

If statement

```
void _toggleFavorite() {
  setState(() {
    if (_isFavorited) {
      _favoriteCount -= 1;
      _isFavorited = false;
    } else {
      _favoriteCount += 1;
      _isFavorited = true;
    }
  });
}
```

And the subclass (cont)

```
    ),
  );
}
```

Chance what happens when you press a widget

```
void _handleTapUp(TapUpDetails details) {
  setState(() {
    _highlight = false;
  });
}
```

Create the super class for a widget

```
class TapboxA extends StatefulWidget {
  TapboxA({Key key}) : super(key: key);
  @override
  _TapboxAState createState() => _TapboxAState();
}
```

Build a widget with a method call when clicked

```
Widget build(BuildContext context) {
  // This example adds a green border on tap
  // down.
  // On tap up, the square changes to the
  // opposite state.
  return GestureDetector(
    onTapUp: _handleTapUp, // when clicked
    child: Container(
      child: Center(
        child: Text(widget.active ? 'Active' :
          'Inactive',
          style: TextStyle(fontSize: 32.0,
            color: Colors.white)),
      ),
      width: 200.0,
      height: 200.0,
      decoration: BoxDecoration(
        color:
          widget.active ? Colors.lightGreen
            en[700] : Colors.grey[600],
        border: _highlight
          ? Border.all(
              color: Colors.teal[700],
              width: 10.0,
            )
          : null,
      ),
    ),
  );
}
```

And the subclass

```
class _TapboxAState extends State<TapboxA> {
  bool _active = false;
  void _handleTap() {
    setState(() {
      _active = !_active;
    });
  }
  Widget build(BuildContext context) {
    return GestureDetector(
      onTap: _handleTap,
      child: Container(
        child: Center(
          child: Text(
            _active ? 'Active' : 'Inactive',
            style: TextStyle(fontSize: 32.0,
              color: Colors.white),
          ),
        ),
        width: 200.0,
        height: 200.0,
        decoration: BoxDecoration(
          color: _active ? Colors.lightGreen[700]
            : Colors.grey[600],
        ),
      ),
    ),
  );
}
```

Get data from a call

```
oid main() {
  runApp(MaterialApp(
    title: 'Passing Data',
    home: TodosScreen(
      todos: List.generate(
        20,
        (i) => Todo(
          'Todo $i',
          'A description of what needs to be done
for Todo $i',
        ),
      ),
    ),
  ));
}
```

C

By **kahmic16**

cheatography.com/kahmic16/

Not published yet.

Last updated 18th March, 2020.

Page 2 of 2.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>