

03

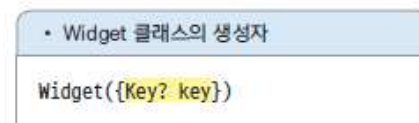
위젯 키

UI Architecture

위젯 키

위젯을 식별하는 키

- 모든 위젯은 키값을 가질 수 있습니다.



- StatelessWidget일 때 식별하기 — 키 미사용
 - StatelessWidget은 상태를 표현할 수 없으며 화면에 표시할 데이터를 위젯이 직접 가지고 있으므로 키로 식별하지 않아도 객체를 이용하는 데 문제가 없습니다.

위젯 키

• StatelessWidget 위젯

```
class MyColorItemWidget extends StatelessWidget {
  Color color;
  MyColorItemWidget(this.color);
  @override
  Widget build(BuildContext context) {
    return Expanded(
      child: Container(
        color: color,
        width: 150,
        height: 150,
      ),
    );
  }
}
```

• 순서 바꾸기

```
class _MyListWidgetState extends State<MyListWidget> {
  List<Widget> widgetList = [
    MyColorItemWidget(Colors.red),
    MyColorItemWidget(Colors.blue),
  ];
  onChange() {
    setState(() {
      widgetList.insert(1, widgetList.removeAt(0));
    });
  }
  @override
  Widget build(BuildContext context) {
    print('print.... ${widgetList.length}');
    return Scaffold(
      appBar: AppBar(title: Text('Key Test')),
      body: Column(
        children: [
          Row(children: widgetList),
          ElevatedButton(onPressed: onChange, child: Text("toggle"))
        ]
      ),
    );
  }
}
```

위젯 키

▶ 실행 결과



그림 8-11 StatelessWidget 테스트

위젯 키

- 다른 타입의 StatefulWidget 식별하기 — 키 미사용

• 다른 타입의 객체 이용

```
class _MyListWidgetState extends State<MyListWidget> {  
  List<Widget> widgetList = [  
    MyREDItemWidget(),  
    MyBLUEItemWidget(),  
  ];  
  onChange() {  
    setState(() {  
      widgetList.insert(1, widgetList.removeAt(0));  
    });  
  }  
  ... (생략) ...  
}
```

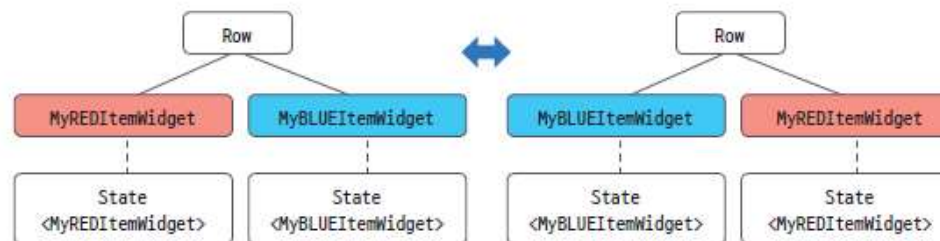


그림 8-12 다른 타입의 StatefulWidget 테스트

위젯 키

- 같은 타입의 StatefulWidget 식별하기 — 키 미사용
 - 같은 타입의 StatefulWidget 객체를 여러 개 이용한다면 위젯의 구조가 변경될 때 바라지 않는 상황이 발생할 수도 있습니다.

• StatefulWidget 위젯

```
class MyColorItemWidget extends StatefulWidget {  
  Color color;  
  MyColorItemWidget(this.color, {Key? key}): super(key: key);  
}
```

```
@override  
State<StatefulWidget> createState() {  
  return _MyColorItemWidgetState(color);  
}  
}  
  
class _MyColorItemWidgetState extends State<MyColorItemWidget> {  
  Color color;  
  _MyColorItemWidgetState(this.color);  
  @override  
  Widget build(BuildContext context) {  
    return Expanded(  
      child: Container(  
        color: color,  
        width: 150,  
        height: 150,  
      ),  
    );  
  }  
}
```

위젯 키

• 같은 타입의 객체 이용

```
class _MyListWidgetState extends State<MyListWidget> {  
  List<Widget> widgetList = [  
    MyColorItemWidget(Colors.red),  
    MyColorItemWidget(Colors.blue),  
  ];  
  onColorChange() {  
    setState(() {  
      widgetList.insert(1, widgetList.removeAt(0));  
    });  
  }  
  ... (생략) ...  
}
```



