Google Codelab Overview

# Dive into Dart's patterns and records:

* Records
* Destructure a record
* If-case statement
* Switch expressions
* Sealed class
* Exhaustive switching

# Building scrolling experiences in Flutter:

* CustomScrollView (in body of scaffold)
* SliverAppbar (pinned, stretched, flexible space etc. )
* Flexible spacebar (collapse modes, strectch modes)
* Box Decoration (gradients)
* Sliver List (delegate : SliverChildBuilderDelegate)

# Dart null safety in Action:

* Non-nullable (int a ), nullable(int? a), null-assertion operator(a!.isNegative).
* Type promotion via null checks
* Late keyword
* Late circular references

# How to manage application states using inherited widgets:

* [Simplifying State Management with  Flutter InheritedWidget (dhiwise.com)](https://www.dhiwise.com/post/simplifying-state-management-in-flutter-inheritedwidget)
* [Inherited Widgets Flutter by cavin macwan | Medium](https://medium.com/@CavinMac/inherited-widgets-in-depth-413794c3b3d9)

# How to debug layout issues with the Flutter Inspector:

* [How to debug layout issues with the Flutter Inspector | by Katie Lee | Flutter | Medium](https://medium.com/flutter/how-to-debug-layout-issues-with-the-flutter-inspector-87460a7b9db)

# Implicit animations:

* Animated Opacity
* Animated Container
* [Implicit animations | Flutter](https://docs.flutter.dev/codelabs/implicit-animations)

NOTE : YOU CAN GENERATE THEMES USING [Material Theme Builder (material-foundation.github.io)](https://material-foundation.github.io/material-theme-builder/)

# Building Beautiful Transitions with Material Motion for Flutter:

* four main Material transition patterns
* Container Transform (Open Container widget)
* Shared Axis
* Fade Through
* Fade
* Custom Navigator (will take custom pages you’ll build from Page class below)
* Class extending Page
* PageTransitionSwitcher (Widget)
* Route method
* If you want to give transition animations on both pages, use custom Navigator and if you only want to give transition on the second page use Navigator.push with a custom route (Route method)

# Take your Flutter app from boring to beautiful:

* Typography (Adding google fonts)
* Themes (dark themes , light themes, ThemeModes, Color scheme from seed with brightness attribute for both dark and light themes)
* Adaptive deisgn (Layout builder, Media query (Should use media query when many different properties are required and if you only just want to adjust layout , then use layout builder ))
* pageTransitionsTheme (defined in themedata)

# MDC-101 Flutter: Material Components Basics:

* Overflowbar (The **OverflowBar** arranges its children in a row.)
* resizeToAvoidBottomInset (false ensures that the keyboard's appearance does not alter the size of the page or its widgets)

# MDC-102 Flutter: Material Structure and Layout:

* Grid layout
* Card
* Stack
* Aspect ratio (use to layout images width/height)
* Backdrop filter widget