

# Flutter cǎn bǎn

...

# Yêu cầu công cụ:

- Cài đặt Flutter SDK
- Có trình soạn thảo Visual Studio Code
- Visual Studio Code đã cài đặt extensions: **Flutter**, **Dart**
- Cập nhật Flutter và Dart lên mới nhất

Flutter Stable: **v1.9.1+hotfix.6**

Dart Stable: **2.5.0**

Lưu ý: Cài **flutter** thành biến môi trường

# Tạo project Flutter

- Tạo project bằng câu lệnh:

```
flutter create <project_name>
```

- Tạo project trong Visual Studio Code:

Visual Studio Code -> View -> Command Palette ->  
Flutter: New Project -> Enter Project Name

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

- Left Sidebar (Explorer):** Displays the project structure. The **OPEN EDITORS** section shows `main.dart`. The **BUO11** folder contains files like `.idea`, `android`, `build`, `ios`, and `lib/main.dart` (which is currently selected). Other sections include `test`, `widget_test.dart`, `.gitignore`, `.metadata`, `.packages`, `buo11.iml`, `pubspec.lock`, `pubspec.yaml`, and `README.md`.
- Editor Area:** Shows the code for `main.dart`. The code defines the `main` function and the `MyApp` widget. The `MyApp` widget's `build` method returns a `MaterialApp` with a title of "Flutter Demo" and a theme defined by `ThemeData`. The `MyHomePage` class is also defined as a stateful widget.
- Bottom Bar:** Contains icons for `OUTLINE` and `DEPENDENCIES`.

```
main.dart ×
lib > main.dart > MyApp > build
1 import 'package:flutter/material.dart';
2
3 void main() => runApp(MyApp());
4
5 class MyApp extends StatelessWidget {
6     // This widget is the root of your application.
7     @override
8     Widget build(BuildContext context) {
9         return MaterialApp(
10             title: 'Flutter Demo',
11             theme: ThemeData([
12                 // This is the theme of your application.
13                 // Try running your application with "flutter run". You'll see the
14                 // application has a blue toolbar. Then, without quitting the app, try
15                 // changing the primarySwatch below to Colors.green and then invoke
16                 // "hot reload" (press "r" in the console where you ran "flutter run",
17                 // or simply save your changes to "hot reload" in a Flutter IDE).
18                 // Notice that the counter didn't reset back to zero; the application
19                 // is not restarted.
20                 primarySwatch: Colors.blue,
21             ], // ThemeData
22             home: MyHomePage(title: 'Flutter Demo Home Page'),
23         ); // MaterialApp
24     }
25 }
26
27
28 class MyHomePage extends StatefulWidget {
29     MyHomePage({Key key, this.title}) : super(key: key);
30
31     // This widget is the home page of your application. It is stateful, meaning
32     // that it has a State object (defined below) that contains fields that affect
33     // how it looks.
34
35     // This class is the configuration for the state. It holds the values (in this
36     // case the title) provided by the parent (in this case the App widget) and
37     // used by the build method of the State. Fields in a Widget subclass are
38     // always marked "final".
39
40     final String title;
41 }
```

# Chạy project Flutter

- Mở ít nhất một thiết bị máy ảo (simulator or emulator) hoặc kết nối thiết bị thật
- cd vào thư mục chứa project trong terminal
- Chạy dòng lệnh:

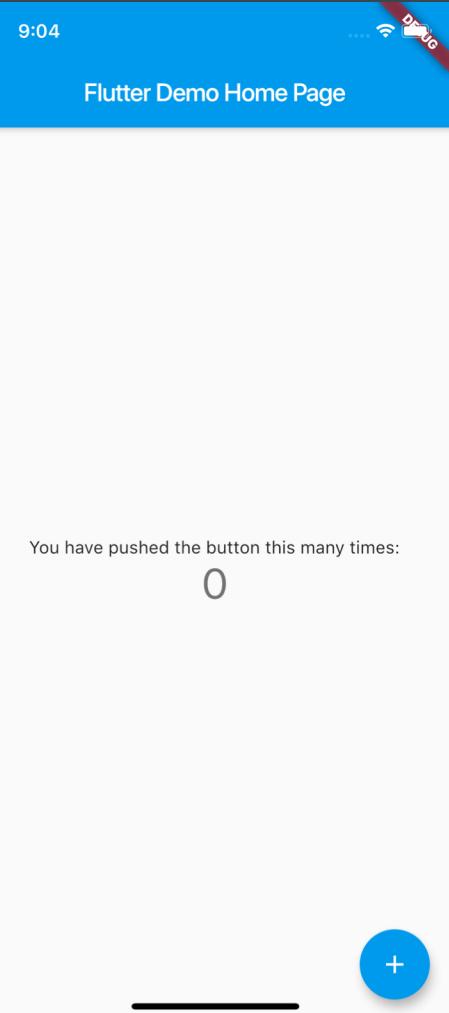
```
$ flutter run
```

# Chạy thành công

```
taof@Taofs-MacBook-Pro ~/Desktop/buoil
```

```
$ flutter run  
Running "flutter pub get" in buoil...                                0.4s  
Launching lib/main.dart on iPhone X in debug mode...  
Running Xcode build...  
  
└─ Assembling Flutter resources...                               6.3s  
  └─ Compiling, linking and signing...                         5.6s  
Xcode build done.                                              14.4s  
Syncing files to device iPhone X...                            1,618ms
```

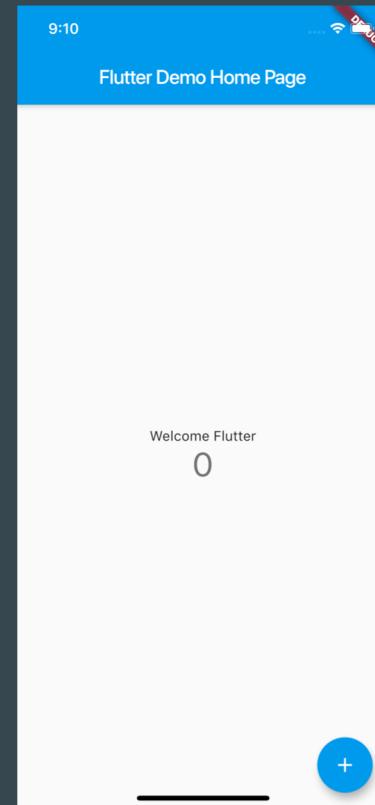
```
🔥 To hot reload changes while running, press "r". To hot restart (and rebuild state), press "R".  
An Observatory debugger and profiler on iPhone X is available at: http://127.0.0.1:50321/M6aVM05tm3o=/  
For a more detailed help message, press "h". To detach, press "d"; to quit, press "q".
```



# Flutter hot reload

```
lib/main.dart
lib/main.dart:_MyHomePageState> build
82    // Invoke "debug painting" (press "p" in the console, choose the
83    // "Toggle Debug Paint" action from the Flutter Inspector in Android
84    // Studio, or the "Toggle Debug Paint" command in Visual Studio Code)
85    // to see the wireframe for each widget.
86    //
87    // Column has various properties to control how it sizes itself and
88    // how it positions its children. Here we use mainAxisAlignment to
89    // center the children vertically; the main axis here is the vertical
90    // axis because Columns are vertical (the cross axis would be
91    // horizontal).
92    mainAxisAlignment: MainAxisAlignment.center,
93    children: <Widget>[
94        Text(
95            'You have pushed the button this many times:',
96        ), // text
97        Text(
98            '$_counter',
99            style: Theme.of(context).textTheme.display1,
100        ), // Text
101    ], // <Widget>[]
102    ), // Column
103    ), // Center
104    floatingActionButton: FloatingActionButton(
105        onPressed: _incrementCounter,
106        tooltip: 'Increment',
107        child: Icon(Icons.add),
108    ), // This trailing comma makes auto-formatting nicer for build methods. // FloatingActionButton
109    ); // Scaffold
110 }
111 }
```

Sửa text -> Welcome Flutter  
Sau đó từ terminal nhấn r



r: hot reload

R: hot restart

h: xem chi  
tiết

d: detach

q: quit

# Cấu trúc thư mục

```
my_app
|- android
| ...
|- ios
| ...
|- lib
| main.dart
|- test
pubspec.lock
pubspec.yaml
README.md
...
```

- File **main.dart** phải tồn tại và nó phải nằm trong thư mục gốc của lib. Đây là tệp mà Dart và Flutter biết để chạy vào ứng dụng.
- Tệp **pubspec.yaml**, đây là nơi liệt kê các dependencies trong dự án.

