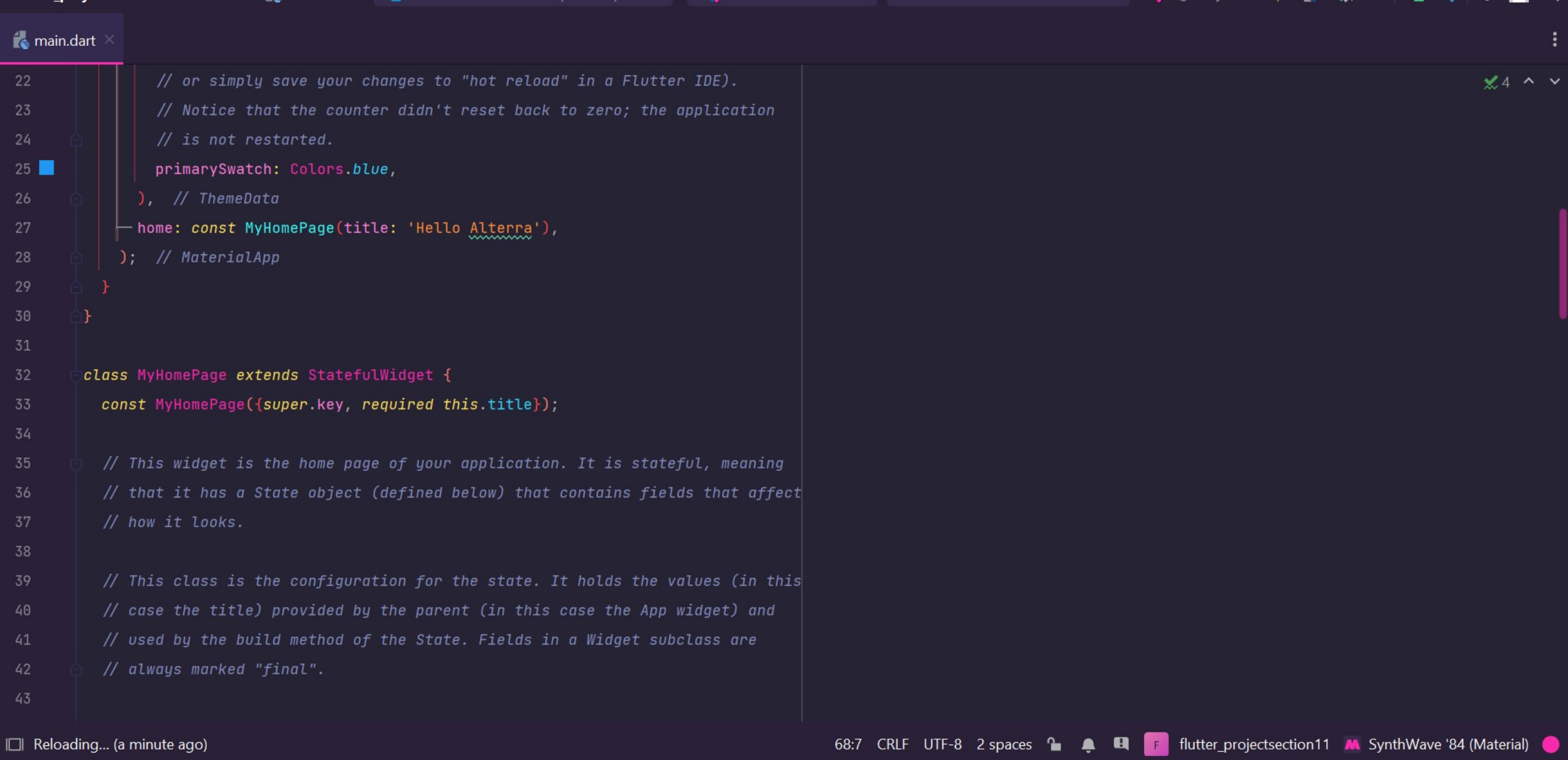


```
1  import 'package:flutter/material.dart';
2
3  void main() {
4    runApp(const MyApp());
5  }
6
7  class MyApp extends StatelessWidget {
8    const MyApp({super.key});
9
10   // This widget is the root of your application.
11   @override
12   Widget build(BuildContext context) {
13     return MaterialApp(
14       title: 'Flutter Demo',
15       theme: ThemeData(
16         // This is the theme of your application.
17         //
18         // Try running your application with "flutter run". You'll see the
19         // application has a blue toolbar. Then, without quitting the app, try
20         // changing the primarySwatch below to Colors.green and then invoke
21         // "hot reload" (press "r" in the console where you ran "flutter run",
22         // or simply save your changes to "hot reload" in a Flutter IDE).
```



```
43
44   final String title;
45
46   @override
47   State<MyHomePage> createState() => _MyHomePageState();
48 }
49
50 class _MyHomePageState extends State<MyHomePage> {
51   int _counter = 0;
52
53   void _incrementCounter() {
54     setState(() {
55       // This call to setState tells the Flutter framework that something has
56       // changed in this State, which causes it to rerun the build method below
57       // so that the display can reflect the updated values. If we changed
58       // _counter without calling setState(), then the build method would not be
59       // called again, and so nothing would appear to happen.
60       _counter++;
61     });
62   }
63
64   @override
```

```
64  @override
65  Widget build(BuildContext context) {
66    // This method is rerun every time setState is called, for instance as done
67    // by the _incrementCounter method above.
68    //
69    // The Flutter framework has been optimized to make rerunning build methods
70    // fast, so that you can just rebuild anything that needs updating rather
71    // than having to individually change instances of widgets.
72    return Scaffold(
73      appBar: AppBar(
74        // Here we take the value from the MyHomePage object that was created by
75        // the App.build method, and use it to set our appbar title.
76        title: Text(widget.title),
77      ), // AppBar
78      body: Center(
79        // Center is a layout widget. It takes a single child and positions it
80        // in the middle of the parent.
81        child: Column(
82          // Column is also a layout widget. It takes a list of children and
83          // arranges them vertically. By default, it sizes itself to fit its
84          // children horizontally, and tries to be as tall as its parent.
85          //
```

