Get started with Jetpack Compose

Jetpack Compose is the modern toolkit for building native Android UI. Here's where you'll find the latest information about using Compose.

- Overview: See all the resources available to Compose developers.
- Tutorial: Get started with Compose, by using it to build a simple UI.

Foundation

- Thinking in Compose: Learn how Compose's declarative approach is different from the view-based approach you may have used in the past, and how to build a mental model of working with Compose.
- Managing state: Learn about setting and using state in your Compose app.
- Lifecycle of composables: Learn about the lifecycle of a composable, and how Compose decides if it needs to be redrawn.
- Modifiers: Learn how to use modifiers to augment or decorate your composables.
- Side-effects in Compose: Learn the best ways to manage side-effects.
- Jetpack Compose Phases: Learn about the steps Compose goes through to render your UI, and how to use that information to write efficient code
- Architectural layering: Learn about the architectural layers that make up Jetpack Compose, and the core principles that informed its design.
- Performance: Learn how to avoid the common programming pitfalls that can hurt your app's performance.
- Semantics in Compose: Learn about the Semantics tree, which organizes your UI in a way that can be used by accessibility services and the testing framework.
- Locally scoped data with CompositionLocal: Learn how to use `CompositionLocal` to pass data through the Composition.

Development environment

- Android Studio with Compose: Set up your development environment to use Compose.
- Tooling for Compose: Learn about Android Studio's new features to support Compose.
- Kotlin for Compose: Learn how certain Kotlin-specific idioms work with Compose.
- Compare Compose and View performance: Learn how migrating to Compose can affect your app's APK size and runtime performance.
- Bill of Materials: Manage all your Compose dependencies by specifying only the BOM's version.

Design

- Layouts: Learn about Compose's native layout components, and how to design your own.
    - Layout basics: Learn about the building blocks for a straightforward app UI.
    - Material Components and layouts: Learn about Material components and layouts in Compose.

    - Custom layouts: Learn how to take control of your app's layout, and how to design a custom layout of your own.
    - Build adaptive layouts: Learn how to use Compose to build layouts that adapt to different screen sizes, orientations, and form factors.
    - Alignment lines: Learn how to create custom alignment lines to precisely align and position your UI elements.
    - Intrinsic measurements: Since Compose only allows you to measure UI elements once per pass, this page explains how to query for information about child elements before measuring them.
    - ConstraintLayout: Learn how to use `ConstraintLayout` in your Compose UI.
- Design Systems: Learn how to implement a design system and give your app a consistent look and feel.
    - Material Design 3: Learn how to implement Material You with Compose's implementation of Material Design 3.
    - Migrating from Material 2 to Material 3: Learn how to migrate your app from Material Design 2 to Material Design 3 in Compose.
    - Material Design 2: Learn how to customize Compose's implementation of Material Design 2 to fit your product's brand.
    - Custom design systems: Learn how to implement a custom design system in Compose, and how to adapt existing Material Design composables to handle this.
    - Anatomy of a theme: Learn about the lower-level constructs and APIs used by `MaterialTheme` and custom design systems.
- Lists and grids: Learn about some of Compose's options for managing and displaying lists and grids of data.
- Text: Learn about Compose's main options for displaying and editing text.
- Graphics: Learn about Compose's features for building and working with custom graphics.
- Animation: Learn about Compose's different options for animating your UI elements.
- Gestures: Learn how to build a Compose UI that detects and interacts with user gestures.
- Handling user interactions: Learn how Compose abstacts low-level input into higher-level interactions, so you can customize how your components respond to user actions.

Adopting Compose

- Migrate existing View-based apps: Learn how to migrate your existing View-based app to Compose.
    - Migration strategy: Learn the strategy to safely and incrementally introduce Compose into your codebase.
    - Interoperability APIs: Learn about Compose's APIs to help you combine Compose with View-based UI.
    - Other considerations: Learn about other considerations like theming, architecture, and testing while migrating your View-based app to Compose.
- Compose and other libraries: Learn how to use view-based libraries in your Compose content.
- Compose architecture: Learn how to implement the unidirectional flow pattern in Compose, how to implement events and state holders, and how to work with `ViewModel` in Compose.

- Navigation: Learn how to use `NavController` to integrate the Navigation component with your Compose UI.
    - Navigation for responsive UIs: Learn how to design your app's navigation so it adapts to different screen sizes, orientations, and form factors.
- Resources: Learn how to work with your app's resources in your Compose code.
- Accessibility: Learn how to make your Compose UI suitable for users with different accessibility requirements.
- Testing: Learn about testing your Compose code.
    - Testing cheat sheet: A quick reference of useful Compose testing APIs.