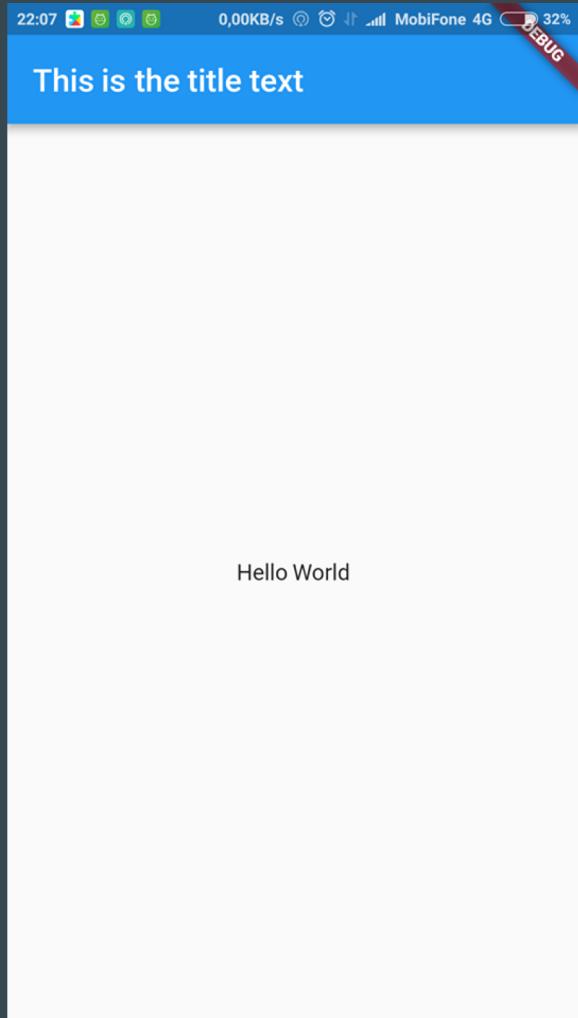
```
// import the Flutter sdk
import 'package:flutter/material.dart';

// Every Dart program ever must have a main() function
// runApp is a Flutter function that runs the app.
// It takes a Widget as an argument.
void main() => runApp(new MyApp());

// EVERYTHING is a Widget. Including the root of your entire App:
class MyApp extends StatelessWidget {
    ...
    // NB: MyApp is an arbitrary name.
}
```

# Sample Project 1

```
import 'package:flutter/material.dart';
void main() => runApp(new MyApp());
class MyApp extends StatelessWidget {
@override
Widget build(BuildContext context) {
    return new MaterialApp(
        title: 'Welcome to Flutter',
        home: new Scaffold(
            appBar: new AppBar(
                title: new Text('This is the title text'),
            ),
            body: new Center(
                child: new Text('Hello World'),
            ),
        ),
    );
}
}
```



<pre>void main() =&gt; runApp(new MyApp());</pre>	<p>Là phương thức gọi để <b>run</b> ứng dụng</p>
<pre>@override Widget build(BuildContext context) { }</pre>	<p>Phương thức <b>build()</b> mô tả cách hiển thị một widget. Flutter gọi build() khi nó chèn một widget vào widget hierarchy</p>
<pre>return new MaterialApp()</pre>	<p>Khởi tạo để áp dụng thiết kế <b>Material design</b> cho ứng dụng</p>
<pre>home: new Scaffold()</pre>	<p>Đối số <b>home</b> tham chiếu đến widget định nghĩa main UI, trong trường hợp này là <b>Scaffold</b>.</p>
<pre>appBar: new AppBar(     title: new Text('This is the title text'), )</pre>	<p>Tạo title cho app bar</p>
<pre>body: new Center(     child: new Text('Hello World') )</pre>	<p>Trong scaffold có thuộc tính là body đây là <b>primary content</b> của scaffold</p>

# Widget

# Widget là gì?

- Flutter đã tạo ra một khái niệm "mới" đó là **widget**.  
Widget chính là nền tảng của Flutter, một widget miêu tả một phần của giao diện người dùng.
- Trong Flutter, mọi thứ đều là **widget**. Tất cả các component như text, image, button hay animation, theme, layout hay thậm chí app cũng là 1 widget

# Widget cơ bản

**Text** - Một tiện ích được sử dụng để chỉ đơn giản là văn bản trên màn hình.

**Image** - Để hiển thị hình ảnh.

**Icon** - Để hiển thị các biểu tượng

**Container** - div của Flutter UI. Đây là một widget cho phép căn chỉnh, hình nền, kích thước. Nó cũng chiếm không gian 0px khi trống.

**TextInput** - Để xử lý phản hồi nhập liệu từ người dùng.

**Row, Column** - Widget hiển thị danh sách children theo hướng ngang hoặc dọc. Chúng tuân theo quy tắc hộp flex để bố trí.

