

Lập trình Mobile



Tuần 5

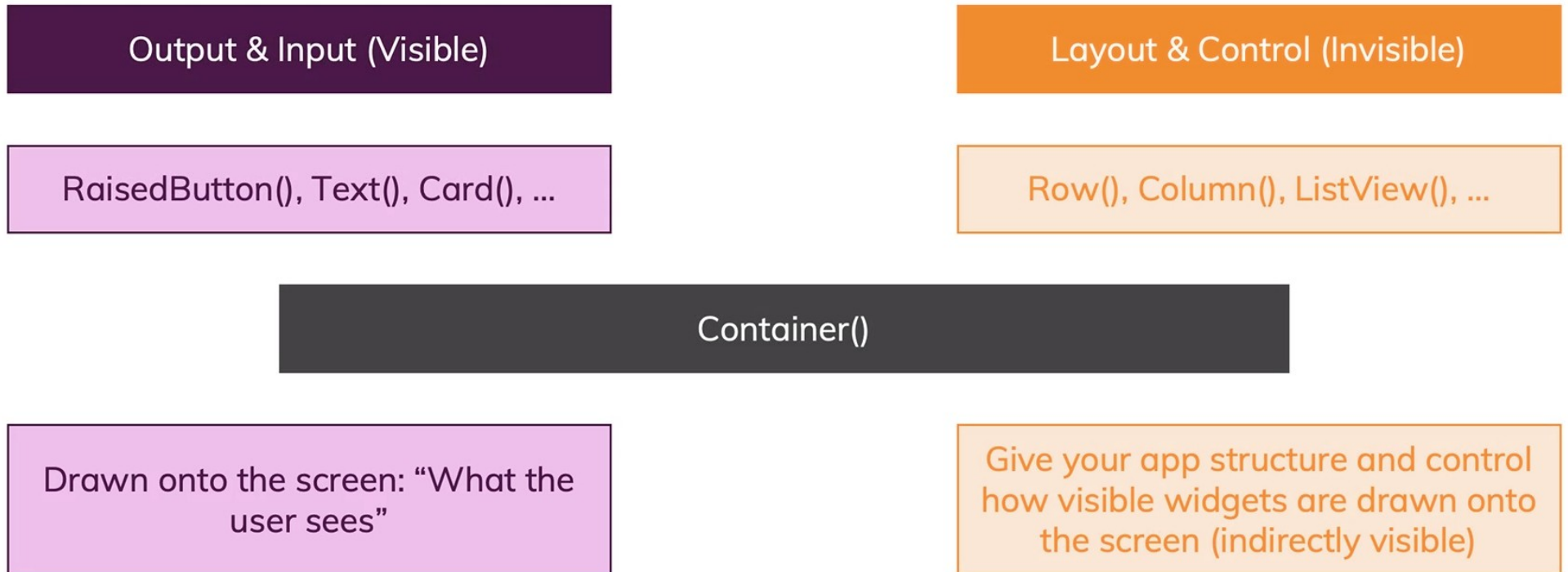
Giảng viên: Trần Đức Minh

Nội dung bài giảng



- Các loại Widget
- Tạo Layout cho Widget
-

Các loại Widget



- Container() thuộc cả hai dạng do mặc định nó là invisible nhưng ta có thể định dạng để có thể thấy nó.

Tạo Layout cho Widget

- Ví dụ:
 - Sử dụng Widget Column()

```
Widget build(BuildContext context) {  
  return MaterialApp(  
    title: 'Material App',  
    home: Scaffold(  
      body: Column(  
        children: [  
          ElevatedButton(  
            style: ElevatedButton.styleFrom(  
              primary: Colors.red, // background  
              onPrimary: Colors.yellow, // foreground  
            ),  
            onPressed: () {},  
            child: const Text('Nút nhấn thứ 1'),  
          ), // ElevatedButton  
          const Text(  
            'Dòng thứ 1',  
            style: TextStyle(  
              fontSize: 28  
            ), // TextStyle  
            textAlign: TextAlign.center,  
          ), // Text  
          ElevatedButton(  
            style: ElevatedButton.styleFrom(  
              primary: Colors.red, // background  
              onPrimary: Colors.yellow, // foreground  
            ),  
            onPressed: null,  
            child: const Text('Nút nhấn thứ 2'),  
          ), // ElevatedButton  
        ],  
      ), // Column  
    ), // Scaffold  
  ); // MaterialApp  
}
```

Tạo Layout cho Widget

- Ví dụ:
 - Sử dụng Widget Row()

```
Widget build(BuildContext context) {  
  return MaterialApp(  
    title: 'Material App',  
    home: Scaffold(  
      body: Row(  
        children: [  
          ElevatedButton(  
            style: ElevatedButton.styleFrom(  
              primary: Colors.red, // background  
              onPrimary: Colors.yellow, // foreground  
            ),  
            onPressed: () {},  
            child: const Text('Nút nhấn thứ 1'),  
          ), // ElevatedButton  
          const Text(  
            'Cột thứ 2',  
            style: TextStyle(  
              fontSize: 28  
            ), // TextStyle  
            textAlign: TextAlign.center,  
          ), // Text  
          ElevatedButton(  
            style: ElevatedButton.styleFrom(  
              primary: Colors.red, // background  
              onPrimary: Colors.yellow, // foreground  
            ),  
            onPressed: null,  
            child: const Text('Nút nhấn thứ 2'),  
          ), // ElevatedButton  
        ],  
      ), // Row  
    ), // Scaffold  
  ); // MaterialApp  
}
```

