart
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: const Text("Title")),
        body: Container(
          width: 400,
          height: 400,
          alignment: Alignment.topRight,
          color: Colors.red,
          margin: EdgeInsets.all(10),
          padding: EdgeInsets.all(20),
          child: const Text("Wael Abo Hamza",
            style: TextStyle(
              color: Color.fromARGB(255, 91, 185, 96),
              fontSize: 40.2,
              fontWeight: FontWeight.w900)),
        ), // Scaffold // MaterialApp
      ),
    );
  }
}

```

ويكونوا داخل container

الارتفاع أو الطول الرأسى height

المسافة من اليمين إلى اليسار width.

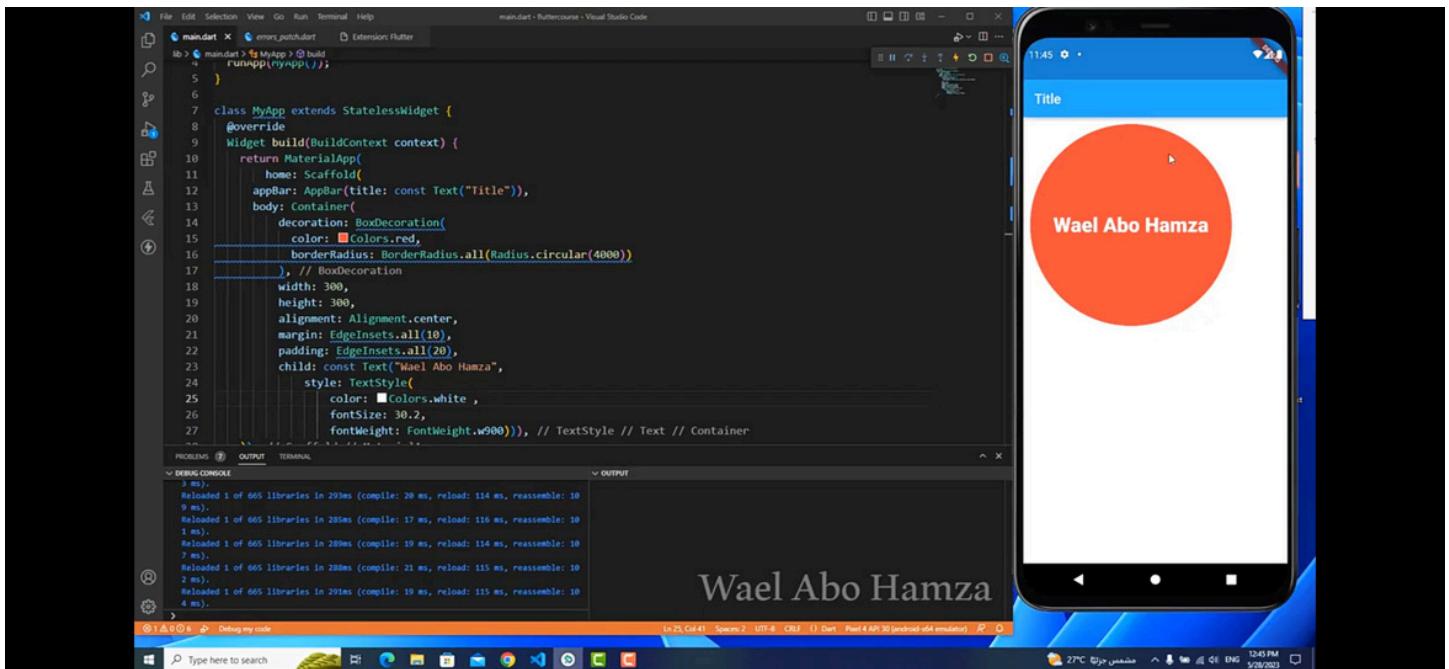
```

main.dart
void main() {
  runApp(MyApp());
}

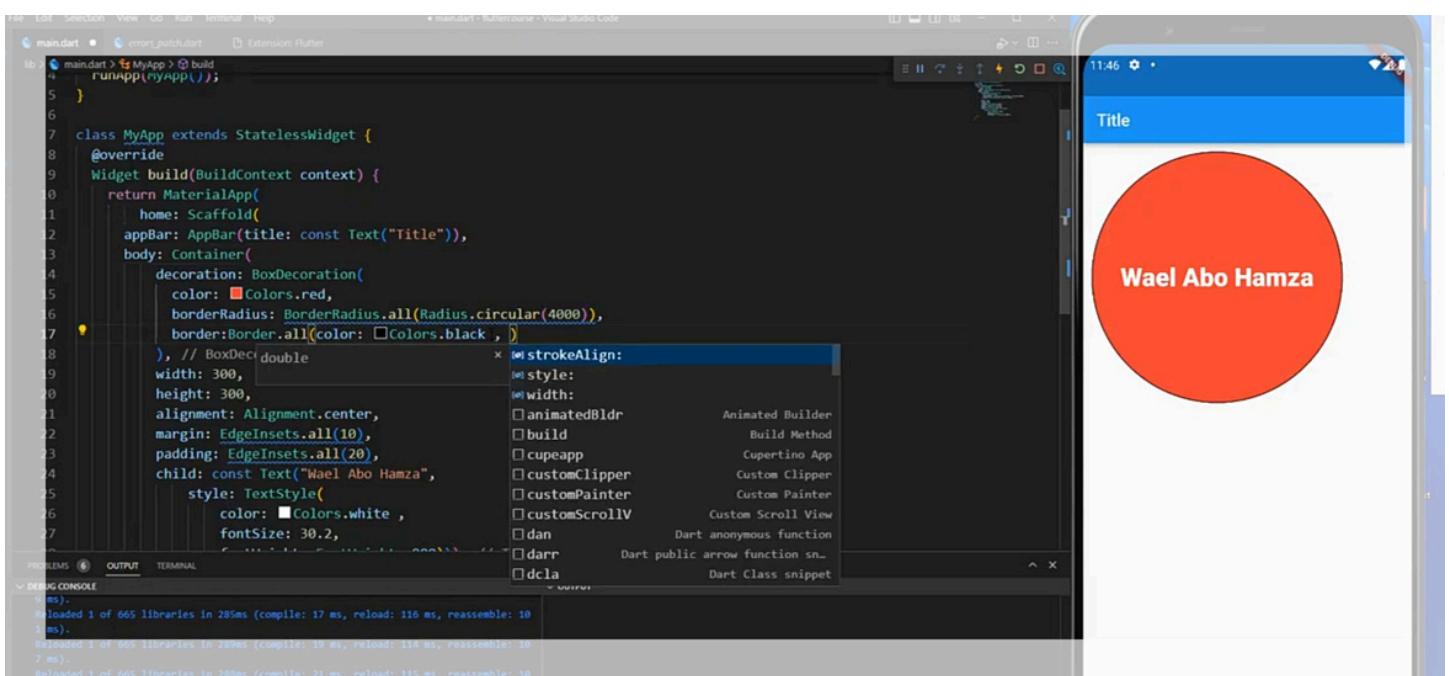
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: const Text("Title")),
        body: Container(
          decoration: BoxDecoration(
            color: Colors.red,
            borderRadius: BorderRadius.circular(200),
          ),
          width: 400,
          height: 400,
          alignment: Alignment.topRight,
          margin: EdgeInsets.all(10),
          padding: EdgeInsets.all(20),
          child: const Text("Wael Abo Hamza",
            style: TextStyle(
              color: Color.fromARGB(255, 91, 185, 96),
              fontSize: 40.2,
              fontWeight: FontWeight.w900)),
        ), // Scaffold // MaterialApp
      ),
    );
  }
}

```

الحدود نقوم بتنغيرها مثل هنا جعلناها دائيرية طبعا نستطيع التخفيض من حدود المربع



وجعلنا alignment center صار النص بالوسط من مربع دائرة من خلال الاعدادات على الحدود



هنا تم إضافة لون لحدود الدائرة وهو اللون الأسود ونستطيع التحكم بسمك الحد من width



A screenshot of an Android application running in an IDE. The code in the main.dart file defines a StatelessWidget named MyApp. It returns a MaterialApp with a Scaffold containing a Container. The Container has a red BoxDecoration with a circular border radius of 4000 and a black border width of 10. The text 'Wael Abo Hamza' is centered in the container with a font size of 30.2. The application's title is 'Title'. The bottom of the screen shows the Android navigation bar.

```
5 }  
6  
7 class MyApp extends StatelessWidget {  
8     @override  
9     Widget build(BuildContext context) {  
10         return MaterialApp(  
11             home: Scaffold(  
12                 appBar: AppBar(title: const Text("Title")),  
13                 body: Container(  
14                     decoration: BoxDecoration(  
15                         color: Colors.red,  
16                         borderRadius: BorderRadius.all(Radius.circular(4000)),  
17                         border: Border.all(color: Colors.black, width: 10)  
18                     ), // BoxDecoration  
19                     width: 300,  
20                     height: 300,  
21                     alignment: Alignment.center,  
22                     margin: EdgeInsets.all(10),  
23                     padding: EdgeInsets.all(20),  
24                     child: const Text("Wael Abo Hamza",  
25                         style: TextStyle(  
26                             color: Colors.white,  
27                             fontSize: 30.2,  
28                         ),  
29                     ),  
30                 ),  
31             ),  
32         );  
33     }  
34 }  
  
PROBLEMS OUTPUT TERMINAL
```

DEBUG CONSOLE

```
2 ms).  
Reloaded 1 of 665 libraries in 291ms (compile: 19 ms, reload: 115 ms, reassemble: 10  
4 ms).  
Reloaded 1 of 665 libraries in 308ms (compile: 18 ms, reload: 120 ms, reassemble: 11  
1 ms).  
Reloaded 1 of 665 libraries in 288ms (compile: 20 ms, reload: 116 ms, reassemble: 10  
3 ms).  
Reloaded 1 of 665 libraries in 292ms (compile: 18 ms, reload: 118 ms, reassemble: 10  
6 ms).  
Reloaded 1 of 665 libraries in 295ms (compile: 21 ms, reload: 119 ms, reassemble: 10  
6 ms).
```

Wael Abo Hamza



A screenshot of an Android application running in an IDE. The code in the main.dart file is identical to the previous one, except for the addition of a box shadow with a blue color and an offset of 10, 10. A tooltip for the Offset class is visible, stating: 'Creates an offset. The first argument sets [dx], the horizontal component, and the second sets [dy], the vertical component.' The application's title is 'Title'. The bottom of the screen shows the Android navigation bar.

```
5 }  
6  
7 class MyApp extends StatelessWidget {  
8     @override  
9     Widget build(BuildContext context) {  
10         return MaterialApp(  
11             home: Scaffold(  
12                 appBar: AppBar(title: const Text("Title")),  
13                 body: Container(  
14                     decoration: BoxDecoration(  
15                         color: Colors.red,  
16                         borderRadius: BorderRadius.all(Radius.circular(4000)),  
17                         border: Border.all(color: Colors.black, width: 10),  
18                         boxShadow: [  
19                             BoxShadow(color: Colors.blue, offset: Offset(10, 10))  
20                         ]  
21                     ), // BoxDecoration  
22                     width: 300,  
23                     height: 300,  
24                     alignment: Alignment.center,  
25                     margin: EdgeInsets.all(10),  
26                     padding: EdgeInsets.all(20),  
27                     child: const Text("Wael Abo Hamza",  
28                         style: TextStyle(  
29                             color: Colors.white,  
30                             fontSize: 30.2,  
31                         ),  
32                     ),  
33                 ),  
34             ),  
35         );  
36     }  
37 }  
  
PROBLEMS OUTPUT TERMINAL
```

DEBUG CONSOLE

```
1 ms).  
Reloaded 1 of 665 libraries in 288ms (compile: 20 ms, reload: 116 ms, reassemble: 10  
3 ms).  
Reloaded 1 of 665 libraries in 292ms (compile: 18 ms, reload: 118 ms, reassemble: 10  
6 ms).  
Reloaded 1 of 665 libraries in 295ms (compile: 21 ms, reload: 119 ms, reassemble: 10  
6 ms).  
Reloaded 1 of 665 libraries in 298ms (compile: 19 ms, reload: 117 ms, reassemble: 10  
2 ms).  
Reloaded 1 of 665 libraries in 300ms (compile: 18 ms, reload: 119 ms, reassemble: 10  
8 ms).
```

Wael Abo Hamza

وهنا لاحظنا ضل الدائرة من Boxshadow لون أزرق وحددنا حجمه

الإزاحة offset تم تحديد القيمة بـ (10, 10) هذا يعني أن الظل تحرك بمقدار 10 بكسل جهة اليمين و10 بكسل للأسفل بالنسبة لموقع الدائرة الأصلي

The screenshot shows the Android Studio interface. On the left is the code editor with Dart code for a StatelessWidget named MyApp. The code defines a Scaffold with a red AppBar titled "Title". The body is a Container with a circular BoxDecoration, a blue BoxShadow with a spreadRadius of 10, and a blurRadius of 20. The Container has a width and height of 300, centered alignment, and padding of 20. The text "Wael Abo Hamza" is its child. Below the code editor are the DEBUG CONSOLE and OUTPUT tabs, both showing logs of library reloads. On the right is a preview of the app running on an Android device, showing a red circle with the text "Wael Abo Hamza" inside it, with a blue shadow effect.

```
5 }
6
7 class MyApp extends StatelessWidget {
8   @override
9   Widget build(BuildContext context) {
10     return MaterialApp(
11       home: Scaffold(
12         appBar: AppBar(title: const Text("Title")),
13         body: Container(
14           decoration: BoxDecoration(
15             color: Colors.red,
16             borderRadius: BorderRadius.all(Radius.circular(4000)),
17             border: Border.all(color: Colors.black, width: 10),
18             boxShadow: [
19               BoxShadow(color: Colors.blue, offset: Offset(10, 10), spreadRadius: 10, blurRadius: 20)
20             ]
21           ), // BoxDecoration
22           width: 300,
23           height: 300,
24           alignment: Alignment.center,
25           margin: EdgeInsets.all(10),
26           padding: EdgeInsets.all(20),
27           child: const Text("Wael Abo Hamza"),
28         ),
29       ),
30     );
31   }
32 }
```

نطاق الانتشارspreadRadius هي 10 وهي المسؤولة عن حجم الظل ومدى توسيعه خلف الدائرة قبل أن يبدأ بالتمويه

(blurRadius) القيمة هي 20 وهي التي تعطي التأثير الضبابي الناعم للظل بدلاً من أن يكون حاداً

وهنا غيرنا بالspreadRadius وخليناها 1

وخليناها 10 وهيك صار معنا blurRadius

The screenshot shows the Android Studio interface. The code is identical to the previous one, except the spreadRadius value in the BoxShadow is changed from 10 to 1. The preview on the right shows the same red circle with the text "Wael Abo Hamza", but the shadow effect is much smaller and less prominent than before.

```
5 }
6
7 class MyApp extends StatelessWidget {
8   @override
9   Widget build(BuildContext context) {
10     return MaterialApp(
11       home: Scaffold(
12         appBar: AppBar(title: const Text("Title")),
13         body: Container(
14           decoration: BoxDecoration(
15             color: Colors.red,
16             borderRadius: BorderRadius.all(Radius.circular(4000)),
17             border: Border.all(color: Colors.black, width: 10),
18             boxShadow: [
19               BoxShadow(color: Colors.blue, offset: Offset(1, 1), spreadRadius: 1, blurRadius: 10)
20             ]
21           ), // BoxDecoration
22           width: 300,
23           height: 300,
24           alignment: Alignment.center,
25           margin: EdgeInsets.all(10),
26           padding: EdgeInsets.all(20),
27           child: const Text("Wael Abo Hamza"),
28         ),
29       ),
30     );
31   }
32 }
```

هنا إضافة صورة لتطبيق من الإنترنت ونستدعي رابط الصورة من خلال network

The screenshot shows the Visual Studio Code interface. On the left is the Explorer sidebar with project files like .idea, android, build, ios, and lib. The lib folder contains main.dart, .gitignore, .metadata, analysis_options.yaml, fluttercourse.ml, pubspec.lock, pubspec.yaml, and README.md. The main.dart file is open in the editor, displaying Dart code that loads an image from a URL. The terminal at the bottom shows the command `flutter pub get` being run, with the output indicating dependencies are being resolved.

