Flutter: Themes

- Themes are the packages that contain the graphical appearances to our website or mobile app screen.

- We mainly use themes for sharing the **color and font styles**.

- In mobile development, it becomes mandatory to add the **Light and Dark** theme for our app.

- You can either define app-wide themes, or use Theme widgets that define the colors and font styles for a particular part of the application.

- In fact, app-wide themes are just Theme widgets created at the root of an app by the MaterialApp.

- Flutter allows us two approaches to do this:
  - **Creating an app theme (app wide themes)**
  - **Themes for part of an application.**
    - By creating a unique Theme Data.
    - By extending the parent theme.

Creating an app theme (app wide themes)

To share a Theme across an entire app, provide a ThemeData to the MaterialApp constructor.

If no theme is provided, Flutter creates a default theme for you.

We have a lot of options available in **ThemeData(theme class for flutter)**. But mostly we use 3 – 4 of it. Some people use 8 -10 of it.

Some of them are given below:

1. **Brightness brightness (*The desired theme*):**

Brightness is an enum with two options dark and light. It helps us to **distinguish** which **theme** (light or dark) to be used.

2. **VisualDensity visualDensity:**

It defines the visual density of user interface components. Density, in the context of a UI, is vertical and horizontal 'compactness' of the components in the UI. It is unitless since it means different for different UI components.

3. **MaterialColor primarySwatch:**

Look at its type. **It is MaterialColor**. That's why it will contain color shades from 50 to 900. PrimarySwatch is used to configure default values for several fields including primaryColor, primaryColorBrightness, primaryColorLight, primaryColorDark, toggableActiveColor, accentColor, colorScheme, secondaryHeaderColor, textSelectionColor,backgroundColor and buttonColor

4. **Color primaryColor:**

This is the **background color of major parts** of the app like toolbars, tab bars, appbar, and many more.

5. **Brightness primaryColorBrightness:**

This is the brightness of the primary color. It is used to determine the **color of text and icons** placed on the top of the primary color.

6. **Color primaryColorLight:**

This is a **lighter version of primaryColor**.

7. **Color primaryColorDark:**

This is a **darker version of primaryColor**.

8. **Color canvasColor:**

This is default color of **MaterialType.canvas** (*Rectangle using default theme canvas color)*

9. **Color accentColor:**

The accent color is also known as a **secondary color**. This is **foreground color for widgets like knobs, text, overscroll edge effect**, etc.

10. **Brightness accentColorBrightness:**

It is the **brightness of the accentColor**. It is used to determine the color of text and icons placed on the top of the accent color(e.g. the icons on a floating action button).

11. **Color scaffoldBackgroundColor:**

It is the **background color of the Scaffold** widget.

12. **Color bottomAppBarColor:**

It is the **default color for *BottomAppBa*r color**. This can be overridden by specifying color in BottomAppBar e.g. BottomAppBar(color:#color);

13. **Color cardColor:**

This is the **color of Material** when used as a Card i.e. default color for Card widget.

14. **Color dividerColor:**

This is **color of Dividers and PopMenuDividers** also used between ListTiles, between rows in DataTables and so forth.

15. **Color focusColor:**

This is a **focus color used to indicate that a component** has input focus.

- **Creating an app theme (app wide themes)**

```
MaterialApp(
  title: appName,
  theme: ThemeData(
    // Define the default brightness and colors.
    brightness: Brightness.dark,
    primaryColor: Colors.lightBlue[800],

    // Define the default font family.
    fontFamily: 'Georgia',

    // Define the default `TextTheme`. Use this to specify the default
    // text styling for headlines, titles, bodies of text, and more.
    textTheme: const TextTheme(
      headline1: TextStyle(fontSize: 72.0, fontWeight: FontWeight.bold),
      headline6: TextStyle(fontSize: 36.0, fontStyle: FontStyle.italic),
      bodyText2: TextStyle(fontSize: 14.0, fontFamily: 'Hind'),
    ),
  ),
  home: const MyHomePage(
    title: appName,
  ),
);
```

- **Themes for part of an application.**

  - By creating a unique Theme Data.

