Basics

Flutter has Stateful and Stateless Widgets. Stateful widgets (Checkbox, Slider) can display a state, Stateless widgets (Text, IconButton) can't.

Override build method in state

```
//override build method to return an IconButton which displays different icons based on _isFavorited, and a Text which displays _favoriteCount
@override
    Widget build( Bui ldC ontext context) {
        return Row(
            mai nAx isSize: MainAx isS ize.min,
            chi ldren: [
                Con tainer(
                    pad ding: EdgeIn set s.a ll(0),
                    child: IconBu tton(
                        icon: (_isFa vorited ? Icon(I con - s.star) : Icon(I con s.s tar - _bo rder)),
                        color: Colors.re d[500],
                        onP - ressed: _toggl eFa vorite,
                    ),
                ),
                Siz edBox(
                    width: 18,
                    child: Container(
                        child: Text(' $_f avo rit eCo unt'),
                    ),
                ),
            ],
        );
    }
```

Creating a stateful widget

```
When createState() is called (which is overridden from StatefulWidget, a new _FavoriteWidgetState is created.
class Favori teW idget extends Statef ulW idget {
    @ov erride
    _Fa vor ite Wid get State create State() => _Favor ite - Wid get Sta te();
}
```

Add toggle method

```
//sets the state when _toggleFavorite is invoked so the widget refreshes; Increase favorite count
void _toggl eFa vor ite() {
    set Sta te(() {
       if (_isFa vor ited) {
            _fa vor ite Count -= 1;
            _is Fav orited = false;
       } else {
            _fa vor ite Count += 1;
            _is Fav orited = true;
       }
   });
}
```

Create state

```
Create _FavoriteWidgetState which holds properties like isFavorited and favoriteCount
class _Favor ite Wid get State extends State< Fav ori teW idg - et> {
   bool _isFav orited = true;
   int _favor ite Count = 41;
```

Create state (cont)

```
> }
```

Types of state managment

Widget manages its own state	Sometimes it makes sense to manage state internally. ListView scrolls automatically if content exceeds render box, this is managed internally so the developer doesnt have to handle it
Parent widget manages the widget's state	Parent manages state of children and tells child widget when to update.
Mix-and-match approach	Some of the state is managed by the widget, some of it is managed by the parent widget.

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