```
1 import 'package:flutter/material.dart';
2
3 Run | Debug | Profile
4 void main() {
5   runApp(MyApp());
6 }
7
8 class MyApp extends StatelessWidget {
9   @override
10  Widget build(BuildContext context) {
11    return MaterialApp(
12      home: Scaffold(
13        appBar: AppBar(title: const Text("Title")),
14        body: Container(
15          child: Image.network("https://cdn.xxl.thumbs.canstockphoto.com/green-nature-landscape-stock-photo_csp6371852.jpg"),
16        ), // Container
17      ); // Scaffold // MaterialApp
18 }
19
```

```
Deleting .dart_tool...
Deleting Generated.xconfig...
Deleting flutter_export_environment.sh...
PS C:\Users\wael\AndroidStudioProjects\fluttercourse> flutter pub get
Resolving dependencies...
collection 1.17.1 (1.17.2 available)
matcher 0.12.15 (0.12.16 available)
material_color_utilities 0.2.0 (0.5.0 available)
source_span 1.9.1 (1.10.0 available)
test_api 0.5.1 (0.6.0 available)
```

Wael Abo Hamza

وإذا كانت الصورة في الملف images عندي نكتب بدلًا

This screenshot shows the same VS Code setup as the first one, but the code in main.dart has been modified. Instead of using `Image.network`, it uses `Image.asset` to load an image from the images directory. The terminal output remains the same, showing the successful execution of `flutter pub get`.

```
1 import 'package:flutter/material.dart';
2
3 Run | Debug | Profile
4 void main() {
5   runApp(MyApp());
6 }
7
8 class MyApp extends StatelessWidget {
9   @override
10  Widget build(BuildContext context) {
11    return MaterialApp(
12      home: Scaffold(
13        appBar: AppBar(title: const Text("Title")),
14        body: Container(
15          child: Image.asset("images/wwwwww.jpg"),
16        ), // Container
17      ); // Scaffold // MaterialApp
18 }
19
```

```
Deleting .dart_tool...
Deleting Generated.xconfig...
Deleting flutter_export_environment.sh...
PS C:\Users\wael\AndroidStudioProjects\fluttercourse> flutter pub get
Resolving dependencies...
collection 1.17.1 (1.17.2 available)
matcher 0.12.15 (0.12.16 available)
material_color_utilities 0.2.0 (0.5.0 available)
source_span 1.9.1 (1.10.0 available)
test_api 0.5.1 (0.6.0 available)
Got dependencies!
PS C:\Users\wael\AndroidStudioProjects\fluttercourse>
```

Wael Abo Hamza

```

 53 # The following section is specific to Flutter packages.
 54 flutter:
 55
 56   # The following line ensures that the Material Icons font is
 57   # included with your application, so that you can use the icons in
 58   # the material icons class.
 59   uses-material-design: true
 60
 61   # To add assets to your application, add an assets section, like this:
 62   assets:
 63     - images/wwwwww.jpg
 64     # - images/a_dot_ham.jpeg
 65
 66   # An image asset can refer to one or more resolution-specific "variants", see
 67   # https://flutter.dev/assets-and-images/#resolution-aware
 68
 69   # For details regarding adding assets from package dependencies, see
 70   # https://flutter.dev/assets-and-images/#from-packages
 71
 72   # To add custom fonts to your application, add a fonts section here,
 73   # in this "flutter" section. Each entry in this list should have a
 74   # "family" key with the font family name, and a "fonts" key with a
 75   # list giving the asset and other descriptors for the font. For
 76   # example:
 77
 78 
```

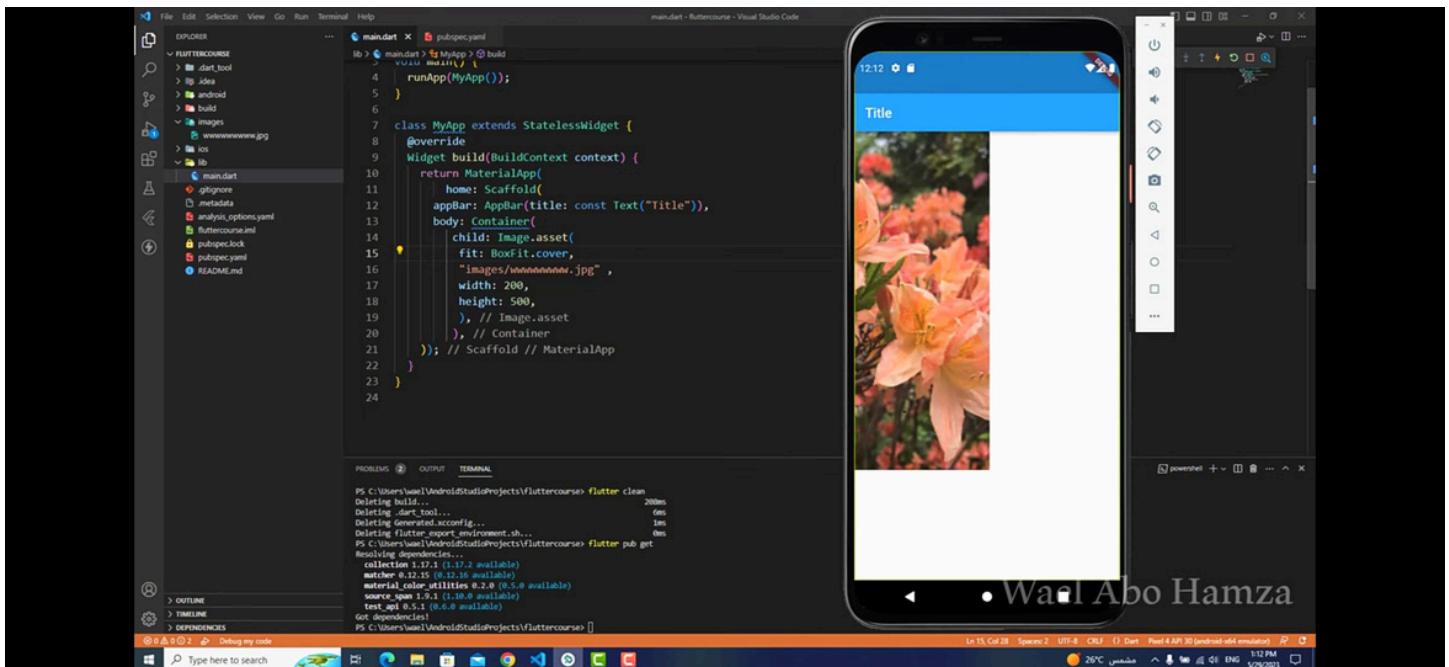
Wael Abo Hamza

وهنا كتبنا إسم الصورة في ملف pubspec.lock بالبرنامج ويوجب الإنتباه لأنه ملف حساس

```

 1 lib > main.dart: void main() {
 2   runApp(MyApp());
 3 }
 4
 5
 6 class MyApp extends StatelessWidget {
 7   @override
 8   Widget build(BuildContext context) {
 9     return MaterialApp(
10       home: Scaffold(
11         appBar: AppBar(title: const Text("Title")),
12         body: Container(
13           child: Image.asset(
14             "images/wwwwww.jpg",
15             fit: BoxFit.fill,
16             width: 200,
17             height: 500,
18           ), // Image.asset
19         ), // Container
20       ); // Scaffold // MaterialApp
21     }
22   }
23 }
24 
```

Wael Abo Hamza

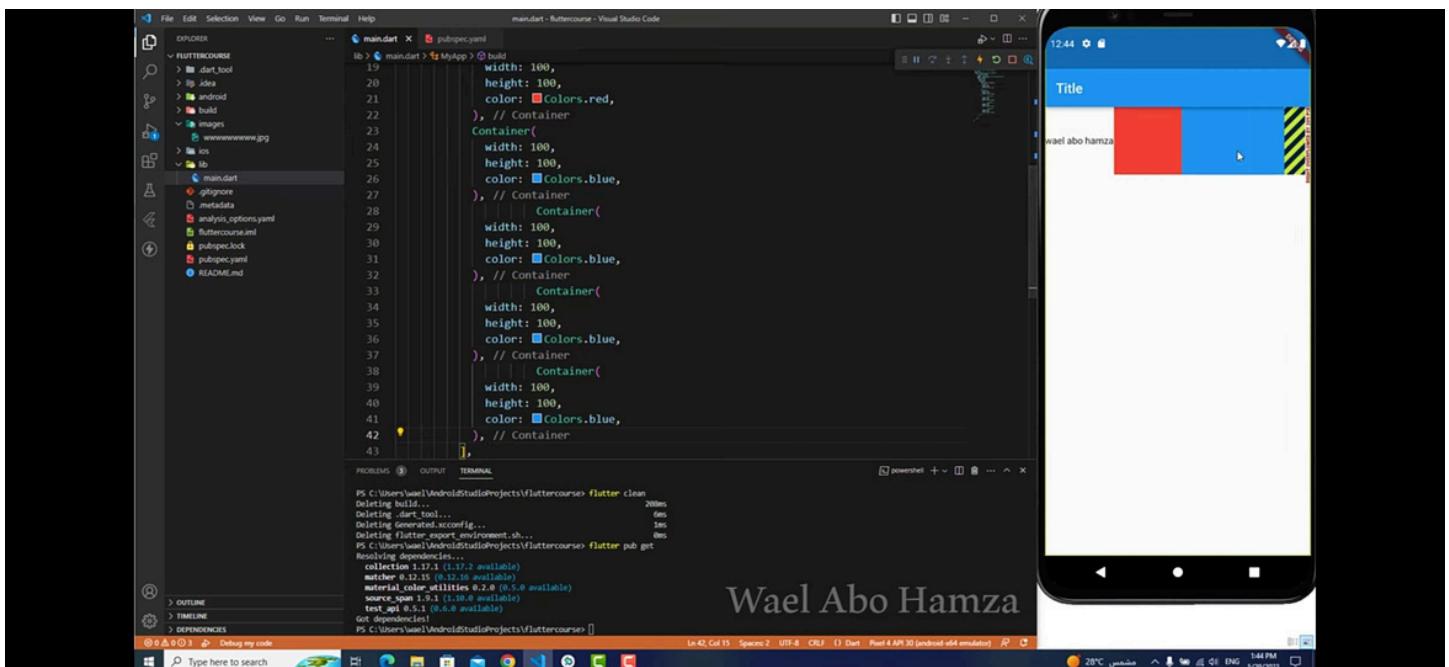


: يقوم بتمثيل الصورة لتملأ المساحة المحددة (العرض والطول) بالكامل، وهذا قد يؤدي إلى تغيير أبعاد الصورة الأصلية (distortion) فتظهر الصورة مضغوطة أو ممددة

: تجعل الصورة تغطي المساحة المحددة بالكامل مع الحفاظ على أبعادها، ولكنها ستقوم بقص (crop)

تحديد الطول والعرض: يتم ذلك عبر صفتين width و height داخل الـ Image.asset

Row . Column



Aligning widgets

You control how a row or column aligns its children using the `mainAxisAlignment` and `crossAxisAlignment` properties. For a row, the main axis runs horizontally and the cross axis runs vertically. For a column, the main axis runs vertically and the cross axis runs horizontally.

The `MainAxisAlignment` and `CrossAxisAlignment` enums offer a variety of constants for controlling alignment.

manard - Fluttercourse - Visual Studio Code

```

EXPLORER
main.dart
lib > main.dart > pubspec.yaml
  14   width: 500,
  15   child: Row(
  16     mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  17     children: [
  18       Text("wael"),
  19       Container(
  20         width: 100,
  21         height: 100,
  22         color: Colors.red,
  23       ), // Container
  24       Container(
  25         width: 100,
  26         height: 100,
  27         color: Colors.blue,
  28       ), // Container
  29       Container(
  30         width: 100,
  31         height: 100,
  32         color: Colors.green,
  33       ), // Container
  34     ], // Row
  35   ), // Container
  36 ); // Scaffold // MaterialApp
  37 }
  38

```

PROBLEMS OUTPUT TERMINAL

Syncing files to device sdk gphone xhdpi 64...

Wael Abo Hamza

12:44 28°C 144 PM 5/29/2023

manard - Fluttercourse - Visual Studio Code

```

EXPLORER
main.dart
lib > main.dart > pubspec.yaml
  14   width: 500,
  15   child: Row(
  16     mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  17     children: [
  18       Text("wael"),
  19       Container(
  20         width: 100,
  21         height: 100,
  22         color: Colors.red,
  23       ), // Container
  24       Container(
  25         width: 100,
  26         height: 100,
  27         color: Colors.blue,
  28       ), // Container
  29       Container(
  30         width: 100,
  31         height: 100,
  32         color: Colors.green,
  33       ), // Container
  34     ], // Row
  35   ), // Container
  36 ); // Scaffold // MaterialApp
  37 }
  38

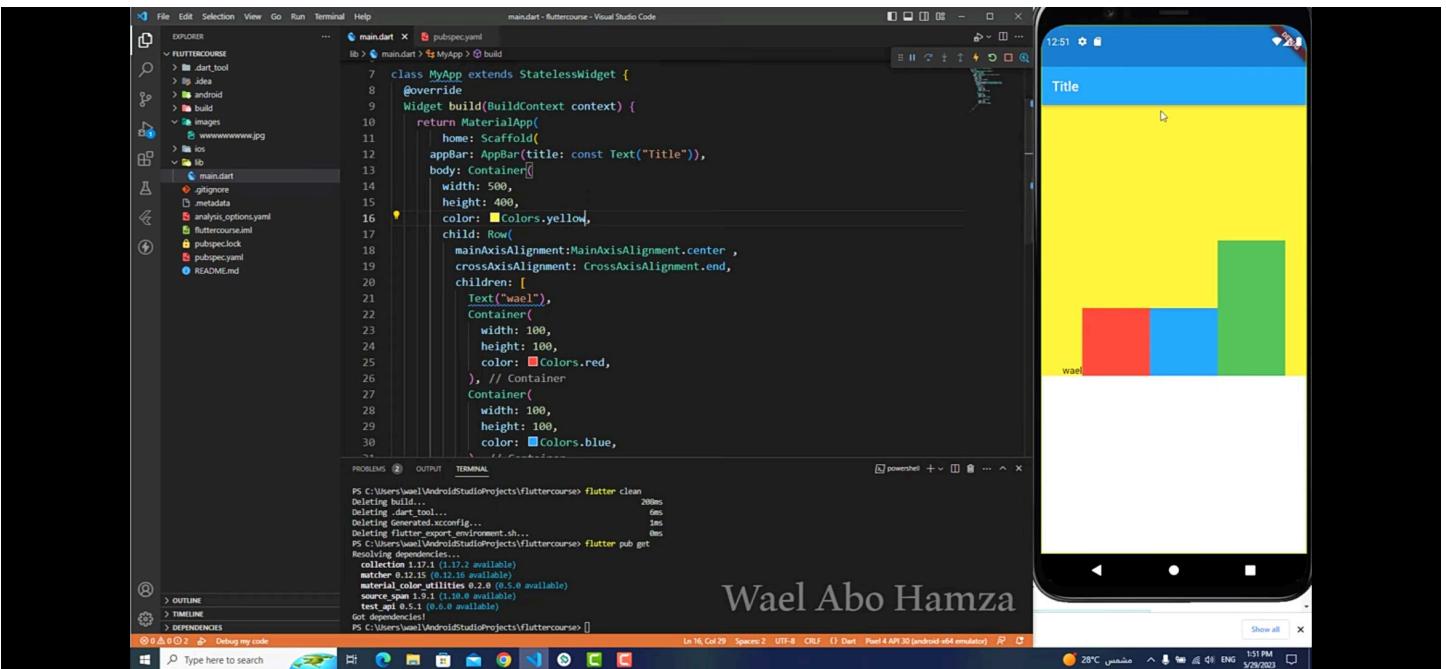
```

PROBLEMS OUTPUT TERMINAL

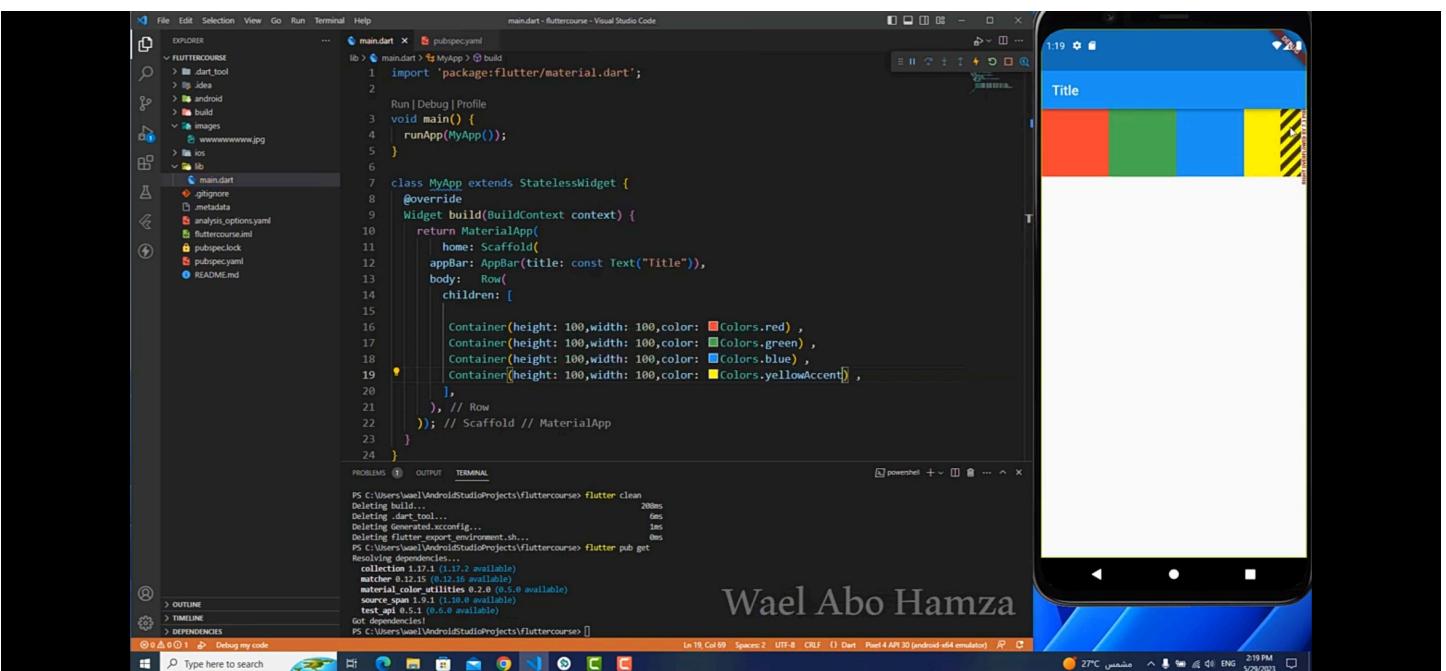
Syncing files to device sdk gphone xhdpi 64...

Wael Abo Hamza

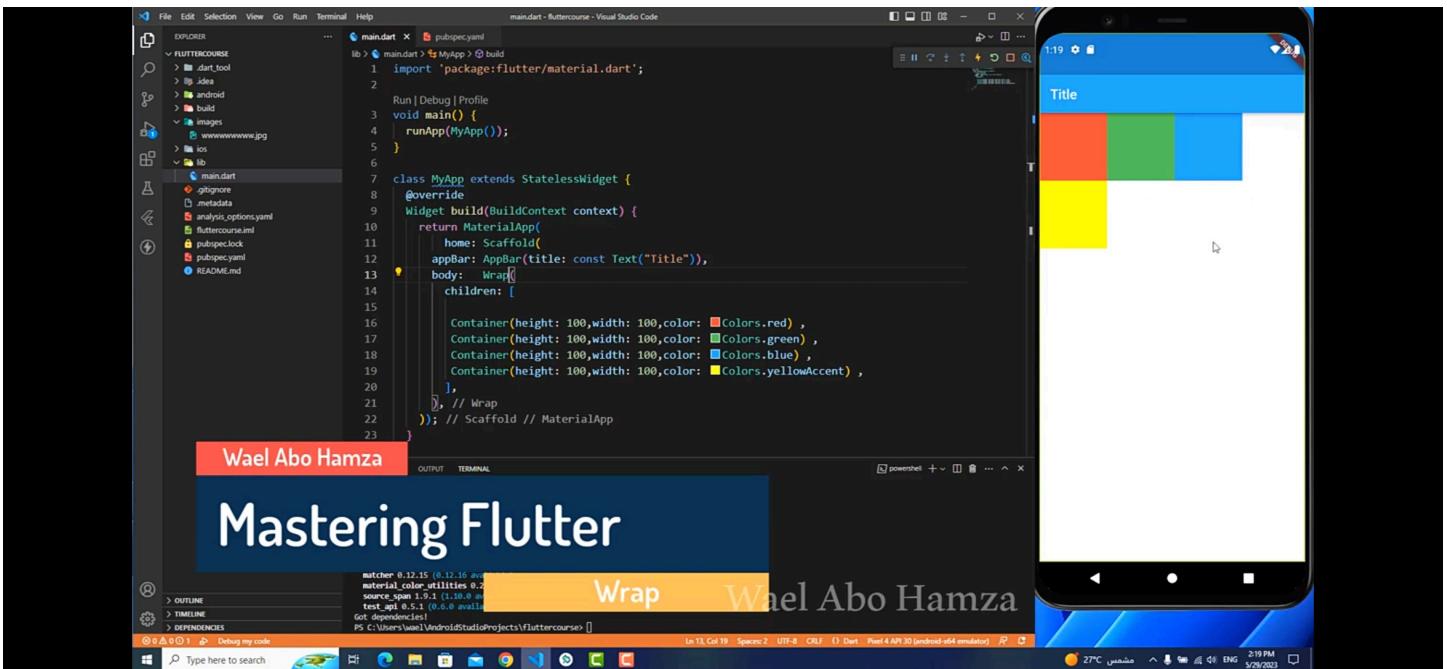
12:45 28°C 145 PM 5/29/2023



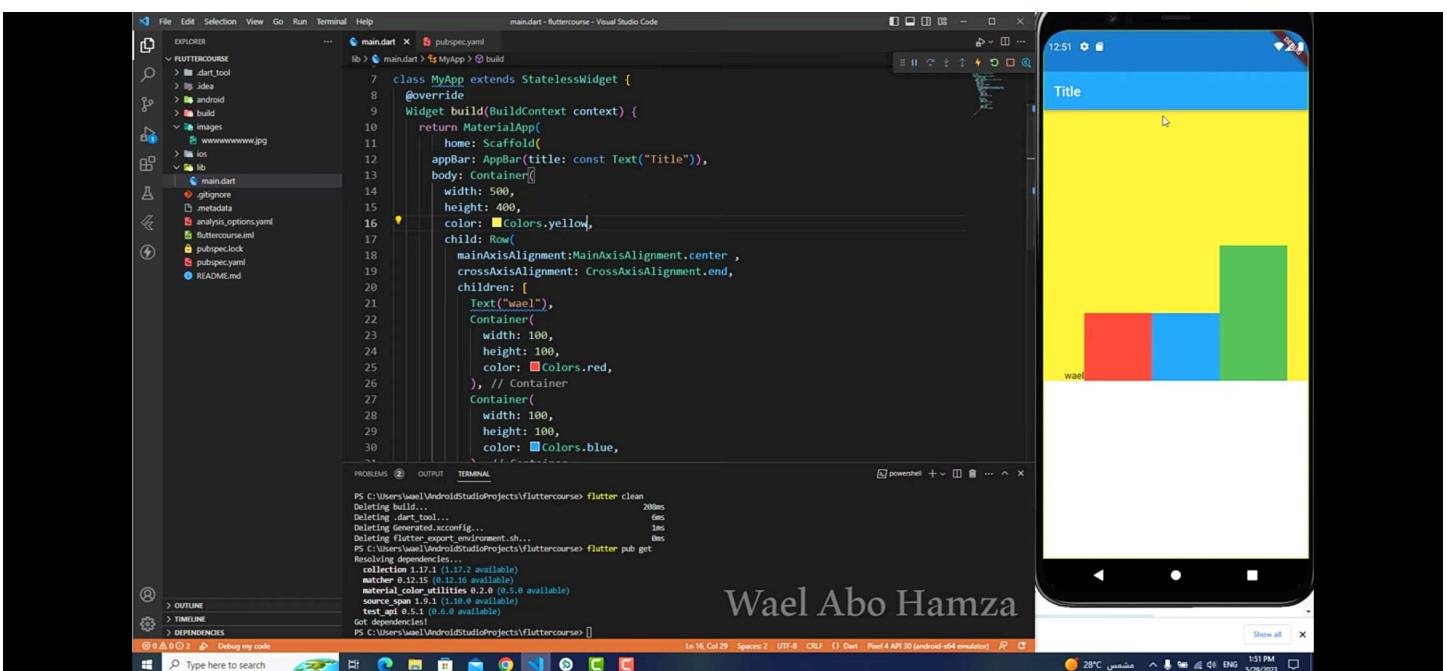
تتمحور فكرة محاذاة العناصر في Flutter حول المحاور الأساسية والفرعية لكل من Row و Column؛ حيث يتحكم mainAxisAlignment في توزيع العناصر على المحور الطولي للمكون، بينما يتحكم crossAxisAlignment في محاذاتها على المحور العرضي. ففي حالة الصف Row يكون المحور الأساسي أفقياً والمحور العرضي رأسياً، والعكس تماماً في العمود Column. تتيح لك هذه الخصائص وضع العناصر في البداية أو الوسط أو النهاية، مما يساعد في تنسيق الواجهة بدقة ومنع حدوث مشاكل مثل تجاوز العناصر لمساحة الشاشة المتاحة التي تظهر خطوط صفراء وسوداء تحذيرية. عند وضع هذه العناصر داخل حاوية Container ذات أبعاد محددة ولون خلفية معين، يظهر تأثير هذه المحاذات بشكل أوضح داخل المساحة المتاحة ولا نستطيع فهمها مية بالمية إلا بالتطبيق العملي



هنا ظهرت مشكلة لما أضفنا containers لأنها أكبر من عرض الشاشة ظهرت خطوط صفراء



لحل هذه المشكلة بشكل ذكي يستخدم Wrap حيث يقوم Wrap بدلًا من Row حيث يوحيت Wrap تلقائيًا بنقل العناصر التي لا تجد مساحة كافية إلى سطر جديد بدلاً من قصها أو إظهار خطأ



The screenshot shows a Visual Studio Code interface with an open Flutter project named 'FLUTTERCOURSE'. The Explorer sidebar on the left displays the project structure, including files like main.dart, pubspec.yaml, and various assets. The main editor window contains the main.dart file, which defines a StatelessWidget named MyApp. This widget's build method creates a Scaffold with a title bar and a body consisting of a Row of four Container widgets, each with a height of 100 and a width of 100, colored red, green, blue, and yellow respectively. A yellow dot marker is placed at the end of the first Container's definition. Below the editor is a terminal window showing the command flutter clean being run, followed by the output of dependency resolution. To the right of the code editor is a mobile device icon representing an Android emulator, displaying a screen with four colored boxes (red, green, blue, yellow) arranged horizontally. The bottom of the screen shows the Windows taskbar with various pinned icons.

```
main.dart - fluttercourse - Visual Studio Code

File Edit Selection View Go Run Terminal Help
EXPLORER
FLUTTERCOURSE
  - dart_tool
  - idea
  - android
  - build
  - images
    - wwwwwwwww.jpg
  - ios
  - lib
    - main.dart
    - .gitignore
    - .metadata
    - analysis_options.yaml
    - fluttercourse.iml
    - pubspec.lock
    - pubspec.yaml
  - README.md

main.dart x pubspec.yaml
lib > main.dart > MyApp > build
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   @override
9   Widget build(BuildContext context) {
10     return MaterialApp(
11       home: Scaffold(
12         appBar: AppBar(title: const Text("Title")),
13         body: Row(
14           children: [
15             Container(height: 100, width: 100, color: Colors.red),
16             Container(height: 100, width: 100, color: Colors.green),
17             Container(height: 100, width: 100, color: Colors.blue),
18             Container(height: 100, width: 100, color: Colors.yellowAccent),
19           ],
20         ), // Row
21     ); // Scaffold // MaterialApp
22   }
23 }
24 }

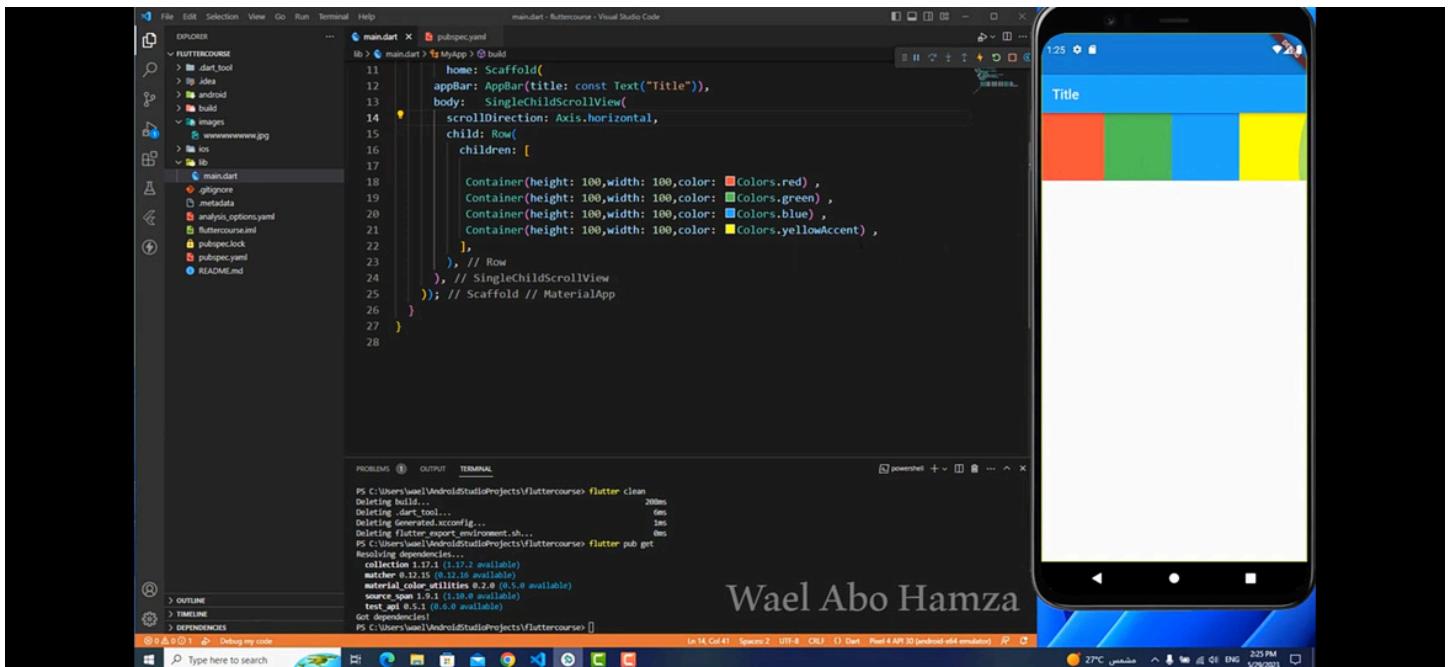
PROBLEMS OUTPUT TERMINAL
powershell + v ☰ ... ^ x
PS C:\Users\Wael\AndroidStudioProjects\fluttercourse> flutter clean
Deleting build...
Deleting dart_tool...
Deleting Generated.xconfig...
Deleting flutter_export_environment.sh...
ps C:\Users\Wael\AndroidStudioProjects\fluttercourse> flutter pub get
resolving dependencies...
collection 1.17.1 (1.17.2 available)
  matcher 0.12.15 (0.12.16 available)
  material 2.3.4+1 (2.3.5+1 available)
  seek_bar 1.0.1 (1.0.2 available)
  test_api 0.5.1+1 (0.6.0 available)
Got dependencies!
PS C:\Users\Wael\AndroidStudioProjects\fluttercourse>

Wael Abo Hamza
```

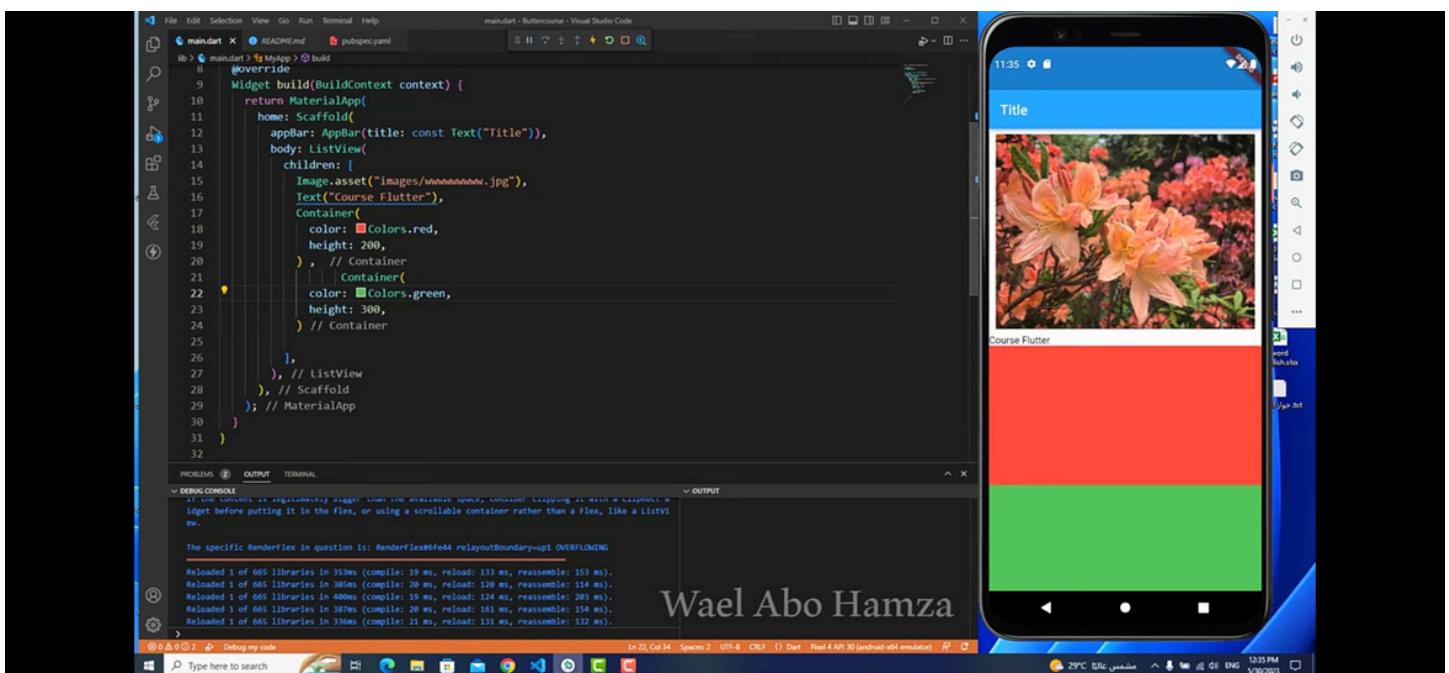
The screenshot shows the Visual Studio Code interface with a Flutter project open. The left sidebar displays the file structure of the 'fluttercourse' directory, including files like main.dart, pubspec.yaml, and analysis_options.yaml. The main editor window contains the main.dart code, which defines a StatelessWidget named MyApp. This widget builds a Scaffold with a title and a body containing a Wrap widget that arranges four Container widgets of different colors (red, green, blue, yellow) in a grid. The bottom right corner of the screen shows an Android emulator displaying the resulting app with the four colored boxes. The status bar at the bottom indicates the code is running on an API 30 emulator.

```
import 'package:flutter/material.dart';
void main() {
  runApp(MyApp());
}
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: const Text("Title")),
        body: Wrap(
          children: [
            Container(height: 100,width: 100,color: Colors.red),
            Container(height: 100,width: 100,color: Colors.green),
            Container(height: 100,width: 100,color: Colors.blue),
            Container(height: 100,width: 100,color: Colors.yellowAccent),
          ],
        ), // Wrap
      ); // Scaffold
  }
}
```

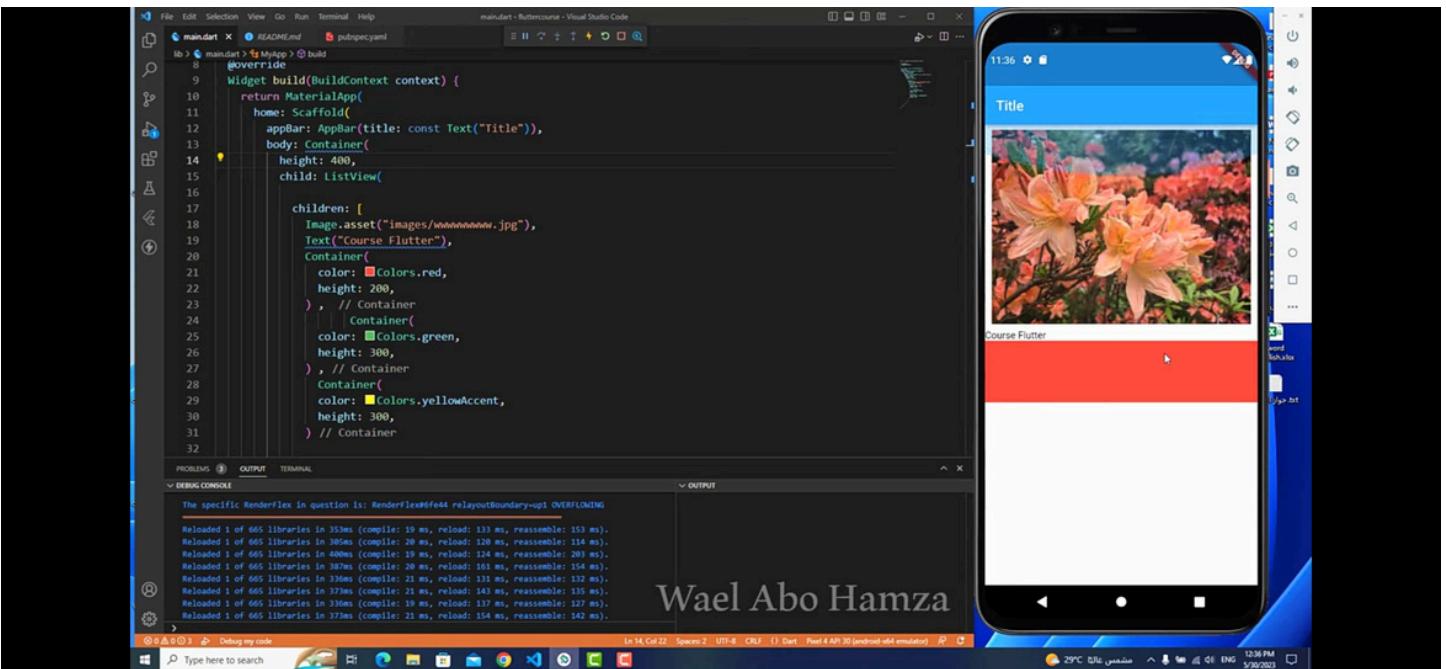
وعندما تزيد العناصر عن مساحة الشاشة المتوفرة يظهر خطأ تجاوز المساحة Overflow لحل هذه المشكلة استخدام `Wrap` الذي يقوم بنقل العناصر الزائدة تلقائياً إلى سطر جديد أو استخدام `SingleChildScrollView` الذي يتيح للمستخدم التمرير (Scroll) للوصول إلى المحتوى المخفى



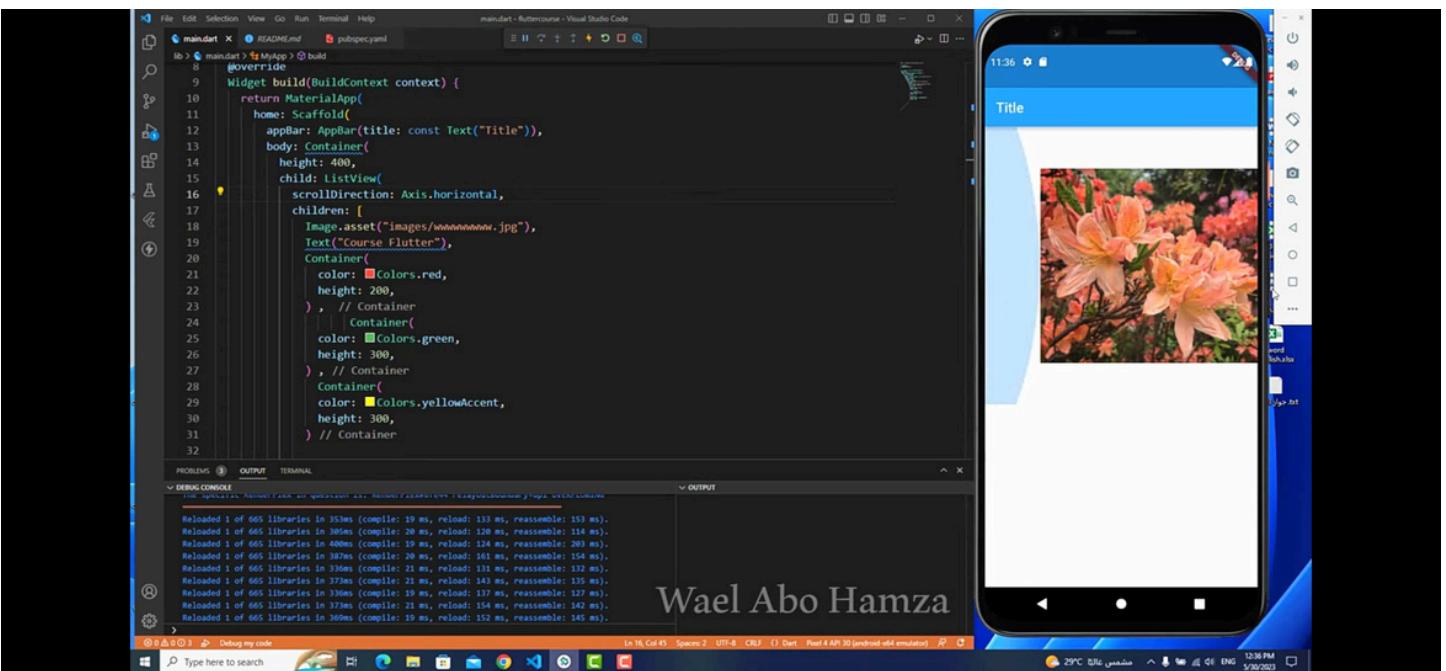
مشكلة تجاوز المساحة (Overflow) كخطوط تحذيرية ملونة عندما تزيد أبعاد العناصر عن حجم الشاشة لحل هذه المشكلة يمكن استخدام ويدجت Wrap الذي ينقل العناصر تلقائياً لسطر جديد أو استخدام SingleChildScrollView الذي يضيف خاصية التمرير (Scrolling) للعناصر سواء كان التمرير رأسياً داخل العمود أو أفقياً داخل الصف عبر scrollDirection تحديد



تعرض الحالة الصحيحة حيث تم وضع العناصر داخل ListView مباشرة مما يسمح بالتمرير (Scroll) وعرض كافة المحتويات الصورة، النص، والألوان دون مشاكل



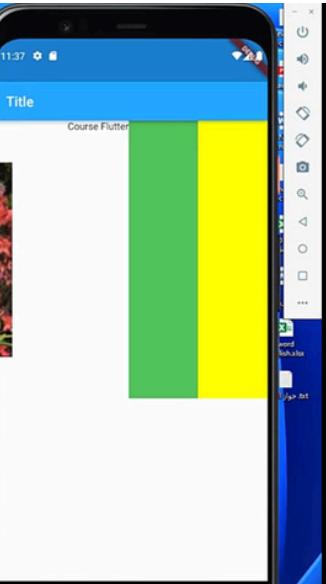
تعرض خطأ برمجي حيث تم تقييد الـ `ListView` داخل `Container` بارتفاع ثابت 400 بكسل لأن محتوى القائمة أطول من هذا الارتفاع، ظهر تحذير الخطوط الصفراء والسوداء (Overflow) في أسفل الشاشة



المشكل. عند وضع `ListView` داخل `Container` له طول محدد (مثل `height: 400`) وكان محتوى القائمة أطول من هذه المساحة، يظهر خطأ overflow (الخطوط الصفراء والسوداء)، لأن العناصر "تفيض" خارج حدود الحاوية.

الحل: قام المبرمج بتغيير اتجاه التمرير إلى أفقي `scrollDirection: Axis.horizontal`. في هذه الحالة، اختفى خطأ الطول لأن القائمة أصبحت تتمدد بالعرض، لكن التصميم تغير ليناسب التمرير من اليمين لليسار

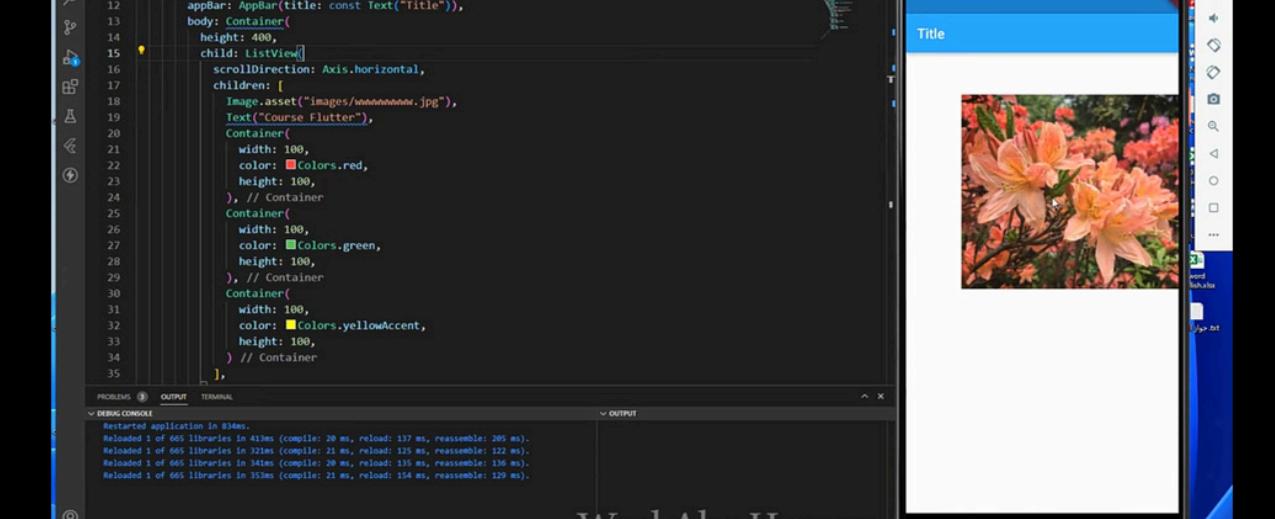
الدرس يشرح أن `ListView` يحتاج مساحة مرنة ليؤدي وظيفته، ووضع قيود ثابتة عليه يؤدي لظهور أخطاء في الواجهة الملخص بإختصار: الدرس يشرح أن `ListView` يحتاج مساحة مرنة ليؤدي وظيفته، ووضع قيود ثابتة عليه يؤدي لظهور أخطاء في الواجهة



The screenshot shows a Flutter application running on an Android emulator. The app's title is "Title" and the content is "Course Flutter". On the left, the Visual Studio Code editor displays the main.dart file with code for a Container widget:

```
body: Container(
  height: 400,
  child: ListView(
    scrollDirection: Axis.horizontal,
    children: [
      Image.asset("images/wwwwww.jpg"),
      Text("Course Flutter"),
      Container(
        alignment: AlignmentGeometry? // Hovered over here
        child: Text("Flutter"),
        constraints: BoxConstraints.expand(),
        decoration: BoxDecoration(color: Colors.yellowAccent),
        height: 100,
      ) // Container
    ],
  ),
)
```

A tooltip for the `Alignment` field is open, providing documentation: "Align the [child] within the container. If non-null, the container will expand to fill its parent and position its child within itself according to the given value. If the incoming constraints are unbounded, then the child will be shrink-wrapped instead. Ignored if [child] is null." It also lists "See also:" items: "[Alignment]" and "[AlignmentDirectional]".



The screenshot shows a Flutter application running on an Android emulator. The app has a blue header bar with the title "Title". The main content area displays a grid of four flower images. Each flower is surrounded by a red border. The images are as follows:

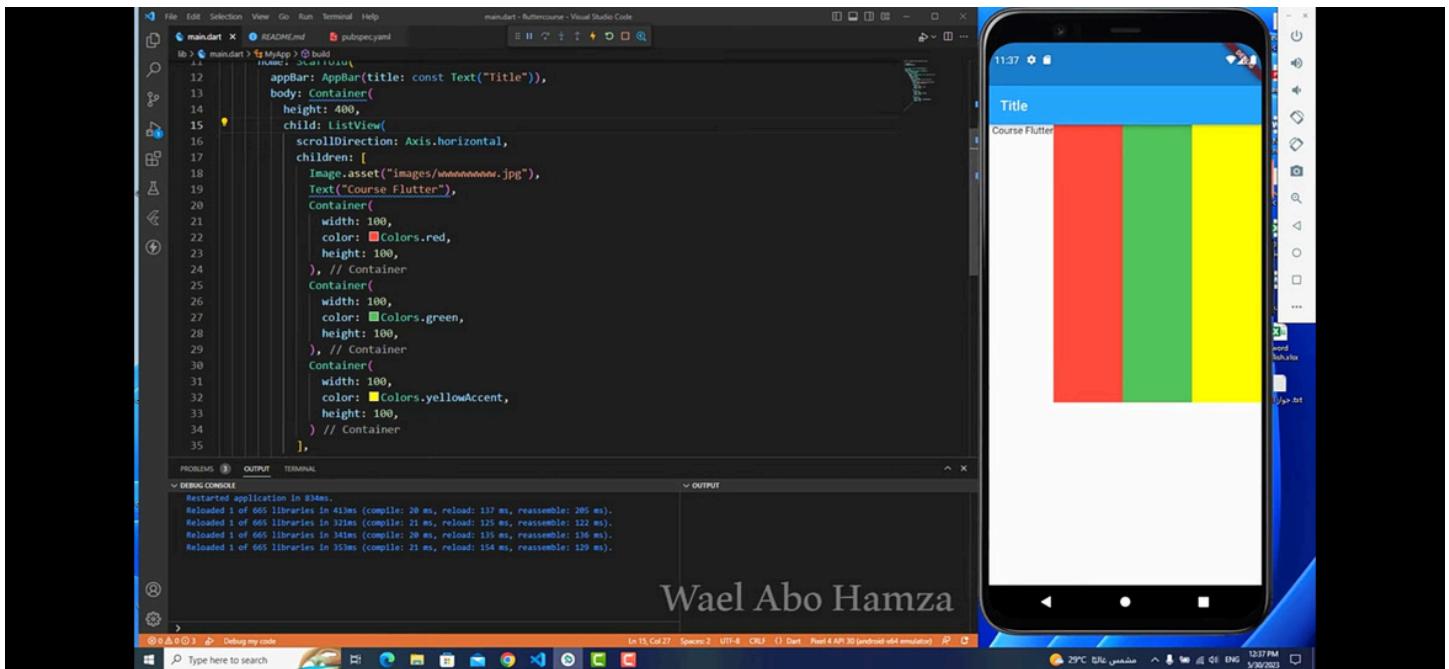
- Top-left: A pink flower.
- Top-right: An orange flower.
- Bottom-left: A yellow flower.
- Bottom-right: A pink flower.

The code editor on the left shows the Dart code for the application's main screen. The code defines a horizontal scroll view containing four containers, each with a different color and width/height of 100 units. The colors used are red, green, yellow, and red again. The background of the scroll view is white.

```
12 appBar: AppBar(title: const Text("Title")),
13 body: Container(
14   height: 400,
15   child: ListView(
16     scrollDirection: Axis.horizontal,
17     children: [
18       Image.asset("images/wwwwww.jpg"),
19       Text("Course Flutter"),
20       Container(
21         width: 100,
22         color: Colors.red,
23         height: 100,
24       ), // Container
25       Container(
26         width: 100,
27         color: Colors.green,
28         height: 100,
29       ), // Container
30       Container(
31         width: 100,
32         color: Colors.yellowAccent,
33         height: 100,
34     ), // Container
35   ],
36 ),
37 )
38 
```

The terminal at the bottom shows the application's log output, indicating it has restarted and reloaded multiple times.

```
Restarted application in 83ms.
Reloaded 1 of 665 libraries in 413ms (compile: 20 ms, reload: 137 ms, reassemble: 205 ms).
Reloaded 1 of 665 libraries in 321ms (compile: 21 ms, reload: 125 ms, reassemble: 122 ms).
Reloaded 1 of 665 libraries in 341ms (compile: 20 ms, reload: 135 ms, reassemble: 136 ms).
Reloaded 1 of 665 libraries in 353ms (compile: 21 ms, reload: 154 ms, reassemble: 129 ms).
```

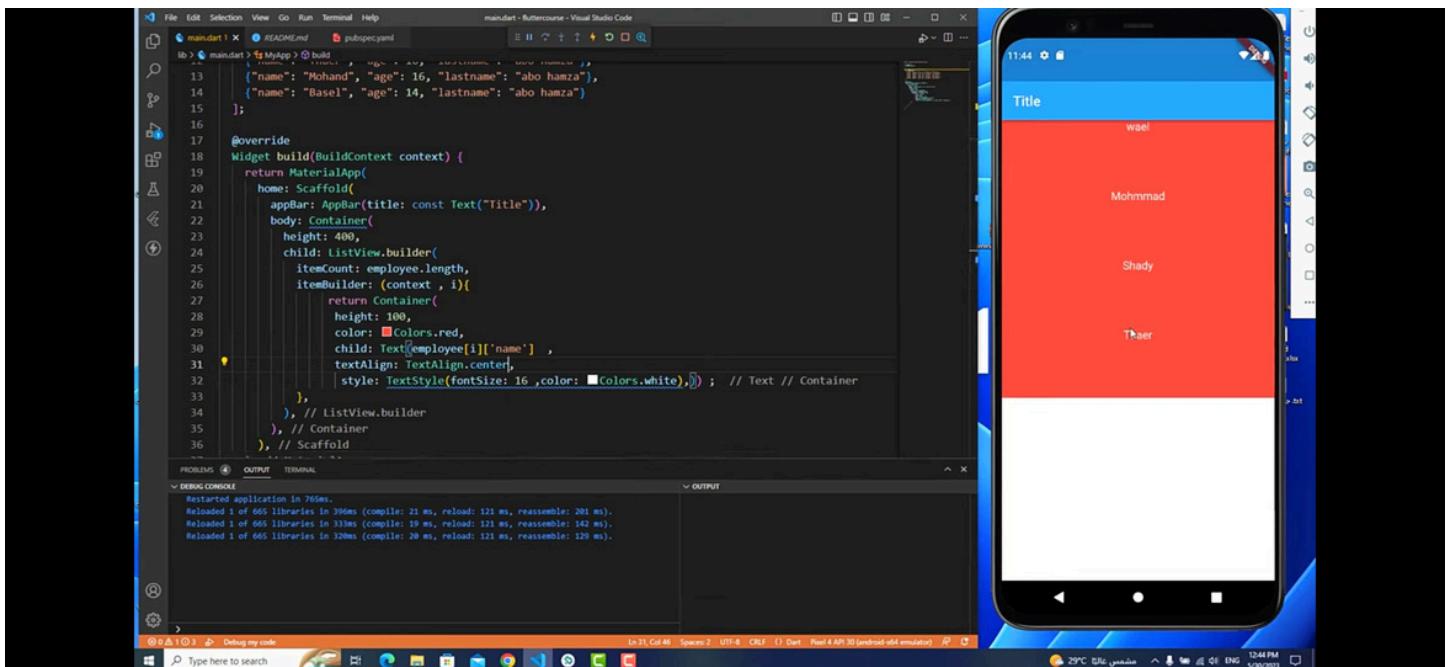


وضع ListView داخل Container بارتفاع ثابت وصغير يؤدي لخطأ (الخطوط الصفراء) لأن العناصر تفيف خارج حدود الحاوية.

الحل الأول تغيير الإتجاه تحويل التمرير إلى أفقي (Axis.horizontal) يزيل خطأ الارتفاع لأن القائمة أصبحت تمدد بالعرض.

تنسيق العناصر في الوضع الأفقي، يجب تحديد عرض width لكل عنصر (مثلاً width: 100) لضمان ظهور العناصر بجانب بعضها بشكل منظم ومنع تداخلها.

الملخص ListView يحتاج مساحة مرنّة، وإذا تم تقييده بمساحة ثابتة يجب التأكد من أن محتواه يناسب هذه المساحة أو تغيير اتجاه التمرير



المشكلة Overflow وضع ListView داخل Container بارتفاع ثابت يسبب خطأ "تجاوز المساحة" (الخطوط الصفراء) لأن العناصر أطول من الحاوية

الحل الأول (اتجاه التمرير): تغيير التمرير ليكون أفقياً (Axis.horizontal) يزيل الخطأ لأن القائمة أصبحت تمدد بالعرض وليس الطول

التنسيق الأفقي في الوضع الأفقي يجب منح العناصر عرضاً ثابتاً width لتظهر بجانب بعضها بشكل منظم

التطبيق ListView.builder بدلًا من القائمة العادية هو الطريقة الاحترافية لعرض بيانات من قائمة (List/Array) بشكل ديناميكي وتلقائي بناء على حجم البيانات

باختصار الدرس يعلمك كيفية إدارة المساحات والفرق بين التمرير الرأسى والأفقي وكيفية بناء قوائم تعتمد على البيانات Dynamic Lists

ListView separated

يُستخدم لعرض قائمة من ListView. separated مخصوص في إطار العمل (Constructor) "هو" منشئ (Constructor) العادي بأنه يضيف عنصر فاصل Separator للعناصر القابلة للتغير، ولكنه يتميز عن الفرق بين كل عنصر والأخر.

هذا الخيار مثالى جداً عندما تريد وضع خطوط أفقية، مسافات، أو حتى أيقونات تفصل بين عناصر القائمة بشكل منظم.

المكونات الأساسية (Properties)

بنجاح، تحتاج لتحديد ثلاثة متغيرات أساسية ListView.separated لعمل:

item bulider: هو المسؤول عن بناء شكل "العنصر" الأساسي (مثل نص أو بطاقة).

seprator bulider: هو المسؤول عن بناء "الفواصل" بين العناصر أو Divider SizedBox).

item account: عدد العناصر الكلى في القائمة

أنه لا يضع فاصلاً قبل العنصر الأول ولا بعد العنصر الأخير، هو يعمل فقط "بين" separatorBuilder الجميل في للتأكد من مكان الفاصل (if statements) العناصر، مما يوفر عليك عناء كتابة شروط برمجية

```
lib/main.dart
22 Widget build(BuildContext context) {
23   return MaterialApp(
24     home: Scaffold(
25       appBar: AppBar(title: const Text("Title")),
26       body: ListView(
27         children: [
28           Container(
29             // height: 400,
30             child: ListView.separated(
31               shrinkWrap: true,
32               physics: NeverScrollableScrollPhysics(),
33               separatorBuilder: (context, i) {
34                 return Divider(color: Colors.black, thickness: 2.0);
35               },
36               itemCount: employee.length,
37               itemBuilder: (context, i) {
38                 return Container(
39                   height: 100,
40                   color: i.isEven ? Colors.red : Colors.green,
41                   child: Text(employee[i]['name'],
42                     textAlign: TextAlign.center,
43                     style: TextStyle(fontSize: 16, color: Colors.white));
44                 },
45               ), // ListView.separated
46             ), // Container
47           ],
48         ),
49       ),
50     ),
51   );
52 }
```

وهنا التطبيق العملي

shrinkWrap: true

بتخليني القائمة تضبضب حالها وما تطلب مساحة زياده، عشان تقدر تعيش بسلام جوا قائمه تانية بدون ما تضرب التصميم.

```
lib/main.dart
21 @override
22 Widget build(BuildContext context) {
23   return MaterialApp(
24     home: Scaffold(
25       appBar: AppBar(title: const Text("Title")),
26       body: Container(
27         // height: 400,
28         child: GridView.builder(
29           gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(crossAxis
30             .count: 2),
31           itemCount: employee.length,
32           itemBuilder: (context, i) {
33             return Container(
34               height: 100,
35               color: i.isRed ? Colors.red : Colors.green,
36               child: Text(
37                 employee[i]['name'],
38                 textAlign: TextAlign.center,
39                 style: TextStyle(fontSize: 16, color: Colors.white));
40             );
41           }, // GridView.builder
42         ), // Container
43       ), // Scaffold
44     ); // MaterialApp
45 }
```

```
main.dart
lib > main.dart > MyApp > pubspec.yaml

7 class MyApp extends StatelessWidget {
8   MyApp({super.key});
9
10  final List employee = [
11    {"name": "wael", "age": 12, "lastname": "abo hamza"},
12    {"name": "Mohammad", "age": 16, "lastname": "abo hamza"},
13    {"name": "Shady", "age": 16, "lastname": "abo hamza"},
14    {"name": "Thaer", "age": 16, "lastname": "abo hamza"},
15    {"name": "Mohand", "age": 16, "lastname": "abo hamza"},
16    {"name": "Yaser", "age": 16, "lastname": "abo hamza"},
17    {"name": "Osama", "age": 16, "lastname": "abo hamza"},
18    {"name": "Basel", "age": 14, "lastname": "abo hamza"}
19  ];
20
21 @override
22 Widget build(BuildContext context) {
23   return MaterialApp(
24     home: Scaffold(
25       appBar: AppBar(title: const Text("Title")),
26       body: Container(
27         // height: 400,
28         child: GridView.builder(
29           gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(crossAxis
30           itemCount: employee.length,
31           itemBuilder: (context, i) {
32             return Container(
33               height: 100,
34               color: Colors.red,
35               child: Text(
36                 employee[i]['name'],
37                 textAlign: TextAlign.center,
38                 style: TextStyle(fontSize: 16, color: Colors.white),
39               ); // Text // Container
40             ),
41           ), // GridView.builder
42         ), // Container
43       ),
44     ),
45   );
46 }

PROBLEMS OUTPUT TERMINAL
```

DEBUG CONSOLE

```
Reloaded 1 of 665 libraries in 407ms (compile: 22 ms, reload: 160 ms, reassemble: 161 ms).
Reloaded 1 of 665 libraries in 385ms (compile: 21 ms, reload: 146 ms, reassemble: 147 ms).
Reloaded 1 of 665 libraries in 373ms (compile: 20 ms, reload: 156 ms, reassemble: 143 ms).
Reloaded 1 of 665 libraries in 384ms (compile: 21 ms, reload: 144 ms, reassemble: 171 ms).
Reloaded 1 of 665 libraries in 411ms (compile: 20 ms, reload: 172 ms, reassemble: 167 ms).
Reloaded 1 of 665 libraries in 371ms (compile: 19 ms, reload: 158 ms, reassemble: 158 ms).
Reloaded 1 of 665 libraries in 371ms (compile: 19 ms, reload: 158 ms, reassemble: 158 ms).
```

OUTPUT

```
Ln 34 Col 26 Spaces: 2 UFT-8 CR LF Dart Pixel 4 API 30 android x64 emulator 10:34 AM 26°C مدونه 6/5/2023
```

GridView

بما أئنا تكلمنا عن الـ ListView اللي بترتيب العناصر فوق بعض الـ "بتصف العناصر في صفو" وأعمدة زي شكل مربعات الصور في الإنستجرام أو تطبيقات المعرض

```
main.dart
lib > main.dart > MyApp > pubspec.yaml

18   {"name": "Basel", "age": 14, "lastname": "abo hamza"}
19  ];
20
21 @override
22 Widget build(BuildContext context) {
23   return MaterialApp(
24     home: Scaffold(
25       appBar: AppBar(title: const Text("Title")),
26       body: Container(
27         // height: 400,
28         child: GridView.builder(
29           gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(crossAxisCount: 4 , mainAxisSpacing: 10.0),
30           itemCount: employee.length,
31           itemBuilder: (context, i) {
32             return Container(
33               height: 100,
34               color: Colors.red,
35               child: Text(
36                 employee[i]['name'],
37                 textAlign: TextAlign.center,
38                 style: TextStyle(fontSize: 16, color: Colors.white),
39               ); // Text // Container
40             ),
41           ), // GridView.builder
42         ), // Container
43       ),
44     ),
45   );
46 }

PROBLEMS OUTPUT TERMINAL
```

DEBUG CONSOLE

```
Reloaded 1 of 665 libraries in 407ms (compile: 22 ms, reload: 162 ms, reassemble: 161 ms).
Reloaded 1 of 665 libraries in 385ms (compile: 21 ms, reload: 145 ms, reassemble: 171 ms).
Reloaded 1 of 665 libraries in 373ms (compile: 20 ms, reload: 156 ms, reassemble: 142 ms).
Reloaded 1 of 665 libraries in 384ms (compile: 21 ms, reload: 144 ms, reassemble: 171 ms).
Reloaded 1 of 665 libraries in 411ms (compile: 20 ms, reload: 172 ms, reassemble: 167 ms).
Reloaded 1 of 665 libraries in 371ms (compile: 19 ms, reload: 158 ms, reassemble: 158 ms).
Reloaded 1 of 665 libraries in 369ms (compile: 20 ms, reload: 152 ms, reassemble: 145 ms).
Reloaded 1 of 665 libraries in 367ms (compile: 20 ms, reload: 155 ms, reassemble: 139 ms).
Reloaded 1 of 665 libraries in 355ms (compile: 19 ms, reload: 149 ms, reassemble: 131 ms).
```

OUTPUT

```
Ln 29 Col 111 Spaces: 2 UFT-8 CR LF Dart Pixel 4 API 30 android x64 emulator 10:35 AM 26°C USD/EGP +0.32% ⌂ ENG 6/5/2023
```

```
18 "name": "Basel", "age": 14, "lastname": "abo hamza"}  
19  
20  
21 override  
22 get build(BuildContext context) {  
23 return MaterialApp(  
24   home: Scaffold(  
25     appBar: AppBar(title: const Text("Title")),  
26     body: Container(  
27       // height: 400,  
28       itemCount: employee.length,  
29       itemBuilder: (context, i) {  
30         return Container(  
31           height: 100,  
32           color: Colors.red,  
33           child: Text(  
34             employee[i]['name'],  
35             textAlign: TextAlign.center,  
36             style: TextStyle(fontsize: 16, color: Colors.white),  
37           );  
38         },  
39       ),  
40     ),  
41   ), // Container  
42 ); // Container
```

أهم المصطلحات اللي لازم نعرفها

عدد العواميد crossAxisCount 2 أو 3

المسافة الطولية بين المربعات mainAxisSpacing

المسافة العرضية بين المربعات crossAxisSpacing

نسبة الطول للعرض للمربع يعني بده المربع يكون مربع مثالي ولا مستطيل نائم ولا مستطيل واقف childAspectRatio

```
main.dart
18     "name": "Basel", "age": 14, "lastname": "abo hamza"
19   ];
20
21   @override
22   Widget build(BuildContext context) {
23     return MaterialApp(
24       home: Scaffold(
25         appBar: AppBar(title: const Text("Title")),
26         body: Container(
27           height: 400,
28           child: GridView.builder(
29             gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(crossAxisCount: 4, mainAxisExtent: 250.0),
30             itemCount: employee.length,
31             itemBuilder: (context, i) {
32               return Container(
33                 height: 100,
34                 color: Colors.red,
35                 child: Text(
36                   employee[i]['name'],
37                   textAlign: TextAlign.center,
38                   style: TextStyle(fontSize: 16, color: Colors.white),
39                 );
40               },
41             ), // GridView.builder
42           ); // Container
43     );
44   }
45 }

Wael Abo Hamza
```

Reloaded 1 of 665 libraries in 407ms (compile: 22 ms, reload: 162 ms, reassemble: 161 ms).
Reloaded 1 of 665 libraries in 385ms (compile: 20 ms, reload: 156 ms, reassemble: 157 ms).
Reloaded 1 of 665 libraries in 384ms (compile: 20 ms, reload: 156 ms, reassemble: 163 ms).
Reloaded 1 of 665 libraries in 384ms (compile: 21 ms, reload: 164 ms, reassemble: 171 ms).
Reloaded 1 of 665 libraries in 411ms (compile: 20 ms, reload: 173 ms, reassemble: 167 ms).
Reloaded 1 of 665 libraries in 371ms (compile: 19 ms, reload: 158 ms, reassemble: 158 ms).
Reloaded 1 of 665 libraries in 369ms (compile: 20 ms, reload: 152 ms, reassemble: 145 ms).
Reloaded 1 of 665 libraries in 367ms (compile: 19 ms, reload: 155 ms, reassemble: 139 ms).

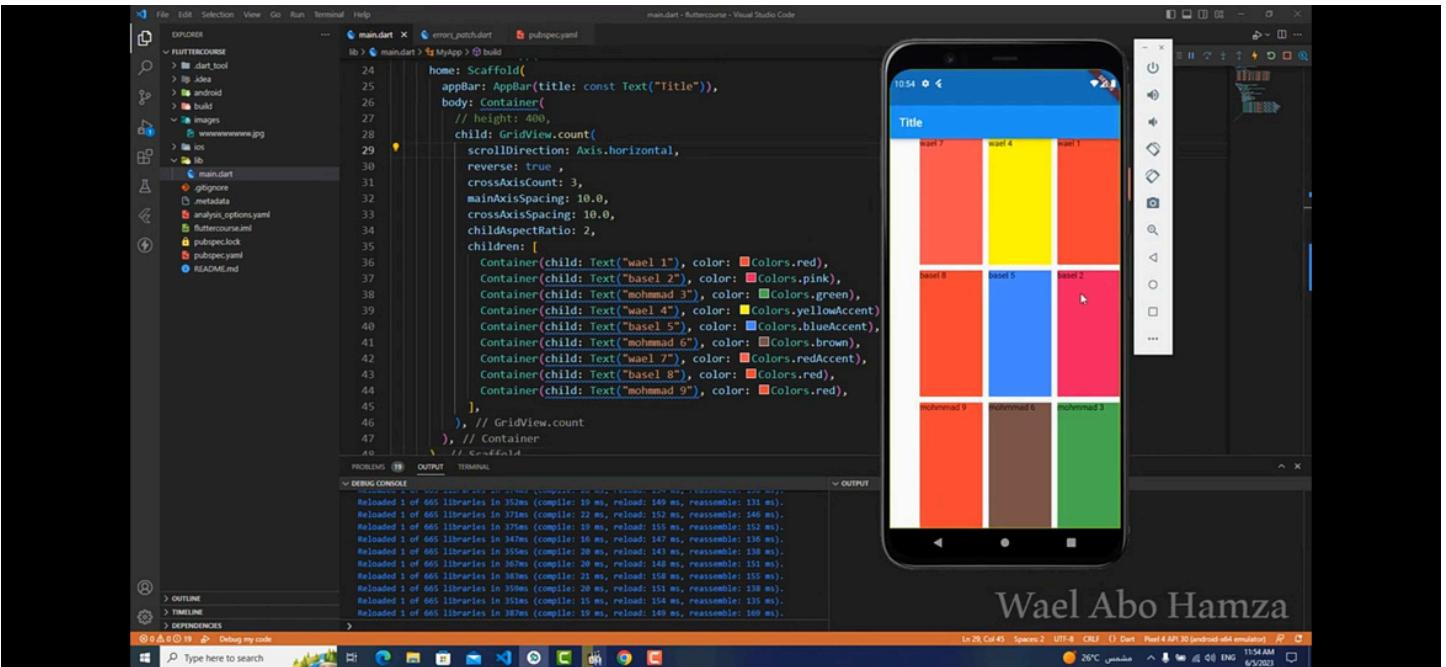
كانت هيک ..

صارت هيک لما غيرنا خليناها هنا 250.

```
main.dart
18     "name": "Basel", "age": 14, "lastname": "abo hamza"
19   ];
20
21   @override
22   Widget build(BuildContext context) {
23     return MaterialApp(
24       home: Scaffold(
25         appBar: AppBar(title: const Text("Title")),
26         body: Container(
27           height: 400,
28           child: GridView.builder(
29             gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(crossAxisCount: 4, mainAxisSpacing: 25.0),
30             itemCount: employee.length,
31             itemBuilder: (context, i) {
32               return Container(
33                 height: 100,
34                 color: Colors.red,
35                 child: Text(
36                   employee[i]['name'],
37                   textAlign: TextAlign.center,
38                   style: TextStyle(fontSize: 16, color: Colors.white),
39                 );
40               },
41             ), // GridView.builder
42           ); // Container
43     );
44   }
45 }

Wael Abo Hamza
```

Reloaded 1 of 665 libraries in 367ms (compile: 22 ms, reload: 156 ms, reassemble: 139 ms).
Reloaded 1 of 665 libraries in 353ms (compile: 20 ms, reload: 150 ms, reassemble: 137 ms).
Reloaded 1 of 665 libraries in 353ms (compile: 20 ms, reload: 156 ms, reassemble: 136 ms).
Reloaded 1 of 665 libraries in 364ms (compile: 22 ms, reload: 166 ms, reassemble: 126 ms).
Reloaded 1 of 665 libraries in 366ms (compile: 20 ms, reload: 160 ms, reassemble: 150 ms).
Reloaded 1 of 665 libraries in 370ms (compile: 21 ms, reload: 147 ms, reassemble: 152 ms).
Reloaded 1 of 665 libraries in 368ms (compile: 20 ms, reload: 158 ms, reassemble: 139 ms).
Reloaded 1 of 665 libraries in 354ms (compile: 20 ms, reload: 149 ms, reassemble: 134 ms).
Reloaded 1 of 665 libraries in 359ms (compile: 20 ms, reload: 146 ms, reassemble: 140 ms).
Reloaded 1 of 665 libraries in 361ms (compile: 21 ms, reload: 147 ms, reassemble: 139 ms).
Reloaded 1 of 665 libraries in 361ms (compile: 23 ms, reload: 152 ms, reassemble: 135 ms).



GridView.count

تم استخدام GridView.count لأن عدد الأعمدة ثابت

: جعل التصفح (التحريك) أفقياً من اليمين لليسار أو العكس بدلاً من الطريقة الرأسية المعتادة.

جعل ترتيب العناصر يبدأ من الجهة المعاكسة (في هذه الحالة يبدأ من اليمين بسبب التصفح الأفقي).

يحدد عدد العناصر في المحور المتقاطع بما أن التصفح أفقي، فهذا يعني وجود 3 صفوف

: إضافة مسافات (10 بكسل) بين المربعات لتبدو منتظمة

يتحكم في نسبة العرض إلى الارتفاع لكل خلية (في الصورة جعل المربعات مستطيلة الشكل) childAspectRatio

Gridview

```

class MyApp extends StatelessWidget {
  MyApp({super.key});

  final List employee = [
    {"name": "wael", "age": 12, "lastname": "abo hamza"},
    {"name": "Mohammad", "age": 16, "lastname": "abo hamza"},
    {"name": "Shady", "age": 16, "lastname": "abo hamza"},
    {"name": "Thaer", "age": 16, "lastname": "abo hamza"},
    {"name": "Mohand", "age": 16, "lastname": "abo hamza"},
    {"name": "Yaser", "age": 16, "lastname": "abo hamza"},
    {"name": "Osama", "age": 16, "lastname": "abo hamza"},
    {"name": "Basel", "age": 14, "lastname": "abo hamza"}
  ];

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: const Text("Title")),
        body: Container(
          // height: 400,
          child: GridView.builder(
            gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(crossAxisCount: 2),
            itemCount: employee.length,
          ),
        ),
      ),
    );
  }
}

```

Wael Abo Hamza

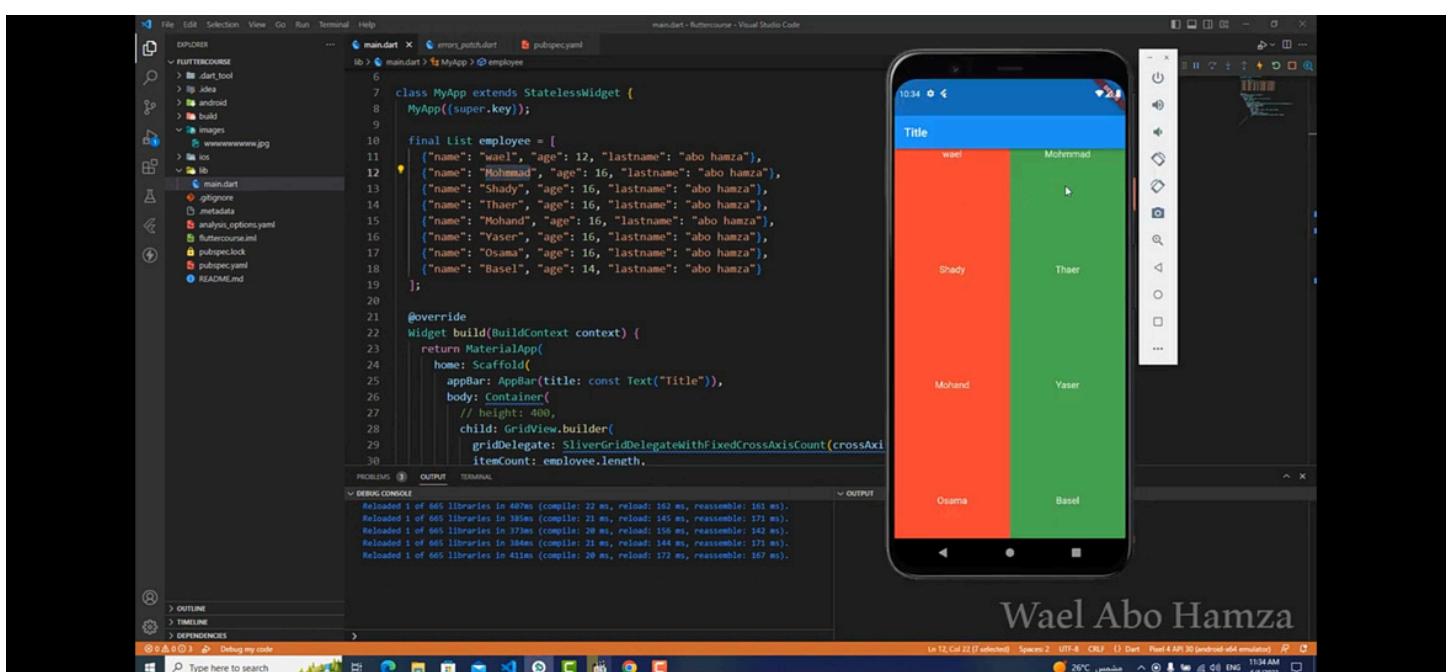
خصائص الـ Gridview المستعملة:

هنا تم تحديد 2: crossAxisCount: وهذا يعني أن الشاشة ستعرض عمودين من البيانات جنباً إلى جنب.

تم ربطه بـ itemCount لكي نعرف التطبيق عدد المربعات التي يجب رسمها بناء على عدد الموظفين الموجودين في القائمة

إختصار الكود بجهز قاعدة بيانات صغيرة للموظفين ويطلب من تطبيق Flutter أن يعرضهم في شاشة مقسمة إلى عمودين بشكل منظم وتلقائي

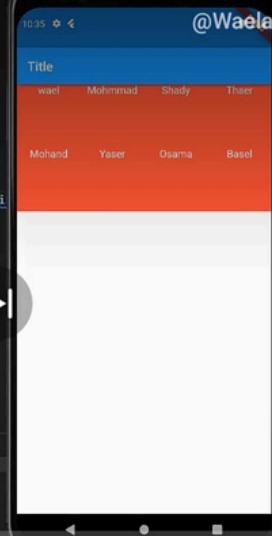
وهيك شكلها بيطلع



وبتقدر خلية 4 على سطر واحد crossAxisCount من خلال هذا الفنكشن

11 - Gridview (flutter 2023)

@WaelabohamzaFlutter



2:22 / 11:53 في هذا الفيديو المزيد من الفيديوهات

```
lib/main.dart:18:18: Error: The argument type 'int' can't be assigned to the parameter type 'double'.  
      crossAxisCount: 4.0,  
           ^^^^^^  
lib/main.dart:18:24: Error: The argument type 'double' can't be assigned to the parameter type 'int'.  
      mainAxisSpacing: 10.0,  
                      ^^^^^^  
lib/main.dart:18:30: Error: The argument type 'double' can't be assigned to the parameter type 'int'.  
      crossAxisSpacing: 10.0,  
                      ^^^^^^  
lib/main.dart:18:36: Error: The argument type 'double' can't be assigned to the parameter type 'int'.  
      childAspectRatio: 4.0,  
                      ^^^^^^  
lib/main.dart:18:42: Error: The argument type 'double' can't be assigned to the parameter type 'int'.  
      itemCount: 8.0,  
                      ^^^^^^  
lib/main.dart:18:48: Error: The argument type 'double' can't be assigned to the parameter type 'int'.  
      itemBuilder: (context, i) {  
                      ^^^^^^  
lib/main.dart:18:54: Error: The argument type 'double' can't be assigned to the parameter type 'int'.  
        height: 100.0,  
                ^^^^^^  
lib/main.dart:18:60: Error: The argument type 'double' can't be assigned to the parameter type 'int'.  
        color: Colors.red,  
                ^^^^^^  
lib/main.dart:18:66: Error: The argument type 'double' can't be assigned to the parameter type 'int'.  
        child: Text(  
                ^^^^^^  
lib/main.dart:18:72: Error: The argument type 'double' can't be assigned to the parameter type 'int'.  
          employee[i]['name'],  
                  ^^^^^^  
lib/main.dart:18:78: Error: The argument type 'double' can't be assigned to the parameter type 'int'.  
          textAlign: TextAlign.center,  
                  ^^^^^^  
lib/main.dart:18:84: Error: The argument type 'double' can't be assigned to the parameter type 'int'.  
          style: TextStyle(fontSize: 16, color: Colors.white),  
                  ^^^^^^  
        ); // Text // Container  
      },  
    ), // GridView.builder  
  ), // Container
```

... Wael Al Hamza

Wael Abo Hamza

```
lib/main.dart:18:18: Error: The argument type 'int' can't be assigned to the parameter type 'double'.  
      crossAxisCount: 4.0,  
           ^^^^^^  
lib/main.dart:18:24: Error: The argument type 'double' can't be assigned to the parameter type 'int'.  
      mainAxisSpacing: 10.0,  
                      ^^^^^^  
lib/main.dart:18:30: Error: The argument type 'double' can't be assigned to the parameter type 'int'.  
      crossAxisSpacing: 10.0,  
                      ^^^^^^  
lib/main.dart:18:36: Error: The argument type 'double' can't be assigned to the parameter type 'int'.  
      childAspectRatio: 4.0,  
                      ^^^^^^  
lib/main.dart:18:42: Error: The argument type 'double' can't be assigned to the parameter type 'int'.  
      itemCount: 8.0,  
                      ^^^^^^  
lib/main.dart:18:48: Error: The argument type 'double' can't be assigned to the parameter type 'int'.  
      itemBuilder: (context, i) {  
                      ^^^^^^  
lib/main.dart:18:54: Error: The argument type 'double' can't be assigned to the parameter type 'int'.  
        height: 100.0,  
                ^^^^^^  
lib/main.dart:18:60: Error: The argument type 'double' can't be assigned to the parameter type 'int'.  
        color: Colors.red,  
                ^^^^^^  
lib/main.dart:18:66: Error: The argument type 'double' can't be assigned to the parameter type 'int'.  
        child: Text(  
                ^^^^^^  
lib/main.dart:18:72: Error: The argument type 'double' can't be assigned to the parameter type 'int'.  
          employee[i]['name'],  
                  ^^^^^^  
lib/main.dart:18:78: Error: The argument type 'double' can't be assigned to the parameter type 'int'.  
          textAlign: TextAlign.center,  
                  ^^^^^^  
lib/main.dart:18:84: Error: The argument type 'double' can't be assigned to the parameter type 'int'.  
          style: TextStyle(fontSize: 16, color: Colors.white),  
                  ^^^^^^  
        ); // Text // Container  
      },  
    ), // GridView.builder  
  ), // Container
```

```
main.dart
18     "name": "Basel", "age": 14, "lastname": "abo hamza"
19   ];
20
21   @override
22   Widget build(BuildContext context) {
23     return MaterialApp(
24       home: Scaffold(
25         appBar: AppBar(title: const Text("Title")),
26         body: Container(
27           height: 400,
28           child: GridView.builder(
29             gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(crossAxisCount: 4),
30             itemCount: employee.length,
31             itemBuilder: (context, i) {
32               return Container(
33                 height: 100,
34                 color: Colors.red ,
35                 child: Text(
36                   employee[i]['name'],
37                   textAlign: TextAlign.center,
38                   style: TextStyle(fontSize: 16, color: Colors.white),
39                 );
40               },
41             ), // GridView.builder
42           ); // Container
43     );
44   }
45 }
```

Wael Abo Hamza

عدد المسافات من خلال mainAxisSpacing

```
main.dart
18     "name": "Basel", "age": 14, "lastname": "abo hamza"
19   ];
20
21   @override
22   Widget build(BuildContext context) {
23     return MaterialApp(
24       home: Scaffold(
25         appBar: AppBar(title: const Text("Title")),
26         body: Container(
27           height: 400,
28           child: GridView.builder(
29             gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(crossAxisCount: 4),
30             itemCount: employee.length,
31             itemBuilder: (context, i) {
32               return Container(
33                 height: 100,
34                 color: Colors.red ,
35                 child: Text(
36                   employee[i]['name'],
37                   textAlign: TextAlign.center,
38                   style: TextStyle(fontSize: 16, color: Colors.white),
39                 );
40               },
41             ), // GridView.builder
42           ); // Container
43     );
44   }
45 }
```

io: 0.5

Wael Abo Hamza

```

main.dart
18 "name": "Basel", "age": 14, "lastname": "abo hamza"}
19
20
21   @override
22   get build(BuildContext context) {
23     return MaterialApp(
24       home: Scaffold(
25         appBar: AppBar(title: const Text("Title")),
26         body: Container(
27           height: 400,
28           child: GridView.builder(
29             gridDelegate: SilverGridDelegateWithFixedCrossAxisCount(crossAxisCount: 4 , mainAxisSpacing: 10.0 , childAspectRatio: 0.5),
30             itemCount: employee.length,
31             itemBuilder: (context, i) {
32               return Container(
33                 height: 100,
34                 color: Colors.red ,
35                 child: Text(
36                   employee[i]['name'],
37                   textAlign: TextAlign.center,
38                   style: TextStyle(fontSize: 16, color: Colors.white),
39                 );
40               }, // Text // Container
41             ), // GridView.builder
42           ), // Container

```

Wael Abo Hamza

```

main.dart
24   home: Scaffold(
25     appBar: AppBar(title: const Text("Title")),
26     body: Container(
27       height: 400,
28       child: GridView.count(
29         reverse: true ,
30         crossAxisCount: 3,
31         mainAxisSpacing: 10.0,
32         crossAxisSpacing: 10.0,
33         childAspectRatio: 2,
34         children: [
35           Container(child: Text("wael 1"), color: Colors.red),
36           Container(child: Text("basel"), color: Colors.pink),
37           Container(child: Text("mohmmad"), color: Colors.green),
38           Container(child: Text("wael"), color: Colors.yellowAccent),
39           Container(child: Text("basel"), color: Colors.blueAccent),
40           Container(child: Text("mohmmad"), color: Colors.brown),
41           Container(child: Text("wael"), color: Colors.redAccent),
42           Container(child: Text("basel"), color: Colors.red),
43           Container(child: Text("mohmmad"), color: Colors.red),
44         ],
45       ), // GridView.count
46     ), // Container
47   ), // Scaffold

```

Wael Abo Hamza

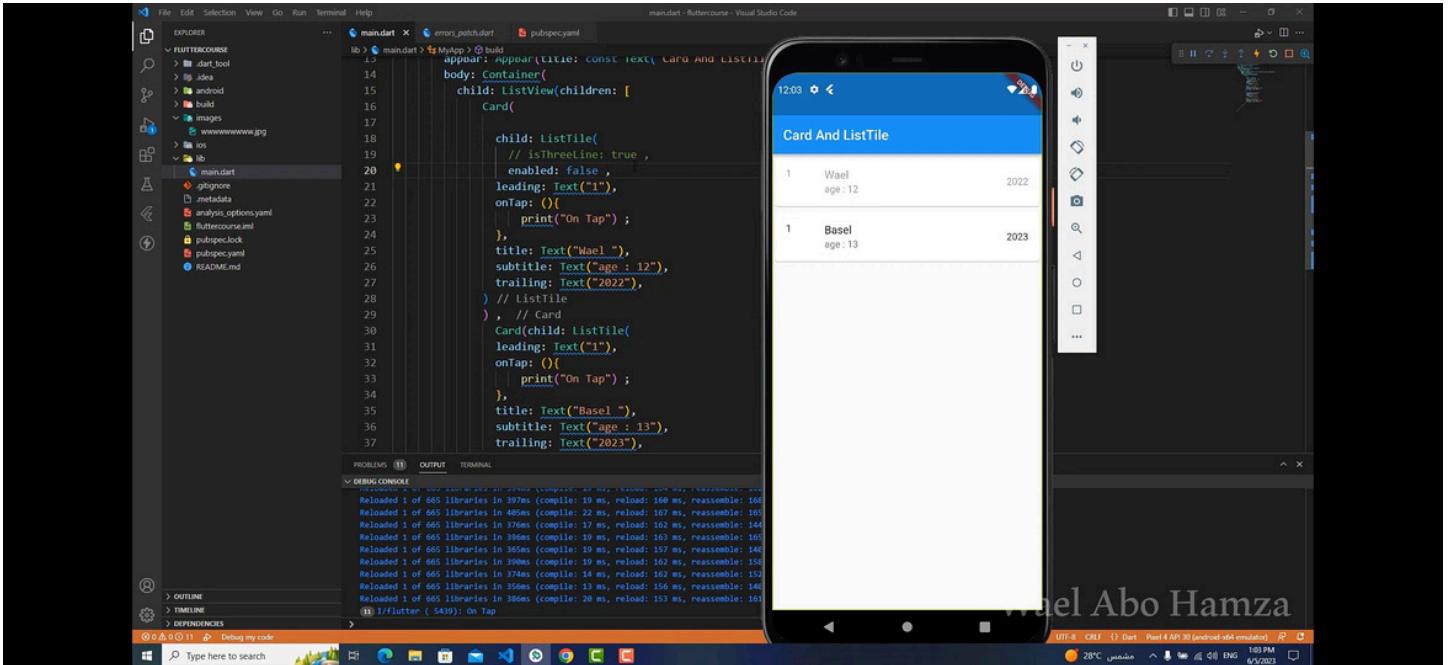
لإنشاء شبكة من العناصر GridView.count

: يعني تقسيم الشاشة إلى 3 أعمدة crossAxisCount: 3

: جعل ترتيب العناصر يبدأ من الأسفل إلى الأعلى reverse: true

: لتحديد المسافات بين المربعات crossAxisSpacing و mainAxisSpacing

ListTitle and card



Card: يُستخدم لإعطاء مظهر "البطاقة" مع خلفية بيضاء وظل خفيف.

ListTile: أداة جاهزة لترتيب محتوى السطر الواحد، وت تكون من:

الجزء الأمامي (هنا يعرض الرقم 1)

title: العنوان الرئيسي أو (الاسم: Wael Basel).

subtitle: العنوان الفرعی (العمر) age: 12).

الجزء الخلفي في نهاية السطر (السنة: 2022).

IconAndIconButton

The screenshot shows the Visual Studio Code interface with the following details:

- File Structure (EXPLORER):** Shows the project structure for "FLUTTERCOURSE" with files like main.dart, errors_patch.dart, and pubspec.yaml.
- Code Editor:** Displays the main.dart file with Dart code:

```
lib > main.dart > MyApp > build
3 void main() {
4   runApp(const MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   const MyApp({super.key});
9   @override
10  Widget build(BuildContext context) {
11    return MaterialApp(
12      home: Scaffold(
13        appBar: AppBar(title: const Text("Icon And IconButton")),
14        body: Container(child: Icon(Icons.star , color: Colors.red,size: 100,)),
15      ), // Scaffold
16    ); // MaterialApp
17  }
18 }
```
- Terminal:** Shows the output of the build process:

```
Reloaded 1 of 665 libraries in 389ms (compile: 16 ms, reload: 126 ms, reassemble: 118 ms).
Reloaded 1 of 665 libraries in 386ms (compile: 17 ms, reload: 120 ms, reassemble: 118 ms).
Reloaded 1 of 665 libraries in 286ms (compile: 17 ms, reload: 121 ms, reassemble: 101 ms).
```
- Output Panel:** Shows the Android emulator output with the application title "Icon And IconButton" and a large red star icon.
- Bottom Status Bar:** Shows the date (6/5/2023), time (1:40 PM), and system status (29°C, ENG).

Wael Abo Hamza

The screenshot shows the Visual Studio Code interface with the following details:

- File Structure (EXPLORER):** Shows the project structure for "FLUTTERCOURSE" with files like main.dart, errors_patch.dart, and pubspec.yaml.
- Code Editor:** Displays the main.dart file with Dart code:

```
lib > main.dart > MyApp > build
3 void main() {
4   runApp(const MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   const MyApp({super.key});
9   @override
10  Widget build(BuildContext context) {
11    return MaterialApp(
12      home: Scaffold(
13        appBar: AppBar(title: const Text("Icon And IconButton")),
14        body: Container(child: Icon(Icons.star , color: Colors.red,size: 10),
15      ), // Scaffold
16    ); // MaterialApp
17  }
18 }
```
- Terminal:** Shows the output of the build process:

```
Reloaded 1 of 665 libraries in 399ms (compile: 16 ms, reload: 126 ms, reassemble: 118 ms).
Reloaded 1 of 665 libraries in 386ms (compile: 17 ms, reload: 120 ms, reassemble: 118 ms).
Reloaded 1 of 665 libraries in 286ms (compile: 17 ms, reload: 121 ms, reassemble: 101 ms).
```
- Output Panel:** Shows the Android emulator output with the application title "Icon And IconButton" and a smaller red star icon.
- Bottom Status Bar:** Shows the date (6/5/2023), time (1:40 PM), and system status (29°C, ENG).

ونستطيع التغيير بالحجم والشكل واللون

The screenshot shows the Visual Studio Code interface with the Flutter project 'FLUTTERCOURSE'. The code editor displays the file 'main.dart' which contains the following code:

```
void main() { runApp(const MyApp()); }

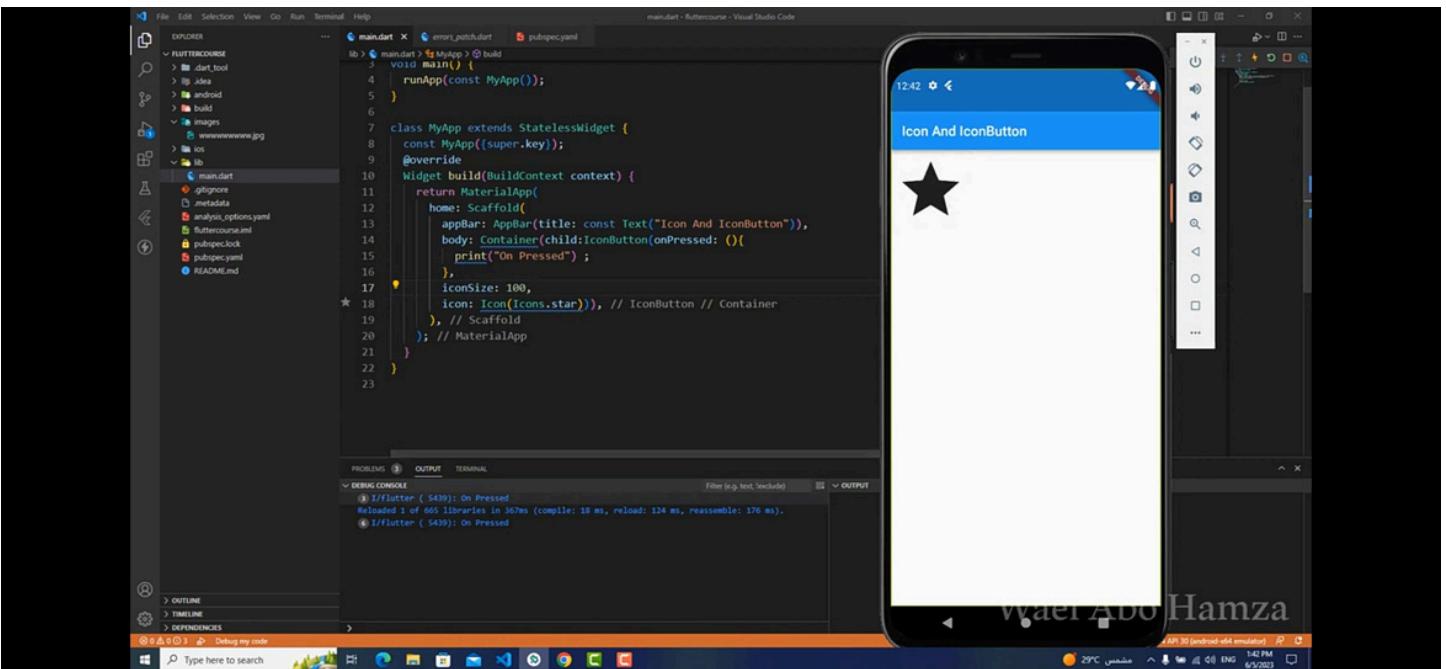
class MyApp extends StatelessWidget {
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      appBar: AppBar(title: const Text("Icon And IconButton")),
      body: Container(child: IconButton(onPressed: () {
        print("On Pressed");
      }, icon: Icon(Icons.star)), // IconButton // Container
      ), // Scaffold
    ); // MaterialApp
  }
}
```

The Android emulator window shows a black screen with the text "Icon And IconButton" at the top and "Wael Abo Hamza" below it. The status bar indicates "API 30 (predicted x64 emulator)".

The screenshot shows the Visual Studio Code interface with the Flutter project 'FLUTTERCOURSE'. The code editor displays the file 'main.dart' which contains the same code as the previous screenshot.

The iOS simulator window shows a black screen with the text "Icon And IconButton" at the top and "Wael Abo Hamza" below it. The status bar indicates "API 30 (predicted x64 emulator)".

وعملنا زر وعند الضغط عليه يظهر نص



وعلمنا تكبير لزر .. ميزة رائعة نحتاجها بكل تطبيق

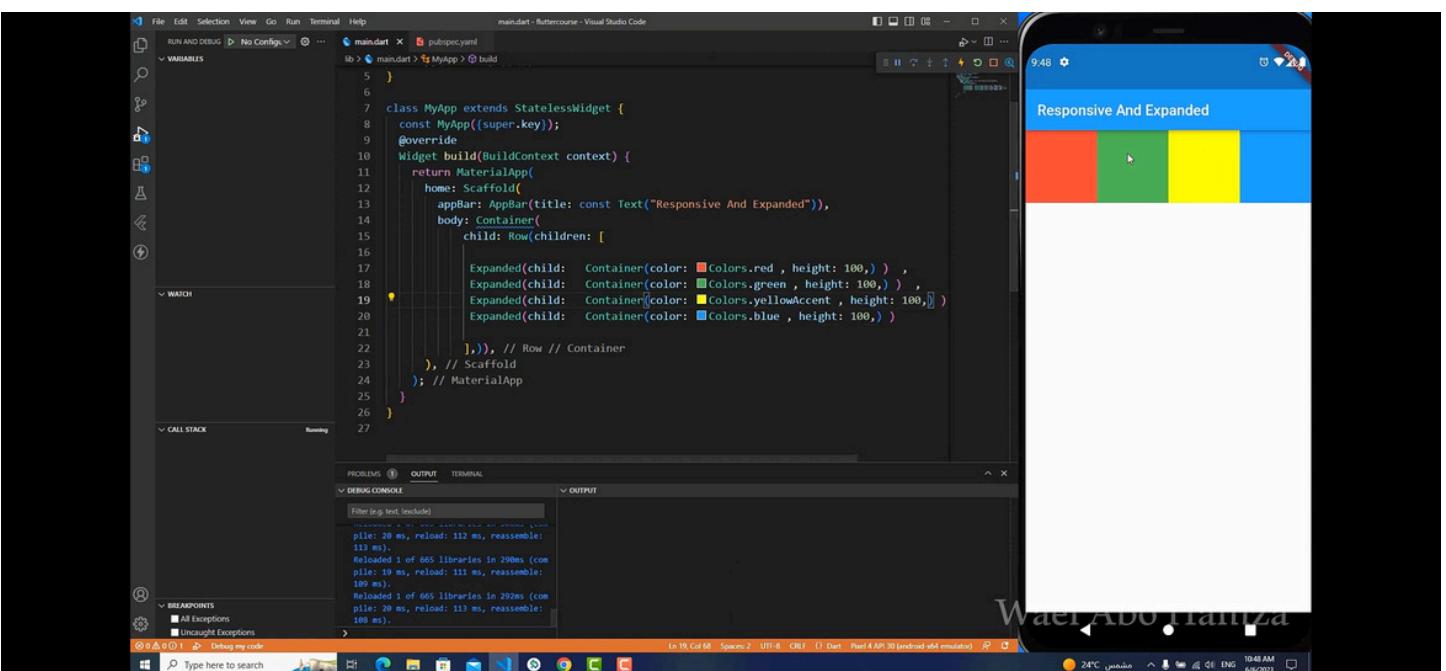
Responsive And Expanded

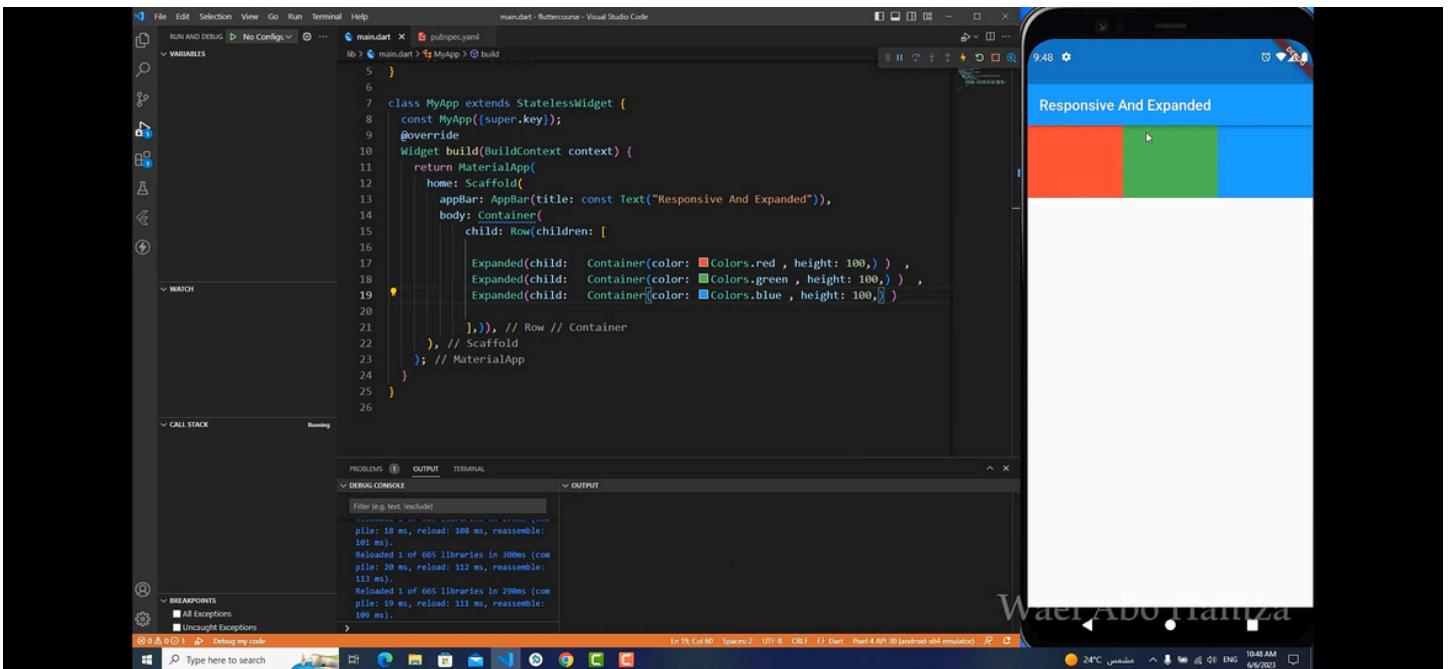
Responsive

وأن يتكيف المحتوى مع حجم الشاشة بدلاً من تصميم نسخة لكل جهاز نقوم بعمل تصميم واحد يتغير شكله تلقائياً

Expanded

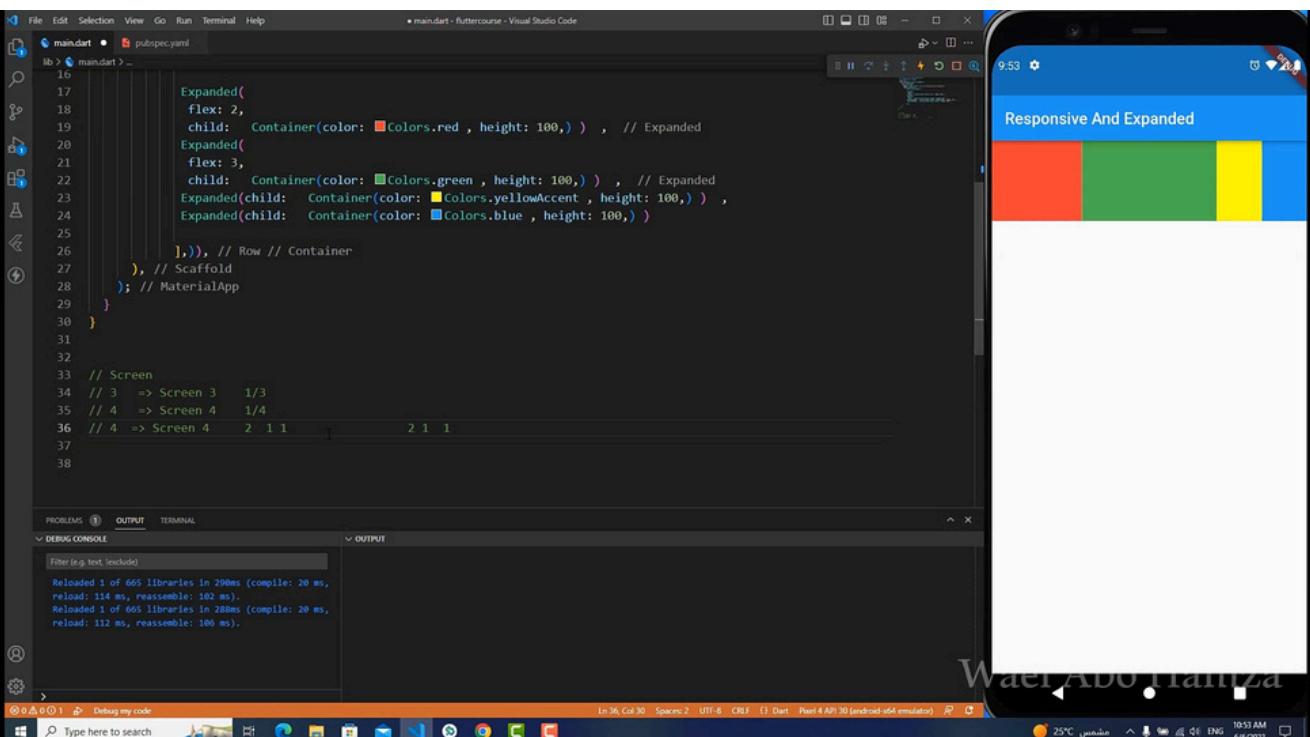
التمدد بالمساحة المتاحة إلها بالشاشة





بدون **Expanded** لو وضعت المربعات بعرض ثابت مثلا 200 لكل مربع ستظهر رسالة خطأ Overflow لأن المساحة لن تكفيهم في الشاشات الصغيرة

مع **Expanded** لأنك بتقول للتطبيق خذ المساحة المتوفرة في الشاشة مهما كان حجمها ووزعها بالتساوي على هذه العناصر



خاصية الـ **Flex** هي اللي بتحدد نسبة الحصة من الشاشة تعني أن هذا العنصر راح يأخذ حصتين من المساحة

تعني أن هذا العنصر سيأخذ ثلات حصص flex

النتيجة: العنصر ذو الـ flex الأكبر يتمدد أكثر من غيره وهذا يظهر بوضوح في الصورة حيث المربع الأخضر 3: flex: 3 أكبر من الأحمر 2: flex: 2