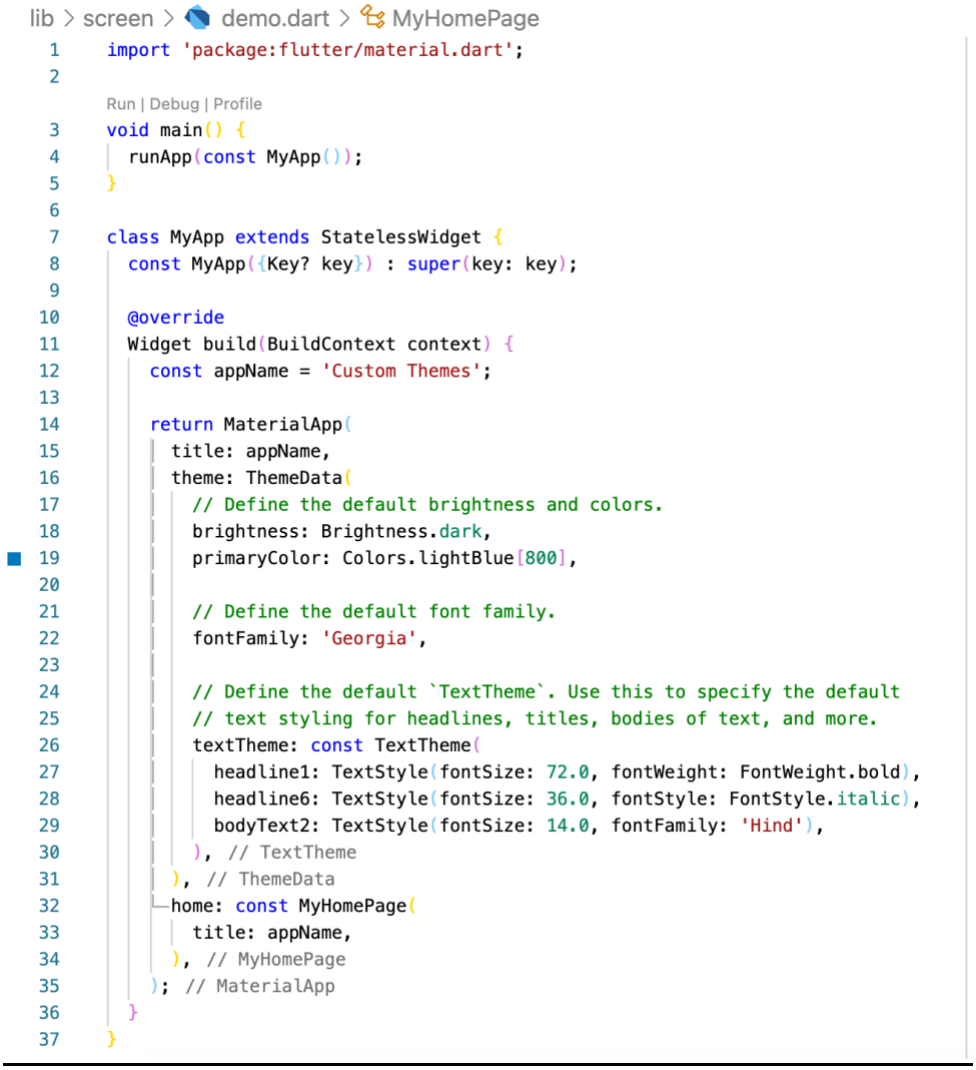
If you don't want to inherit any application colors or font styles, create a ThemeData() instance and pass that to the Theme widget.

```
Theme(
  // Create a unique theme with `ThemeData`
  data: ThemeData(
    splashColor: Colors.yellow,
  ),
  child: FloatingActionButton(
    onPressed: () {},
    child: const Icon(Icons.add),
  ),
);
```

- By extending the parent theme.

Rather than overriding everything, it often makes sense to extend the parent theme. You can handle this by using the `copyWith()` method.

```
Theme(
  // Find and extend the parent theme using `copyWith`. See the next
  // section for more info on `Theme.of`.
  data: Theme.of(context).copyWith(splashColor: Colors.yellow),
  child: const FloatingActionButton(
    onPressed: null,
    child: Icon(Icons.add),
  ),
);
```




```dart
import 'package:flutter/material.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    const appName = 'Custom Themes';

    return MaterialApp(
      title: appName,
      theme: ThemeData(
        // Define the default brightness and colors.
        brightness: Brightness.dark,
        primaryColor: Colors.lightBlue[800],

        // Define the default font family.
        fontFamily: 'Georgia',

        // Define the default `TextTheme`. Use this to specify the default
        // text styling for headlines, titles, bodies of text, and more.
        textTheme: const TextTheme(
          headline1: TextStyle(fontSize: 72.0, fontWeight: FontWeight.bold),
          headline6: TextStyle(fontSize: 36.0, fontStyle: FontStyle.italic),
          bodyText2: TextStyle(fontSize: 14.0, fontFamily: 'Hind'),
        ), // TextTheme
      ), // ThemeData
      home: const MyHomePage(
        title: appName,
      ), // MyHomePage
    ); // MaterialApp
  }
}
```

```
38
39  class MyHomePage extends StatelessWidget {
40    final String title;
41
42    const MyHomePage({Key? key, required this.title}) : super(key: key);
43
44    @override
45    Widget build(BuildContext context) {
46      return Scaffold(
47        appBar: AppBar(
48          title: Text(title),
49        ), // AppBar
50        body: Center(
51          child: Container(
52            color: Theme.of(context).colorScheme.secondary,
53            child: Text(
54              'Text with a background color',
55              style: Theme.of(context).textTheme.headline6,
56            ), // Text
57          ), // Container
58        ), // Center
59        floatingActionButton: Theme(
60          data: Theme.of(context).copyWith(splashColor: Colors.yellow),
61          child: FloatingActionButton(
62            onPressed: () {},
63            child: const Icon(Icons.add),
64          ), // FloatingActionButton
65        ), // Theme
66      ); // Scaffold
67    }
68  }
```