

Flutter: Các widget cơ bản

Bùi Võ Quốc Bảo

Khoa CNTT | Trường CNTT-TT | Đại học Cần Thơ



Tài liệu tham khảo

- Chương 3-4-5, **Flutter Apprentice** by Vincenzo Guzzi, Kevin D Moore, Vincent Ngo and Michael Katz
- <https://docs.flutter.dev/development/ui/widgets-intro>
- <https://docs.flutter.dev/development/ui/layout>





Các widget cơ bản

(Hầu như) mọi thứ đều là widget!

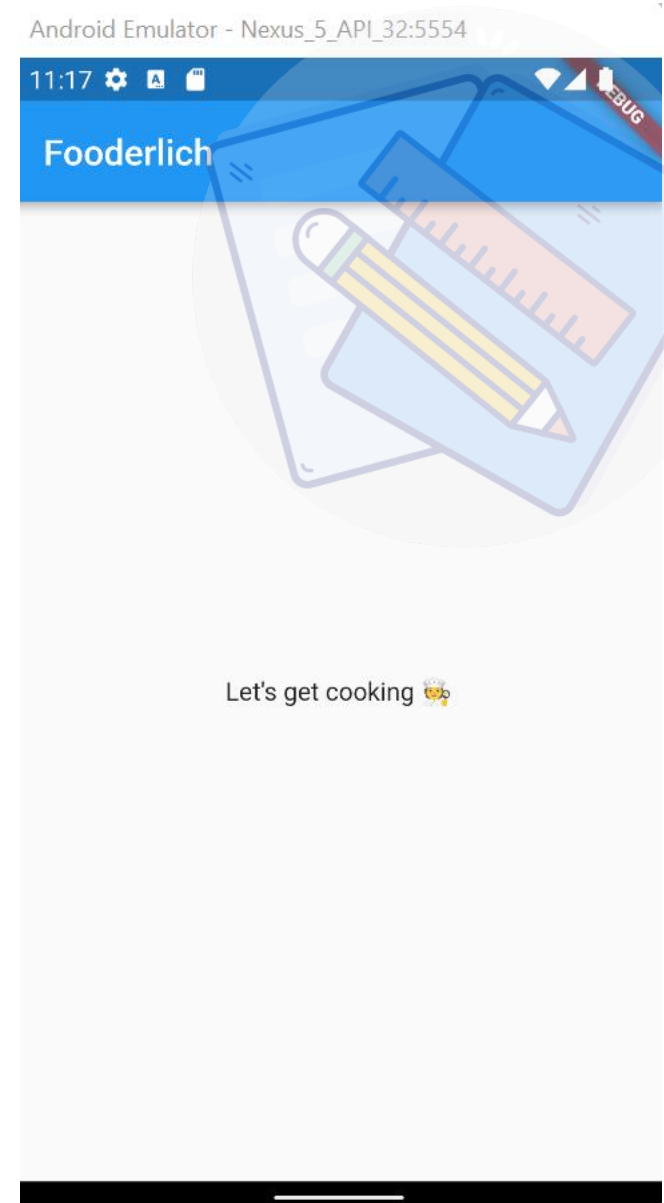
- Xét theo cách nhìn của nhà phát triển, hầu như mọi thứ trong Flutter đều là widget
- Các widget có thể được sử dụng cho:
 - Hiển thị thông tin (painting)
 - Bố cục / định vị trí (layout)
 - Nhận dạng cử chỉ (hit-testing)



Fooderlich

lib/main.dart

```
1  import 'package:flutter/material.dart';
2
3  Run | Debug | Profile
4  void main() {
5    | runApp(const Fooderlich());
6  }
7
8  class Fooderlich extends StatelessWidget {
9    | const Fooderlich({Key? key}) : super(key: key);
10
11  @override
12  Widget build(BuildContext context) {
13    | return MaterialApp(
14    |   title: 'Fooderlich',
15    |   home: Scaffold(
16    |     | appBar: AppBar(title: const Text('Fooderlich')),
17    |     | body: const Center(child: Text('Let\'s get cooking 🍳')),
18    |   ), // Scaffold
19  ); // MaterialApp
20 }
```

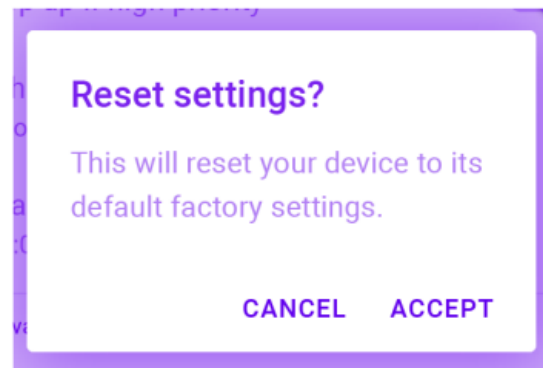


Các hệ thống thiết kế UI

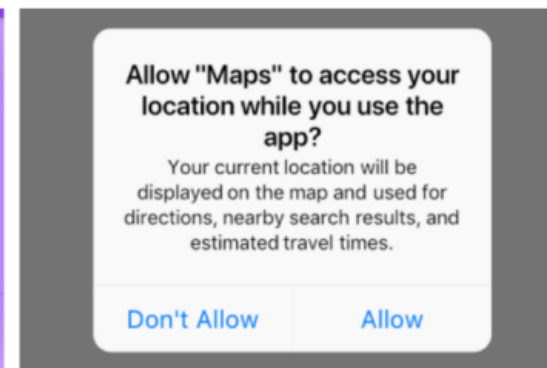
- Android sử dụng hệ thống **thiết kế Material**
 - import 'package:flutter/material.dart';
 - <https://docs.flutter.dev/development/ui/widgets/material>
- iOS sử dụng **hệ thống Cupertino**
 - import 'package:flutter/cupertino.dart';
 - <https://docs.flutter.dev/development/ui/widgets/cupertino>



Material Alert Dialog



Cupertino Alert Dialog



Định nghĩa theme cho ứng dụng Material

- Cần cung cấp một đối tượng [ThemeData](#) cho MaterialApp để định nghĩa theme cho toàn bộ ứng dụng
 - Theme mặc định sẽ được tạo nếu không cung cấp đối tượng ThemeData
- Có thể truy xuất theme trong phương thức build của các widget với [Theme.of\(context\)](#)
- Để tùy biến theme cho *một phần UI* của ứng dụng có thể sử dụng widget [Theme](#)
 - Tạo mới một đối tượng ThemeData hoặc
 - Mở rộng đối tượng ThemeData hiện có với .copyWith()

Định nghĩa theme cho ứng dụng Material

- Cần cung cấp một đối tượng [ThemeData](#) cho MaterialApp để định nghĩa theme cho toàn bộ ứng dụng, ví dụ:

```
MaterialApp(  
  title: appName,  
  theme: ThemeData(  
    // Define the default brightness and colors.  
    brightness: Brightness.dark,  
    primaryColor: Colors.lightBlue[800],  
  
    // Define the default font family.  
    fontFamily: 'Georgia',  
  
    // Define the default `TextTheme`. Use this to specify the default  
    // text styling for headlines, titles, bodies of text, and more.  
    textTheme: const TextTheme(  
      headline1: TextStyle(fontSize: 72.0, fontWeight: FontWeight.bold),  
      headline6: TextStyle(fontSize: 36.0, fontStyle: FontStyle.italic),  
      bodyText2: TextStyle(fontSize: 14.0, fontFamily: 'Hind'),  
    ),  
  ),  
  home: const MyHomePage(  
    title: appName,  
  ),  
);
```



Định nghĩa theme cho ứng dụng Material

- TextTheme: tạo theme cho các loại văn bản theo đặc tả của hệ thống thiết kế Material, chia làm 5 nhóm:
 - Display{Large/Medium/Small}
 - Headline{Large/Medium/Small}
 - Title{Large/Medium/Small}
 - Label{Large/Medium/Small}
 - Body{Large/Medium/Small}

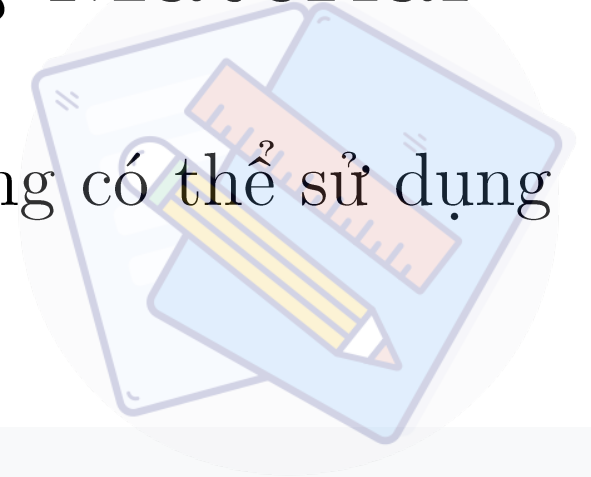


Định nghĩa theme cho ứng dụng Material

- Để tùy biến theme cho *một phần UI* của ứng dụng có thể sử dụng widget [Theme](#), ví dụ:

```
Theme(  
  // Create a unique theme with `ThemeData`  
  data: ThemeData(  
    splashColor: Colors.yellow,  
  ),  
  child: FloatingActionButton(  
    onPressed: () {},  
    child: const Icon(Icons.add),  
  ),  
);
```

```
Theme(  
  // Find and extend the parent theme using `copyWith`. See the next  
  // section for more info on `Theme.of`.  
  data: Theme.of(context).copyWith(splashColor: Colors.yellow),  
  child: const FloatingActionButton(  
    onPressed: null,  
    child: Icon(Icons.add),  
  ),  
);
```



Định nghĩa theme cho Fooderlich

pubspec.yaml

```
dependencies:  
  flutter:  
    sdk: flutter  
  cupertino_icons: ^1.0.2  
  google_fonts: ^2.3.2
```

Thêm gói thư viện [google_font](#)
vào dự án



```
// lib/fooderlich_theme.dart  
import 'package:flutter/material.dart';  
import 'package:google_fonts/google_fonts.dart';  
  
class FooderlichTheme {  
  static TextTheme lightTextTheme = TextTheme(  
    bodyText1: GoogleFonts.openSans(  
      fontSize: 14.0,  
      fontWeight: FontWeight.w700,  
      color: Colors.black,  
    ),  
    ...  
  );  
  
  static TextTheme darkTextTheme = TextTheme(  
    ...  
    headline1: GoogleFonts.openSans(  
      fontSize: 32.0,  
      fontWeight: FontWeight.bold,  
      color: Colors.white,  
    ),  
    ...  
  );  
  
  static ThemeData light() {
```

Định nghĩa theme cho Fooderlich

lib/fooderlich_theme.dart

```
static ThemeData light() {  
  return ThemeData(  
    brightness: Brightness.light,  
    ...  
    textTheme: lightTextTheme,  
  );  
}  
  
static ThemeData dark() {  
  return ThemeData(  
    brightness: Brightness.dark,  
    ...  
    textTheme: darkTextTheme,  
  );  
}
```

lib/main.dart

```
@override  
Widget build(BuildContext context) {  
  final theme = FooderlichTheme.dark();  
  return MaterialApp(  
    title: 'Fooderlich',  
    theme: theme,  
    home: Scaffold(  
      appBar: AppBar(  
        title: Text(  
          'Fooderlich',  
          style: theme.textTheme.headline6,  
        ), // Text  
      ), // AppBar  
      body: Center(  
        child: Text(  
          'Let\'s get cooking 🍳',  
          style: theme.textTheme.headline1,  
        ), // Text  
      ), // Center  
    ), // Scaffold  
  ); // MaterialApp  
}
```