**Stack** - Một ngăn xếp hiển thị một danh sách children chồng lên nhau (giống bánh kếp)

**Scaffold** - Đây là thư mục gốc của mọi màn hình trong ứng dụng, giúp ứng dụng có cấu trúc bố cục cơ bản.

# Text và các thuộc tính hay sử dụng

- `data: "",` nội dung văn bản
- `style: TextStyle(color, fontSize, textAlign, textDirection)`
- `maxLines:` số dòng tối đa có thể hiển thị
- ...
- Ví dụ:

```
Text('Theme',  
      style: TextStyle(  
          color: Color.white,  
          fontSize: 32  
      ),  
)
```

# Container và thuộc tính

- Widget container có nhiều thuộc tính như: `alignment`, `padding`, `color`, `decoration`, `width`, `height`, `margin`, `transform`, `child`.
- Ví dụ:

```
Container(  
    color: Color.red,  
    child: Text(  
        'Theme',  
        style: TextStyle(  
            color: Color.teal,  
            fontSize: 32  
        ),  
    ),  
)
```

# Center và thuộc tính

- Center có các thuộc tính widthFactor, heightFactor, child
- Ví dụ:

```
Center(  
    child: Text("Element 1")  
)
```

# Icon và thuộc tính

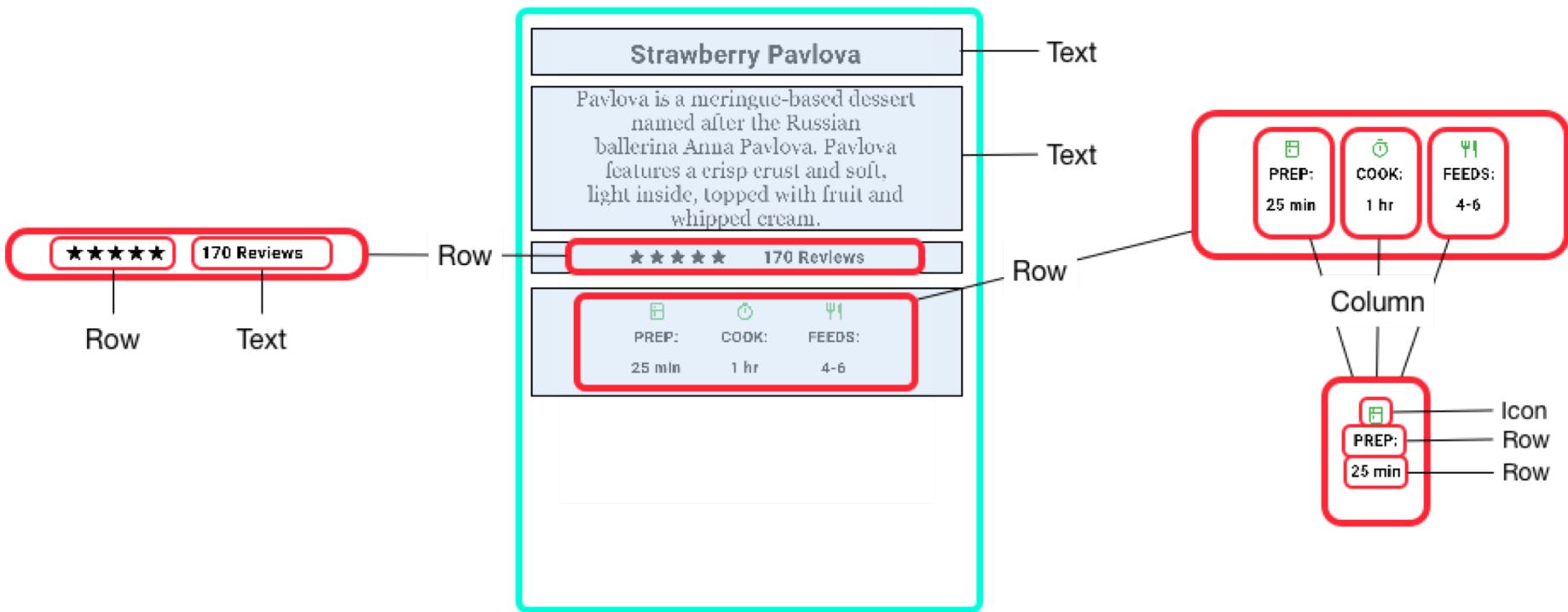
- Icon có các thuộc tính như `icon`, `size`, `color`,  
`textDirection`, `semanticLabel`
- Ví dụ:

```
Icon(Icons.add_to_queue, color: Colors.white,)
```

# Row và column

- Row và column là 2 widget được sử dụng khá phổ biến, chúng ta có thể chia giao diện thành các hàng, các cột để layout dễ dàng.

## Left Column



- Row và Column đều có các thuộc tính như:  
mainAxisAlignment, crossAxisAlignment, textDirection,  
verticalDirection, textBaseline, children
- Ví dụ:

```
Row(  
    mainAxisAlignment: MainAxisAlignment.min,  
    children: [  
        Icon(Icons.star, color: Colors.green[500]),  
        Icon(Icons.star, color: Colors.green[500]),  
        Icon(Icons.star, color: Colors.green[500]),  
        Icon(Icons.star, color: Colors.black),  
        Icon(Icons.star, color: Colors.black),  
    ],  
)
```

# Bài tập:

