Stateful Widget

A stateful widget is dynamic:it can change its appearance in response to events triggered by user interactions. Checkbox, Form, and TextField are examples of stateful widgets. Stateful widgets subclass StatefulWidget.

Stateless

A stateless widget never changes. Icon, IconButton, and Text are examples of stateless widgets. Stateless widgets subclass StatelessWidget.

Managing State

```
class TapboxA extends
StatefulWidget {
  TapboxA({Key key}) :
super(key: key);
  @override
  _TapboxAState createState() =>
_TapboxAState();
}
class _TapboxAState extends
State<TapboxA> {
  bool _active = false;
  void _handleTap() {
    setState(() {
      _active = !_active;
    });
  }
```

Widget manages own state

Example Stateful Implementation

```
class _FavoriteWidgetState
extends State<FavoriteWidget> {
  // ...
  @override
  Widget build(BuildContext
context) {
    return Row(
      mainAxisSize: MainAxisS-
ize.min,
      children: [
```

Example Stateful Implementation (cont)

```
        Container(
          padding: EdgeInsets.a-
ll(0),
          child: IconButton(
            icon: (_isFavorited
? Icon(Icons.star) : Icon(Icon-
s.star_border)),
            color: Colors.re-
d[500],
            onPressed: _toggl-
eFavorite,
          ),
        ),
        SizedBox(
          width: 18,
          child: Container(
            child: Text('$_favo-
riteCount'),
          ),
        ),
```

Managing State

```
lass ParentWidget extends
StatefulWidget {
  @override
  _ParentWidgetState create-
State() => _ParentWidgetState();
}
class _ParentWidgetState extends
State<ParentWidget> {
  bool _active = false;
  void _handleTapboxChanged(bool
newValue) {
    setState(() {
      _active = newValue;
    });
  }
```

Parent widget manages state

Navigation

Push new route	`Navigator.push( context, MaterialPageRoute(bui-lder: (context) => SecondRoute()), ); }`
Return to old route	`onPressed: () { Naviga-tor.pop(context); }`
Define routes	`routes: { '/': (context) => FirstScreen(), '/second': (context) => SecondScreen(), },`
Push data to new screen	`Navigator.push( context, MaterialPageRoute( builder: (context) => DetailScreen(todo: todos[index]), ),`
Return data from screen	`RaisedButton( onPressed: () { // The Yep button returns "Yep!" as the result. Navigator.pop(c-ontext, 'Yep!'); }, child: Text('Yep!'), );`

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com