

# State in Flutter

ทำความเข้าใจกับ state ใน widget และ widget ที่เกี่ยวข้อง

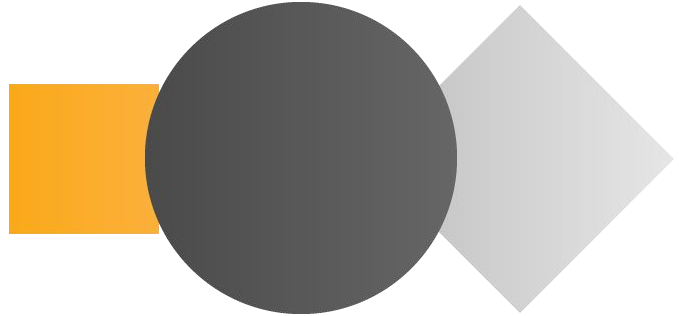


## About this section

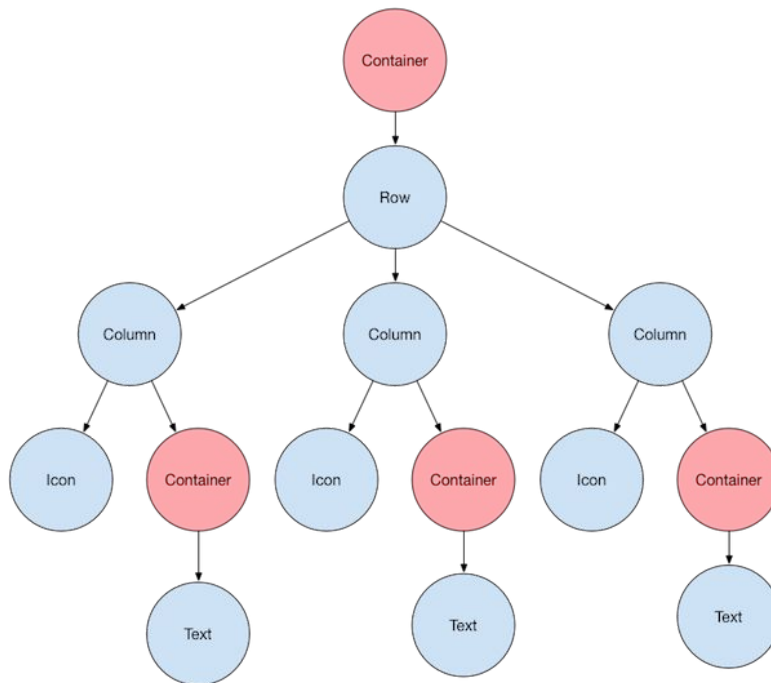
- Flutter Widget Tree
- What is State in Flutter
- Stateful Widget
- Stateless Widget
- Inherited Widget



