**首先来张图**

**代码打印这些方法在何时执行**

class \_LifeDemoState extends State<LifeDemo>

with WidgetsBindingObserver {

@override

void initState() {

// TODO: implement initState

super.initState();

WidgetsBinding.instance.addObserver(this);

print('initState');

}

@override

void didChangeAppLifecycleState(AppLifecycleState state) {

print(state.toString());

}

@override

void didChangeDependencies() {

// TODO: implement didChangeDependencies

super.didChangeDependencies();

print('didChangeDependencies');

}

@override

void didUpdateWidget(LifeDemo oldWidget) {

// TODO: implement didUpdateWidget

super.didUpdateWidget(oldWidget);

print('didUpdateWidget');

}

@override

Widget build(BuildContext context) {

print('build');

// TODO: implement build

return MaterialApp(

home: Center(

child: GestureDetector(

child: new Text('lifeCycle'),

onTap: () {

Navigator.of(context)

.push(new MaterialPageRoute(builder: (BuildContext c) {

return new Text('sdfs');

}));

},

)),

);

}

@override

void reassemble() {

// TODO: implement reassemble

super.reassemble();

print('reassemble');

}

@override

void deactivate() {

// TODO: implement deactivate

super.deactivate();

print('deactivate');

}

@override

void dispose() {

// TODO: implement dispose

super.dispose();

WidgetsBinding.instance.addObserver(this);

print('dispose');

}

}

**1、创建一个wedget到显示 打印结果如下**

01-15 14:06:44.751 19005-19025/com.example.flutterapp I/flutter: initState

01-15 14:06:44.751 19005-19025/com.example.flutterapp I/flutter: didChangeDependencies

01-15 14:06:44.752 19005-19025/com.example.flutterapp I/flutter: build

**2、退出这个页面执行**

01-15 14:08:40.956 19005-19561/com.example.flutterapp I/flutter: deactivate

01-15 14:08:40.970 19005-19561/com.example.flutterapp I/flutter: dispose

**3、点击热重载按钮**

01-15 14:09:55.467 19005-19561/com.example.flutterapp I/flutter: reassemble

01-15 14:09:55.476 19005-19561/com.example.flutterapp I/flutter: didUpdateWidget

01-15 14:09:55.476 19005-19561/com.example.flutterapp I/flutter: build

**4、app由显示切换到后台（home状态）**

01-15 14:11:10.062 19005-19561/com.example.flutterapp I/flutter: AppLifecycleState.inactive

01-15 14:11:10.524 19005-19561/com.example.flutterapp I/flutter: AppLifecycleState.paused

**5、app由后台切回前台**

01-15 14:11:54.836 19005-19561/com.example.flutterapp I/flutter: AppLifecycleState.inactive

01-15 14:11:54.837 19005-19561/com.example.flutterapp I/flutter: AppLifecycleState.resumed

各个方法的解释：  
**initState**：widget创建执行的第一个方法，可以再里面初始化一些数据，以及绑定控制器 如下代码所示

@override

void initState() {

// TODO: implement initState

super.initState();

WidgetsBinding.instance.addObserver(this);

\_scrollController=new ScrollController();

print('initState');

}

**didChangeDependencies**：当State对象的依赖发生变化时会被调用；例如：在之前build() 中包含了一个InheritedWidget，然后在之后的build() 中InheritedWidget发生了变化，那么此时InheritedWidget的子widget的didChangeDependencies()回调都会被调用。InheritedWidget这个widget可以由父控件向子控件共享数据，案例可以参考 scoped\_model开源库。

**build** ：它主要是用于构建Widget子树的。

**reassemble**：此回调是专门为了开发调试而提供的，在热重载(hot reload)时会被调用，此回调在Release模式下永远不会被调用。

**didUpdateWidget**：当树rebuid的时候会调用该方法。

**deactivate**:当State对象从树中被移除时，会调用此回调。

**dispose()**:当State对象从树中被永久移除时调用；通常在此回调中释放资源。

**注：**

代码中的**didChangeAppLifecycleState**方法复写需要State with WidgetsBindingObserver这个抽象类

作者：冒泡人生  
链接：https://www.jianshu.com/p/f39cf2f7ad78  
来源：简书  
简书著作权归作者所有，任何形式的转载都请联系作者获得授权并注明出处。