Xử lý sự kiện nhấn ElevatedButton



- Thuộc tính onPressed
 - Có kiểu dữ liệu là **VoidCallback**.
 - VoidCallback là kiểu đại diện cho các hàm callback không có tham số và không có dữ liệu trả về. Hàm callback này sẽ được gọi khi Button được nhấn.
 - Nếu thuộc tính này là **null** thì Button sẽ bị vô hiệu hóa (disable)
 - Ví dụ:

```
onPressed: () {  
    print('Button vừa được nhấn');  
    ...  
},
```

hoặc

```
onPressed: () => print('Button vừa được nhấn'),
```

Cập nhật dữ liệu



- Phương thức build của lớp kế thừa từ lớp StatelessWidget sẽ chỉ chạy 1 lần duy nhất. Phương thức build không thể chạy lại mỗi khi thuộc tính bị thay đổi.

Hiểu về State (Trạng thái)



In General

State is Data/ Information used by your App

App State

Authenticated Users
Loaded Jobs

...

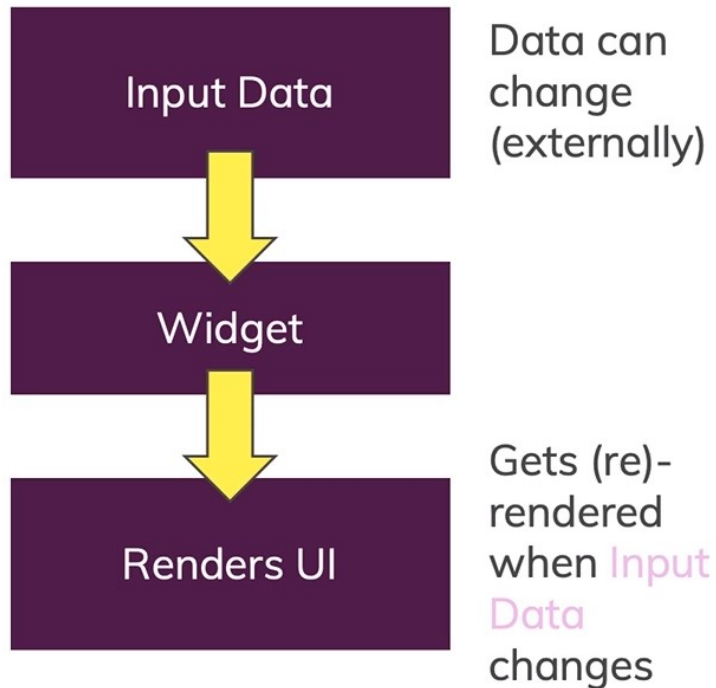
Widget State

Current User Input
Is a Loading Spinner being shown?

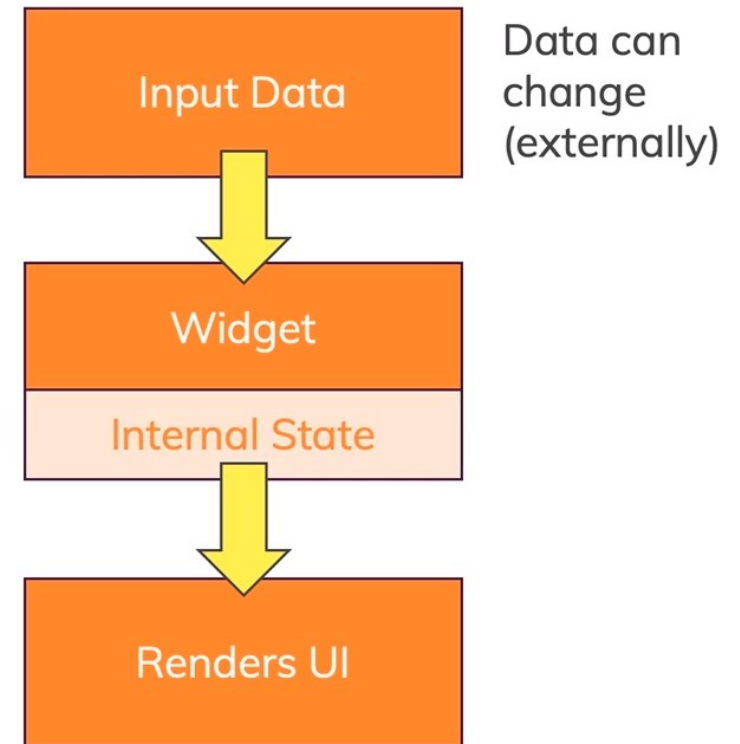
Stateless và Stateful



Stateless



Stateful



Cấu trúc chung khi xây dựng Stateful



```
class MyApp extends StatefulWidget {  
  @override  
  State<StatefulWidget> createState() {  
    return MyAppState();  
  }  
}
```

```
class MyAppState extends State<MyApp> {  
  Widget build(BuildContext build) {  
    return MaterialApp(  
      home: Scaffold(  
        body: <Widget Tree>  
      ),  
    );  
  }  
}
```

Hàm setState



- Hàm setState được dùng để chạy lại phương thức build()
- Ví dụ:

```
setState() {  
    if (_questionIndex < 2) _questionIndex++;  
    else _questionIndex = 0;  
});
```

Hết Tuần 5



Cảm ơn các bạn đã chú ý lắng nghe !!!