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(BuildContext
context) {
    return Row(
      mainAxisSize: MainAxisS-
ize.min,
      children: [
        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'),
          ),
        ),
      ],
    );
  }
```

Creating a stateful widget

```
When createState() is called
(which is overridden from
StatefulWidget, a new
_FavoriteWidgetState is created.
class FavoriteWidget extends
StatefulWidget {
  @override
  _FavoriteWidgetState create-
State() => _FavoriteWidgetSta-
te();
}
```

Add toggle method

```
//sets the state when
_toggleFavorite is invoked so
the widget refreshes; Increase
favorite count
void _toggleFavorite() {
  setState(() {
    if (_isFavorited) {
      _favoriteCount -= 1;
      _isFavorited = false;
    } else {
      _favoriteCount += 1;
      _isFavorited = true;
    }
  });
}
```

Create state

```
Create _FavoriteWidgetState
which holds properties like
isFavorited and favoriteCount
class _FavoriteWidgetState
extends State<FavoriteWidget> {
  bool _isFavorited = true;
  int _favoriteCount = 41;

}
```

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.