# Widget Tree



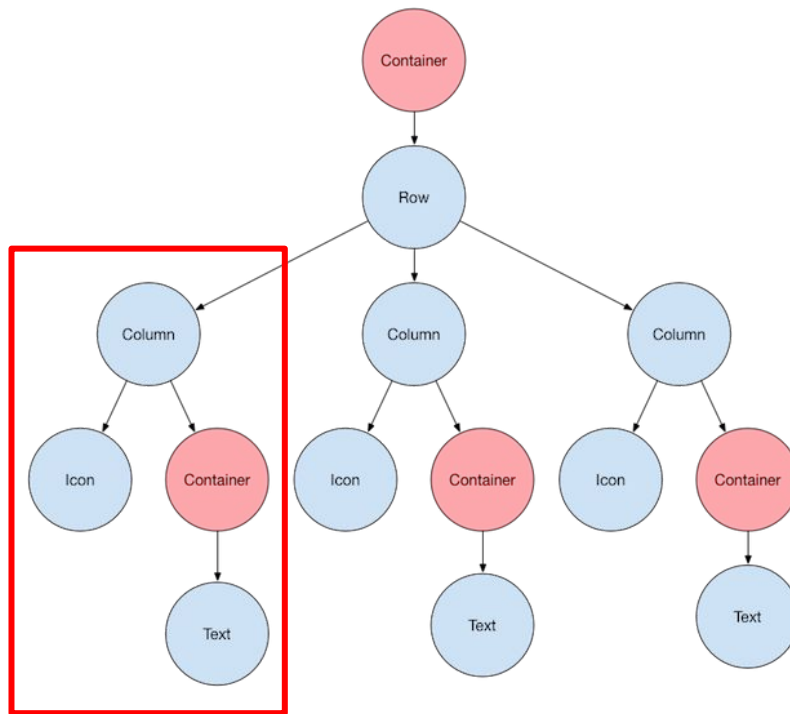
# Widget Hierarchy in Flutter



หน้าตาของโค้ดที่เขียน หลังจากถูกแปลงเป็น tree



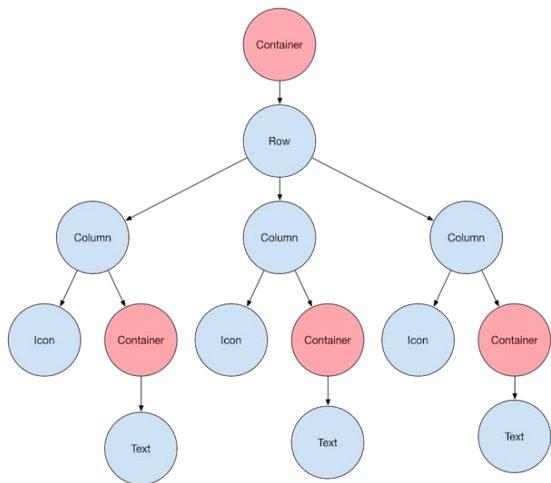
# Widget Hierarchy in Flutter



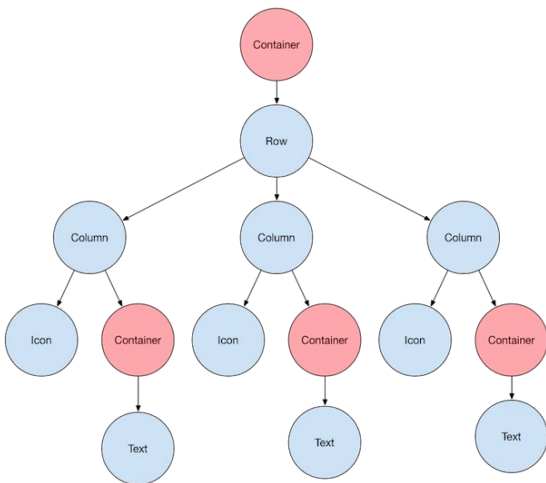
Flutter จะรู้ได้ยังไงว่าส่วนไหนต้องเปลี่ยน ต้องวาดใหม่



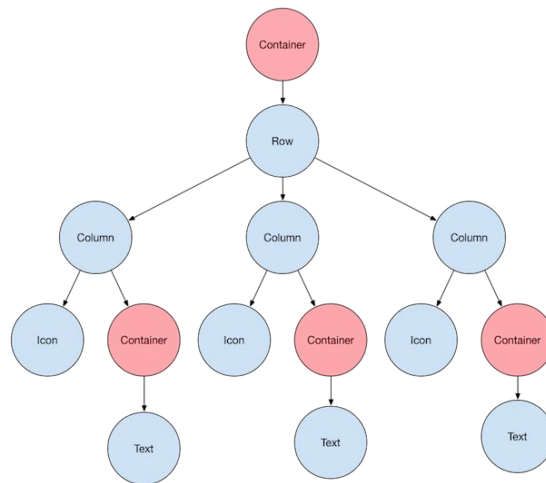
# Widget Hierarchy in Flutter



**Widget tree**



**Element tree**

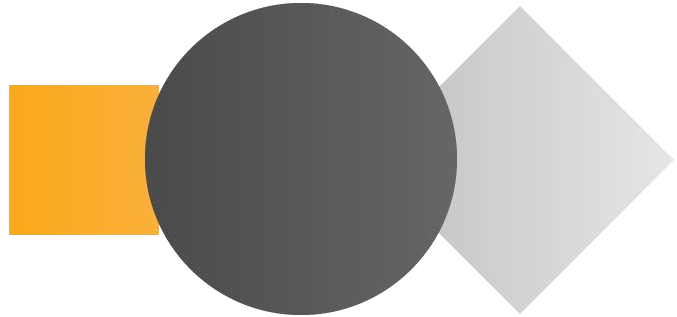


**RenderObject tree**

**Widget -> Configuration ของ UI, Element -> จัดการ State, RenderObject -> Sizing, painting**

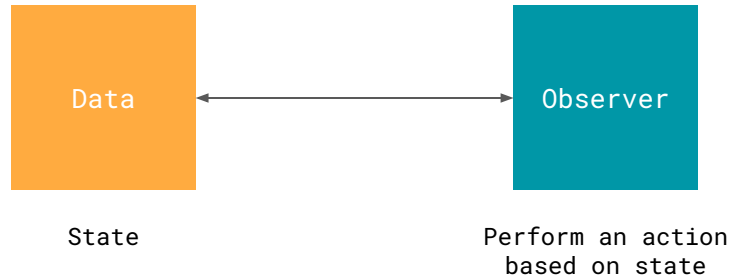


# State



# State

*(noun) the particular condition that someone or something is in at a specific time.*







