

# Flutter: Quản lý trạng thái

Bùi Võ Quốc Bảo

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



# Tài liệu tham khảo

- Chương 6-13, **Flutter Apprentice** by Vincenzo Guzzi, Kevin D Moore, Vincent Ngo and Michael Katz
- <https://suragch.medium.com/flutter-state-management-for-minimalists-4c71a2f2f0c1>
- <https://docs.flutter.dev/development/data-and-backend/state-mgmt/declarative>
- <https://docs.flutter.dev/cookbook/forms>



# Xây dựng UI dạng khai báo

- Flutter cho phép nhà phát triển tạo UI cho ứng dụng *dạng khai báo (declarative)*: Flutter xây dựng UI để phản chiếu trạng thái của ứng dụng


$$\text{UI} = f(\text{state})$$

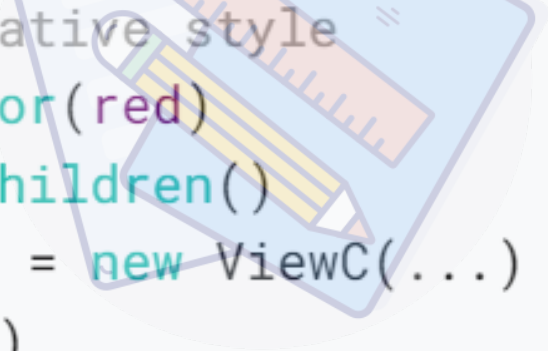
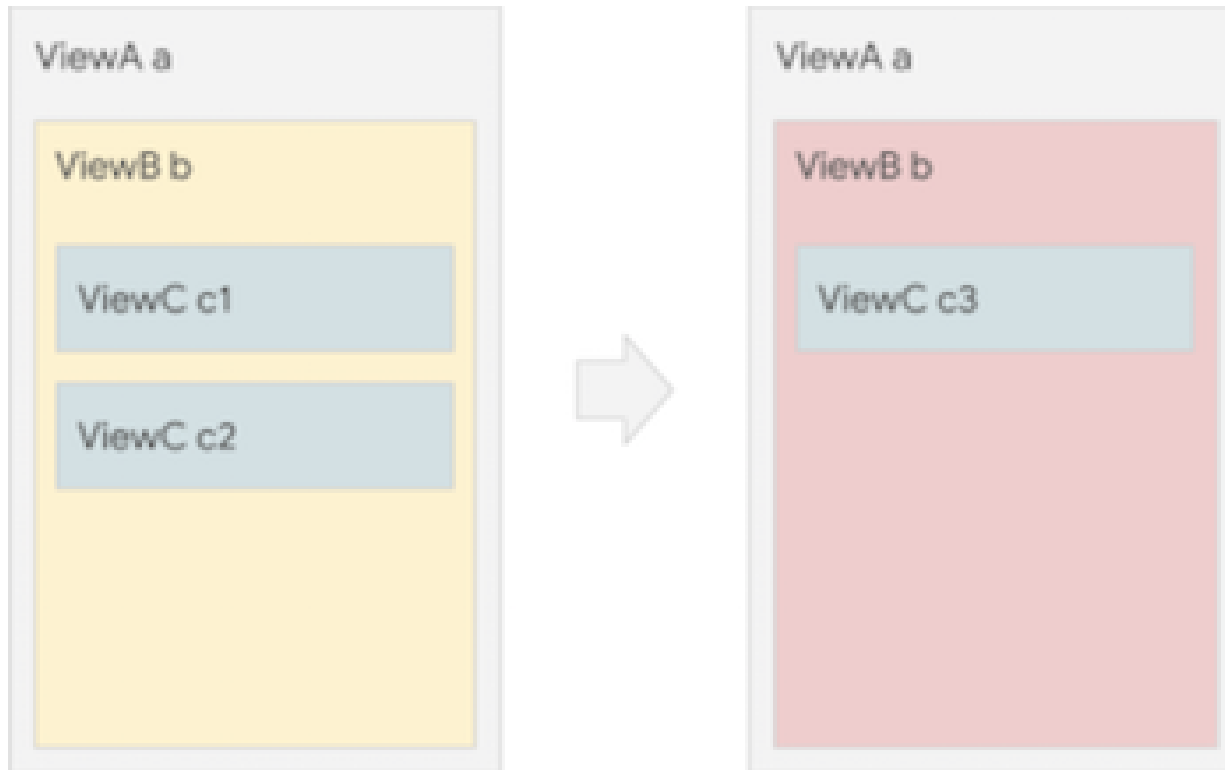
The layout on the screen

Your build methods

The application state

- Trạng thái ứng dụng thay đổi kích hoạt việc vẽ lại/tái tạo lại UI

# Xây dựng UI dạng khai báo



```
// Imperative style
b.setColor(red)
b.clearChildren()
ViewC c3 = new ViewC(...)
b.add(c3)
```

```
// Declarative style
return ViewB(
    color: red,
    child: ViewC(...),
)
```

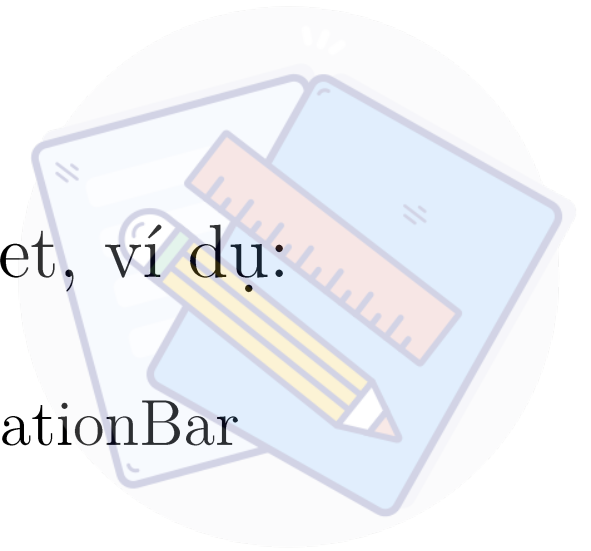
# Trạng thái ứng dụng

- Theo nghĩa rộng: trạng thái ứng dụng là mọi thứ tồn tại trong bộ nhớ khi ứng dụng chạy
- Định nghĩa hữu ích hơn cho nhà phát triển: *bất kỳ dữ liệu nào cần thiết để xây dựng lại UI*
- Có thể được chia thành hai loại:
  - Trạng thái cục bộ (local state/UI state/ephemeral state)
  - Trạng thái chia sẻ (shared state/app state)



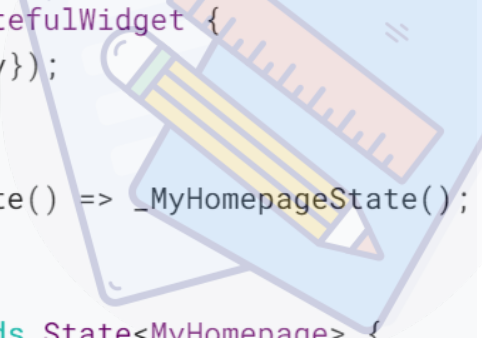
# Trạng thái cục bộ

- Trạng thái có thể được chứa gọn trong một widget, ví dụ:
  - Tiến trình hiện tại của một hoạt ảnh phức tạp
  - Thẻ đang được chọn hiện tại trong một BottomNavigationBar
  - Trang hiện thời trong một PageView, ...
- Các phần khác trong cây widget ít khi cần đến trạng thái này
- Chỉ cần sử dụng một **StatefulWidget** để quản lý



# Trạng thái cục bộ

- Trong ví dụ này, **\_\_index** là trạng thái cục bộ
- Các phần khác trong ứng dụng không truy cập **\_\_index**
- Giá trị **\_\_index** chỉ thay đổi bên trong MyHomePage
- Không quan tâm **\_\_index = 0** khi ứng dụng được đóng và khởi động lại



```
class MyHomepage extends StatefulWidget {  
  const MyHomepage({super.key});  
  
  @override  
  _MyHomepageState createState() => _MyHomepageState();  
}  
  
class _MyHomepageState extends State<MyHomepage> {  
  int _index = 0;  
  
  @override  
  Widget build(BuildContext context) {  
    return BottomNavigationBar(  
      currentIndex: _index,  
      onTap: (newIndex) {  
        setState(() {  
          _index = newIndex;  
        });  
      },  
      // ... items ...  
    );  
  }  
}
```

# Trạng thái chia sẻ

- Còn gọi là trạng thái ứng dụng (app state)
- Trạng thái được sử dụng tại nhiều nơi trong ứng dụng và muốn được lưu giữ giữa các phiên làm việc
  - Thông tin thiết lập của người dùng (user preferences)
  - Thông tin đăng nhập
  - Trạng thái đọc/chưa đọc của các bài đăng trong một ứng dụng tin tức
  - Các thông báo trong một ứng dụng mạng xã hội
  - Giỏ hàng mua sắm trong một ứng dụng thương mại điện tử, ...





# Trạng thái chia sẻ

- Có nhiều giải pháp quản lý tùy thuộc ứng dụng
  - Trong trường hợp đơn giản, widget cha có thể gửi trạng thái, callback cập nhật trạng thái (kết hợp với setState) xuống cho widget con
  - Lựa chọn giải pháp phù hợp và giúp cho mã nguồn đơn giản, dễ hiểu

