

Lesson(0) [Core Basics]

تحفيز: كل تطبيق عظيم... بدأ بـ Hello World



MaterialApp

✓ **MaterialApp** إيه هي ؟

"Material Design" مبني على تصميم جوجل **Flutter** هي الأساس اللي بيبدأ بيه أي تطبيق ،
يعني تقدر تقول إنها "الغلاف الكبير" اللي بيلف حوالين كل التطبيق
بتعمل حاجات مهمة زي:

- (home) تحدد الشاشة الرئيسية للتطبيق
- (title) تسمي التطبيق
- (الألوان والخطوط) theme تختار الـ
- لو هتتنقل بينهم (الصفحات) routes وتتحكم في حاجات زي الـ



MaterialApp كود بسيط على

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

void main() {
  runApp(MyApp()); // يبدأ التطبيق من هنا
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'تطبيقي الجامد',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: Scaffold(
        appBar: AppBar(
          title: Text('أهلاً بيك'),
        ),
        body: Center(
          child: Text('ليّا Flutter ده أول تطبيق! 🥳'),
        ),
      ),
    );
  }
}
```

(Output) الشكل اللي هيطلع:

- أول ما تفتح التطبيق
 - مكتوب فيه "أهلاً بيك (الشريط اللي فوق) AppBar هتلاقي
 - "ليّا Flutter وتحت، في نص الشاشة، فيه "ده أول تطبيق 🥳"
 - (blue) اللي اخترناه ThemeData والألوان جاية من الـ

ملخص سريع:

- Flutter بتبدأ بيها أي أبلكيشن = `MaterialApp`
- هي المسؤولة عن الشكل العام للتطبيق
- (`home`) جواها بتحت الصفحة الرئيسية

تحفيز: سطر كود اليوم، مشروع بكرة

CupertinoApp 🍏:

- iOS المستخدم في (تصميم آبل) **Cupertino Design التصميم**: يعتمد على
- **المكونات**: يتضمن مكونات مثل الأزرار ذات الزوايا الدائرية، شريط التنقل العلوي بتصميم وأيقونات مشابهة للتطبيقات الأصلية، iOS.
- مثل التمرير من اليمين iOS **الانتقالات**: الانتقالات بين الشاشات تكون مشابهة لتطبيقات (Swipe) اليسار.
- الأصلية ويعطي مظهرًا يشبه تطبيقات آبل iOS **التخصيص**: يركز أكثر على تقديم تجربة تمامًا.

CupertinoApp مثال لكود

```
dart
CopyEdit
CupertinoApp(
  title: 'Cupertino App',
  theme: CupertinoThemeData(
    primaryColor: CupertinoColors.activeBlue,
  ),
  home: CupertinoHomePage(),
);
```

الفرق بينهم اي بقا:

- **Material Design** للأندرويد 📱 ، تصميم **MaterialApp**.
- **Cupertino** (تصميم آبل) **Cupertino** تصميم 🍏 iOS ، **CupertinoApp**.

تحفيز: كل خطأ تتعلم منه، مش نهاية العالم



Scaffold

✓ Scaffold إيه هي ؟

Scaffold Flutter هي الهيكل الرئيسي لأي صفحة في تطبيق.

عشان تنظم المحتوى وتضيف الحاجات Scaffold يعني لما تيجي تبني شاشة، بتستخدم الأساسيّة زي:

- AppBar  → الشريط اللي فوق
- body  → المحتوى الرئيسي
- FloatingActionButton  → زر عائِم
- Drawer  → قائمة جانبية
- BottomNavigationBar  → شريط التنقل السفلي

Scaffold كود بسيط على

```
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: Text('Scaffold Demo'),
        ),
        body: Center(
          child: Text('!بتاع الصفحة Body ده الـ'),
        ),
        floatingActionButton: FloatingActionButton(
```

```

onPressed: () {
  print('تم الضغط على الزر');
},
child: Icon(Icons.add),
),
),
);
}
}

```

(Output): الشكل اللي هيطلع

- AppBar فيه عنوان "Scaffold Demo"
- "إبتاع الصفحة Body في نص الشاشة: "ده الـ"
- زر دائري فيه علامة + في الزاوية السفلى من الشاشة

ملخص سريع:

- **Scaffold** = لبنة أي صفحة في التطبيق
- محتوى، قوائم، أزرار، وكل حاجة ف مكانها AppBar، بيسهّل عليك تحط




تحفيز: برمج مستقبلك بيدك




AppBar إيه هي ؟

هو الشريط اللي بيظهر في أعلى كل صفحة **AppBar**

بيكون فيه غالباً:

- (title) عنوان الصفحة 
- مثل البحث أو الإعدادات (actions) أزرار في الجنب 
- زر القائمة الجانبية (**Drawer**) 

وده بيدي التطبيق شكل احترافي ومنظم 

كود بسيط على AppBar

```
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: Text('عنوان الصفحة'),
          actions: [
            IconButton(
              icon: Icon(Icons.search),
              onPressed: () {
                print('ضغطت على بحث');
              },
            ),
          ],
        ),
        body: Center(
          child: Text('AppBar محتوى الصفحة تحت ال👉'),
        ),
      ),
    );
  }
}
```

(Output) الشكل اللي هيطلع:

- شريط علوي أزرق (افتراضي)

- "مكتوب فيه" عنوان الصفحة
- على الجنب فيه أيقونة 🔍 للبحث
- AppBar وتحت الشريط، محتوى الصفحة مكتوب فيه: "محتوى الصفحة تحت الـ"

🎯 ملخص سريع:

- الشريط العلوي في الصفحة = AppBar
- Scaffold وببشتغل تلقائي مع , actions , title فيه
- تقدر تضيف فيه أيقونات أو قوائم بكل سهولة

تحفيز: التكرار يبخليك محترف، استمر بس



Container

✅ ؟ Container إيه هي

هو زي صندوق في التطبيق، بتستخدمه لتنسيق وتحديد حجم و محتوى العناصر Container ال داخله.

1. تحديد الحجم (Width و Height)

يقدر يحدد العرض و الارتفاع Container ال

🔧 كود:

```
Container(
  width: 200.0, // العرض
  height: 200.0, // الارتفاع
  color: Colors.blue, // اللون
)
```

🖼️ Output:

- بكسل 200x200 هتلاقى مربع أزرق بحجم 200

2. تحديد اللون (Color)

. `color` باستخدام خاصية `Container` تقدر تحدد لون الخلفية للـ

 **كود:**

```
Container(  
  width: 200.0,  
  height: 200.0,  
  color: Colors.green, // اللون  
)
```



Output:

- بكسل 200x200 هتلاقي مربع أخضر بحجم 200

3. المسافات الداخلية (Padding)

. `Container` يحدد المسافة بين المحتوى وحدود الـ `padding` الـ

 **كود:**

```
Container(  
  width: 200.0,  
  height: 200.0,  
  color: Colors.blue,  
  padding: EdgeInsets.all(20.0), // المسافة الداخلية  
)
```



Output:

- هتلاقي مربع أزرق لكن النص جوه بعيد عن الحدود بمقدار 20 بكسل من جميع الجوانب

4. المسافات الخارجية (Margin)

والعناصر الثانية اللي حواليه `Container` يحدد المسافة بين الـ `margin` الـ

 **كود:**


```
Container(  
  width: 200.0,  
  height: 200.0,  
  color: Colors.red,  
  margin: EdgeInsets.all(30.0), // المسافة الخارجية  
)
```

Output:

- هتلاقى مربع أحمر بعيد عن العناصر الثانية بمقدار 30 بكسل.

5. المحاذاة (Alignment)

. `Container` يحدد مكان المحتوى جوه ال `alignment` ال

كود:

```
Container(  
  width: 200.0,  
  height: 200.0,  
  color: Colors.yellow,  
  alignment: Alignment.center, // المحاذاة في النص  
  child: Text('محتوى'),  
)
```

Output:

- هتلاقى مربع أصفر مع النص في النص تمامًا.

6. الزخرفة (Decoration)

. تقدر من خلالها تضيف حدود أو ظلال أو زوايا منحنية `decoration` ال

كود:

```

Container(
  width: 200.0,
  height: 200.0,
  color: Colors.pink,
  decoration: BoxDecoration(
    borderRadius: BorderRadius.circular(15.0), // الزوايا المنحنية
    boxShadow: [ // الظلال
      BoxShadow(
        color: Colors.black.withOpacity(0.2),
        blurRadius: 5.0,
        offset: Offset(2, 2),
      ),
    ],
  ),
)

```

Output:

- هتلاقى مربع وردي مع زوايا منحنية وظلال خفيفة.

ملخص سريع:

- هو صندوق يمكننا من تنسيق العناصر وتحديد الحجم، اللون، المسافات `Container` الـ الداخلية والخارجية، المحاذاة، و التأثيرات زي الظلال و الزوايا المنحنية.

تحفيز: البرمجة مش صعبة، محتاجة صبر وتدريب



Padding

`Padding` إيه هي ؟

أو أي `Container` الـ **Widget** بتستخدم لتحديد المسافة بين المحتوى وحدود الـ `Padding` الـ تكون مريحة بين العناصر اللي فيه `Widget` يعني بتخلي المسافة داخل الـ (تاني `Widget`).

1. `EdgeInsets.all()`

بتحدد **مسافة متساوية** من جميع الاتجاهات (فوق، تحت، يمين، شمال).

 **كود:**

```
Container(  
  padding: EdgeInsets.all(20.0), // المسافة الداخلية من جميع الاتجاهات  
  color: Colors.blue,  
  child: Text('Padding محتوى داخل'),  
)
```

 **Output:**

- هتلاقى **مربع أزرق** في النص فيه نص بـ 20 بكسل مسافة من كل الجوانب.

2. **EdgeInsets.symmetric()**

Vertical أو **Horizontal** (يمين وشمال) بتحدد **المسافة بشكل متساوي** في الـ (فوق) **Vertical** أو الـ (يمين وشمال) **Horizontal** بتحدد **المسافة بشكل متساوي** في الـ (وتحت).

 **كود:**

```
Container(  
  padding: EdgeInsets.symmetric(horizontal: 30.0, vertical: 10.0), // المسافات الـ  
  متساوية عموديًا وأفقيًا  
  color: Colors.green,  
  child: Text('Padding متساوي من اليمين والشمال'),  
)
```

 **Output:**

- هتلاقى **مربع أخضر** فيه نص مع مسافة 30 بكسل من اليمين والشمال، و 10 بكسل من فوق وتحت.

3. **EdgeInsets.only()**

بتحدد **المسافة في اتجاهات معينة** (فوق، تحت، يمين، شمال).

كود:

```
Container(  
  padding: EdgeInsets.only(top: 20.0, left: 40.0), // المسافة فقط من فوق واليسار  
  color: Colors.red,  
  child: Text('Padding معينة اتجاهات'),  
)
```



Output:

- هتلاقى مربع أحمر فيه نص مع مسافة 20 بكسل من فوق و 40 بكسل من اليسار.

ملخص سريع:

- **Padding** بين المحتوى وحدوده **Widget** بتحدد المسافة داخل الـ **Widget**.
- **EdgeInsets.all()** = مسافة متساوية من جميع الاتجاهات.
- **EdgeInsets.symmetric()** = مسافة متساوية أفقيًا أو عموديًا.
- **EdgeInsets.only()** = تحدد المسافة في اتجاهات معينة.

تحفيز: ابدأ صغير... واحلم كبير 🚀



Margin

✓ Margin إيه هي ؟

بس الفرق إن **Padding** زيها زي **Margin** الـ

- **Padding** = Widget المسافة داخل حدود الـ **Widget**.
- **Margin** = و اللي حواليه **Widget** يعني بين الـ **Widget** المسافة خارج حدود الـ **Widget**.

1. **EdgeInsets.all()**

بنستخدمها لمسافة متساوية من كل الاتجاهات، **Padding** زي ما عملنا في الـ

كود:

```
Container(  
  margin: EdgeInsets.all(16.0), // مسافة خارجية من كل الجهات  
  color: Colors.orange,  
  child: Text('Container خارجية خارجية'),  
)
```

Output:

- هتشوف مربع لونه برتقالي، وفيه نص، وعامل مسافة حوالية من كل الجهات.

2. **EdgeInsets.symmetric()**

بتدي مسافات خارجية بشكل أفقي (يمين وشمال) أو عمودي (فوق وتحت)

كود:

```
Container(  
  margin: EdgeInsets.symmetric(horizontal: 30.0, vertical: 10.0),  
  color: Colors.purple,  
  child: Text('Margin عمودي وأفقي'),  
)
```

Output:

- المربع بيكون عنده 30 بكسل من الجنبين، و 10 من فوق وتحت.

3. **EdgeInsets.only()**

بتستخدم لما تحب تعمل مسافة خارجية في اتجاهات معينة.

كود:

```
Container(  
  margin: EdgeInsets.only(bottom: 20.0, right: 40.0),  
  color: Colors.teal,
```

```
child: Text('Margin من تحت واليمين'),  
)
```

Output:

- المربع هيبقى لازق فوق وشمال، لكن فيه مسافة من تحت واليمين.

ملخص سريع:

- Margin** = Widget مسافة خارجية حوالين الـ
- Padding** = Widget مسافة داخلية بين المحتوى وحدود الـ
- برضو **margin** بتتطبق على الـ **EdgeInsets** نفس أنواع:
 - all** , **symmetric** , **only**

تحفيز: كل مطور كان مبتدئ، الفرق إنه كمل

Decoration

✓ **Decoration** ؟ يعني إيه

علشان تضيف له **Container** بتستخدمها جوا **decoration** الـ

- لون 🎨
- بوردر (حدود) ■
- زوايا دائرية ●
- ظل ✨
- خلفية بصورة 🖼️
- Gradient (تدرج لوني) 🌈

كل ده بنكتبه جوا **BoxDecoration**

1 (**color**) لون الخلفية

```
Container(  
  width: 150,  
  height: 80,  
  decoration: BoxDecoration(  
    color: Colors.amber,  
  ),  
  child: Center(child: Text('لون الخلفية')),  
)
```



Output:

مربع أصفر فيه كلمة "لون الخلفية" في النص.

2 (borderRadius) زوايا دائرية

```
Container(  
  width: 150,  
  height: 80,  
  decoration: BoxDecoration(  
    color: Colors.lightBlue,  
    borderRadius: BorderRadius.circular(20),  
  ),  
  child: Center(child: Text('زوايا ناعمة')),  
)
```



Output:

مربع أزرق فاتح بزوايا ناعمة مدورة.

3 (Borders) حدود

```
Container(  
  width: 150,  
  height: 80,  
  decoration: BoxDecoration(  
    color: Colors.white,
```

```
border: Border.all(
  color: Colors.red,
  width: 3,
),
),
child: Center(child: Text('مربع بحد أحمر')),
)
```



Output:

مربع أبيض حوالينه خط أحمر سمكه 3

4 ظل (Shadow)

```
Container(
  width: 150,
  height: 80,
  decoration: BoxDecoration(
    color: Colors.green,
    boxShadow: [
      BoxShadow(
        color: Colors.black.withOpacity(0.3),
        blurRadius: 8,
        offset: Offset(4, 4),
      ),
    ],
  ),
  child: Center(child: Text('فيه ظل')),
)
```



Output:

مربع أخضر وعليه ظل خفيف من تحت يعين

5 تدرج لوني (Gradient)


```
Container(  
  width: 150,  
  height: 80,  
  decoration: BoxDecoration(  
    gradient: LinearGradient(  
      colors: [Colors.pink, Colors.orange],  
    ),  
    borderRadius: BorderRadius.circular(12),  
  ),  
  child: Center(child: Text('تدرج ألوان 🔥')),  
)
```



Output:

مربع بلون بينك وأورانج متدرجين ناعمين مع زوايا مدورة



ملخص سريع:

المهمة `BoxDecoration` خصائص

- `color`: لون الخلفية
- `borderRadius`: زوايا مدورة
- `border`: حدود حوالين الـ Container
- `boxShadow`: ظل
- `gradient`: تدرج لوني

تحفيز: خليك فضولي، وجرب كل حاجة