If statement

```
void _toggleFavorite() {
    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 the super class for a widget

```
class TapboxA extends StatefulWidget {
    Tap box A({Key key}) : super(key: key);
    @ov erride
    _Ta pbo xAState create State() => _Tapbo xAS -
tate();
}
```

And the subclass

```
class _TapboxAState extends State<TapboxA> {
   bool _active = false;
   void _handl eTap() {
        set Sta te(() {
            _active = !_active;
       });
   }
    Widget build( Bui ldC ontext context) {
        return Gestur eDe tector(
            onTap: _handl eTap,
            child: Container(
                child: Center(
                    child: Text(
                        _active ? 'Active' :
'Inact ive',
                        style: TextSt yle (fo -
ntSize: 32.0, color: Colors.wh ite),
                    ),
                ),
                width: 200.0,
                height: 200.0,
                dec ora tion: BoxDec ora tion(
```

And the subclass (cont)

```
>           color: _active ? Colors.lightGreen[700] : Colors.grey[600],
      ),
    ),
  );
 }
}
```

Chance what happens when you press a widget

```
void _handleTapUp(TapUpDetails details) {
        set Sta te(() {
            _hi ghlight = false;
       });
   }
```

Get data from a call

```
oid main() {
    run App (Ma ter ialApp(
        title: 'Passing Data',
        home: TodosS creen(
            todos: List.g ene rate(
               20,
               (i) => Todo(
                    'Todo $i',
                   'A descri ption of what needs
to be done for Todo $i',
               ),
           ),
       ),
   ));
}
```


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