# Stateful Widgets

## Implicit State

มี state เป็นของตัวเอง

## Mutable State

สามารถเปลี่ยนแปลง state ได้ผ่าน setState

## Support Caching

ถ้า state ไม่เปลี่ยน = ไม่ rebuild



# Code Example

GROW

GROW

GROW

```
class StatefulBox extends StatefulWidget {  
  const StatefulBox({  
    Key? key,  
  }) : super(key: key);  
  
  @override  
  _StatefulBoxState createState() => _StatefulBoxState();  
}  
  
class _StatefulBoxState extends State<StatefulBox> {  
  double _size = 1.0; //state name _size  
  
  void grow() {  
    setState(() { _size += 0.1; }); //mutate state  
  }  
  
  @override  
  Widget build(BuildContext context) {  
    return Container(  
      transform: Matrix4.diagonal3Values(_size, _size, 1.0),  
      child: ElevatedButton(onPressed: () { grow(); } , child:  
Text("GROW")),  
    );  
  }  
}
```





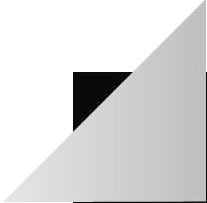
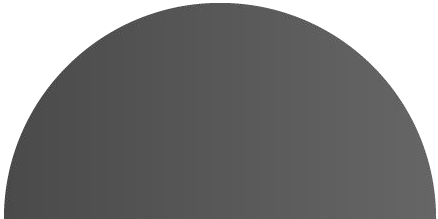
# Stateless Widgets

## Explicit State

ไม่มี state ในตัวเอง, ใช้ค่าที่ได้รับจาก  
parent/configuration

## Immutable State

ตัวเองไม่สามารถเปลี่ยน state ได้ ต้องให้  
parent จัดการ



# Code Example



```
class Box extends StatelessWidget {  
  const Box({  
    Key? key,  
    this.color = const Color(0xFF2DBD3A),  
    this.child,  
  }) : super(key: key);  
  
  final Color color;  
  final Widget? child;  
  
  @override  
  Widget build(BuildContext context) {  
    return Container(color: color, child: child);  
  }  
}
```





# Build Context

- Handle a location of widget in widget tree
- Can be used to locate Ancestor widget

```
//StatelessWidget.build, State.build  
@override  
Widget build(BuildContext context) {  
    return ...  
}
```





# Build Context

```
class ChildWidget extends StatelessWidget {  
  @override  
  Widget build(BuildContext context){  
    final ParentWidget widget = context.ancestorWidgetOfExactType(ParentWidget);  
    final ParentWidgetState parentState = widget?.someState;  
  
    ...  
  }  
}
```

**Navigator.of(context)...**

**MediaQuery.of(context)...**

**Theme.of(context)...**

context.ancestorWidgetOfExactType





# Inherited Widgets

Propagated information

Works within same widget tree



# Code Example

Hello World

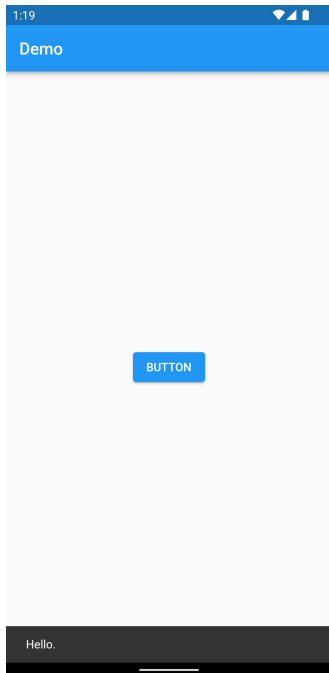
context.dependOnInheritedWidgetOfExactType

```
const ColorProvider({  
  Key? key,  
  required Widget child,  
}) : super(key: key, child: child);  
  
final Color color = Colors.blue;  
  
static ColorProvider of(BuildContext context) {  
  final ColorProvider? result =  
    context.dependOnInheritedWidgetOfExactType<ColorProvider>();  
  assert(result != null, 'No ColorProvider found in context');  
  return result!;  
}  
  
@override  
bool updateShouldNotify(ColorProvider old) => color != old.color;  
}  
  
class MyPage extends StatelessWidget {  
  const MyPage({Key? key}) : super(key: key);  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      body: ColorProvider(  
        child: Builder(  
          builder: (BuildContext innerContext) {  
            return Center(  
              child: Text(  
                'Hello World',  
                style: TextStyle(color: ColorProvider.of(innerContext).color),  
              ),  
            );  
          },  
        ),  
      ),  
    );  
  }  
}
```





# Scaffold + Snackbar



```
class _MyStatefulWidgetState extends State<MyStatefulWidget> {  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(title: const Text('Demo')),  
      body: Center(  
        child: ElevatedButton(  
          child: const Text('BUTTON'),  
          onPressed: () {  
            ScaffoldMessenger.of(context).showSnackBar(const  
Snackbar(  
          content: Text('Hello.')  
        ));  
      }),  
    ),  
  );  
}
```



# State in Flutter