main.dart

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

```
// Toggle debug paint settings from the Flutter Inspector in Android Studio, or the "Toggle Debug Paint" command in Visual Studio Code)
// to see the wireframe for each widget.
//
// Column has various properties to control how it sizes itself and
// how it positions its children. Here we use mainAxisAlignment to
// center the children vertically; the main axis here is the vertical
// axis because Columns are vertical (the cross axis would be
// horizontal).
mainAxisAlignment: MainAxisAlignment.center,
children: <Widget>[
  const Text(
    'Nama Saya Adalah Tobias Samuel Salim',
  ), // Text
  Text(
    '$_counter',
    style: Theme.of(context).textTheme.headlineMedium,
  ), // Text
], // <Widget>[]
), // Column
), // Center
floatingActionButton: FloatingActionButton(
  onPressed: _incrementCounter,
```

✓ 4

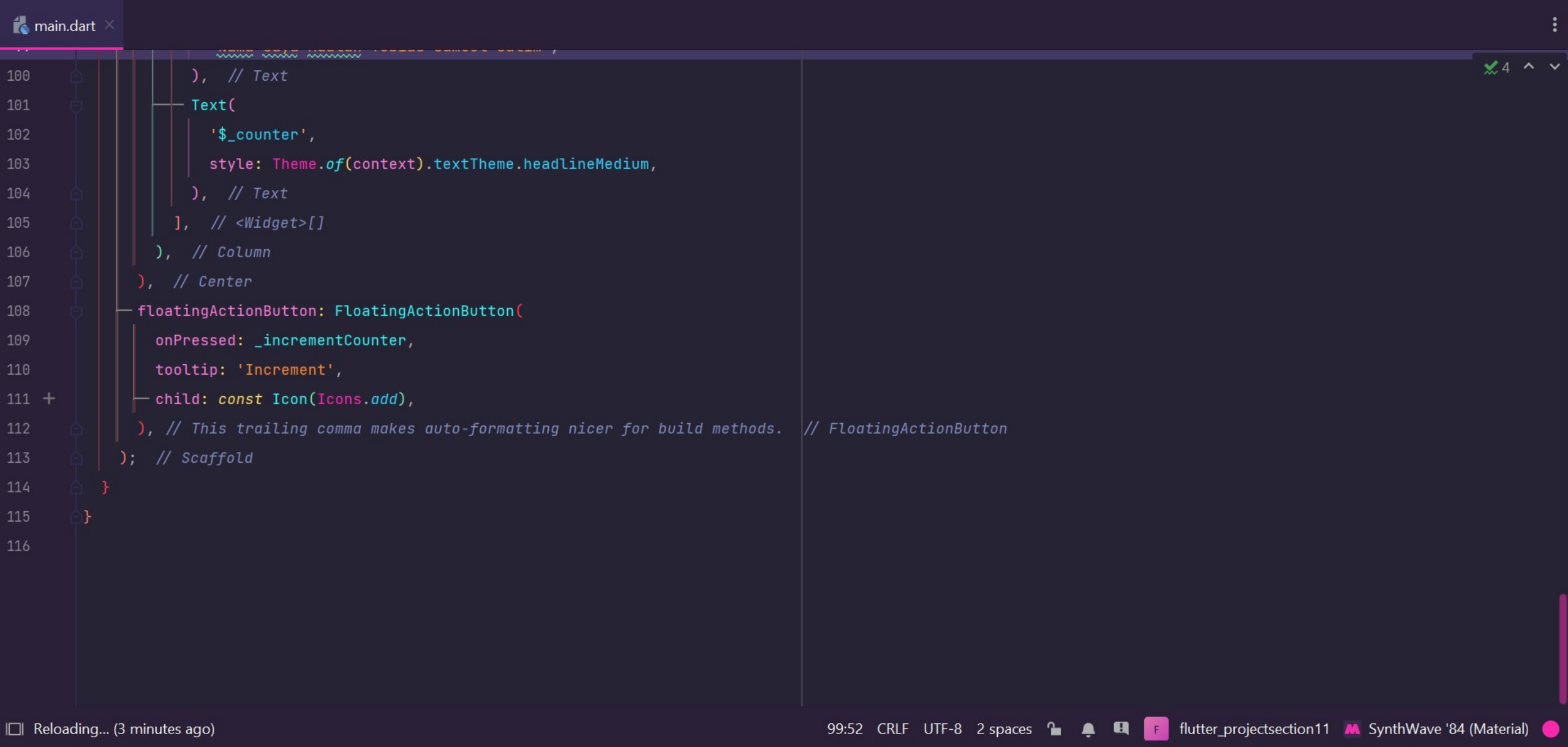
^

▼

Reloading... (3 minutes ago)

99:52 CRLF UTF-8 2 spaces

flutter_projectsection11 SynthWave '84 (Material)



```
100     ), // Text
101     Text(
102       '$_counter',
103       style: Theme.of(context).textTheme.headlineMedium,
104     ), // Text
105   ], // <Widget>[]
106 ), // Column
107 ), // Center
108 floatingActionButton: FloatingActionButton(
109   onPressed: _incrementCounter,
110   tooltip: 'Increment',
111   child: const Icon(Icons.add),
112 ), // This trailing comma makes auto-formatting nicer for build methods. // FloatingActionButton
113 ); // Scaffold
114 }
115 }
116
```