그림 8-13 같은 타입의 StatefulWidget 테스트

위젯 키

- 같은 타입의 StatefulWidget 식별하기 — 키 사용

• 위젯 키 사용

```
class _MyListWidgetState extends State<MyListWidget> {  
  List<Widget> widgetList = [  
    MyColorItemWidget(Colors.red, key: UniqueKey()),  
    MyColorItemWidget(Colors.blue, key: UniqueKey()),  
  ];  
  onChange() {  
    print(widgetList.elementAt(0).key);  
    setState(() {  
      widgetList.insert(1, widgetList.removeAt(0));  
    });  
  }  
  ... (생략) ...  
}
```

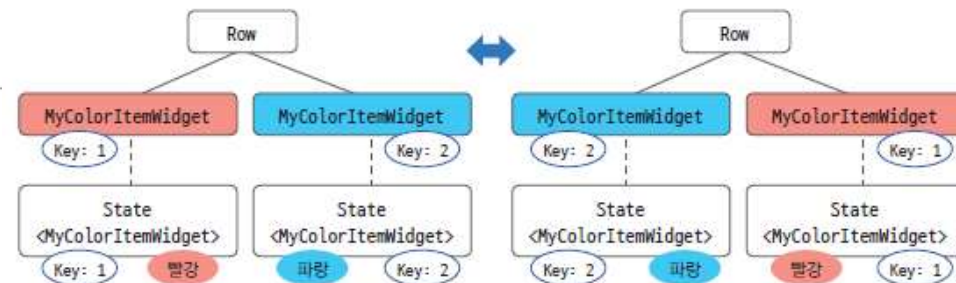


그림 8-14 같은 타입의 StatefulWidget 객체를 키로 식별 테스트

위젯 키

키 클래스

- GlobalKey는 앱 전체, GlobalKey는 이 키값이 지정된 위젯의 부모부터 자식 위젯에서 유일한 값
- GlobalKey는 currentState, currentWidget 속성을 제공
- GlobalKey 하위 클래스로 GlobalKey, UniqueKey, GlobalKey가 있고 주로 이 클래스를 이용해 위젯의 키값을 지정

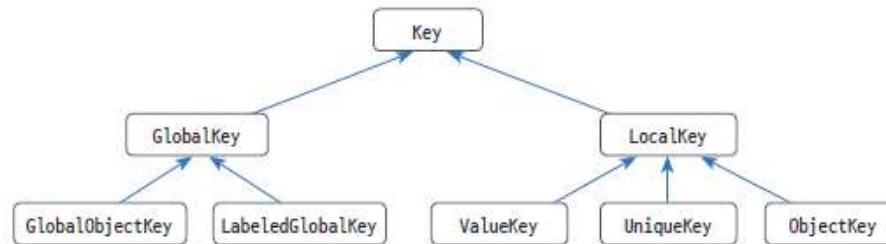


그림 8-15 키 클래스

- •ValueKey: 문자열, 숫자 키값
- •ObjectKey: 객체 키값
- •UniqueKey: 유일한 난수 키값

위젯 키

키 클래스

• 다양한 형태의 키값 지정하기

```
class _MyListWidgetState extends State<MyListWidget> {  
  List<Widget> widgetList = [  
    MyColorItemWidget(Colors.red, key: UniqueKey(),),  
    MyColorItemWidget(Colors.blue, key: ValueKey(10)),  
    MyColorItemWidget(Colors.green, key: ObjectKey(User('kkang', 30)))  
  ];  
  
  onChange() {  
    var uniqueKey = widgetList.elementAt(0).key;  
    var valueKey = widgetList.elementAt(1).key;  
    var objectKey = widgetList.elementAt(2).key as ObjectKey;  
    User user = objectKey.value as User;  
    print('${uniqueKey}, ${valueKey}, ${user.name}'); // [<10>], [#06854], kkang  
    ... (생략) ...  
  }  
  ... (생략) ...  
}
```



감사합니다

단단히 마음먹고 떠난 사람은
산꼭대기에 도착할 수 있다.
산은 올라가는 사람에게만 정복된다.



윌리엄 셰익스피어
William Shakespeare