

What is Android Studio?

Android Studio is a tool that makes developing an Android App much easier.

Android Studio is an **Integrated Development Environment**, that contains a text editor, debugging tools, and tools for running your app.

Android Studio has its own version of U.I Components; more specifically **Navigation Components**:

Navigation

Navigation refers to the interactions that allow users to navigate across, into, and back out from the different pieces of content within your app. Android Jetpack's Navigation component helps you implement navigation, from simple button clicks to more complex patterns, such as app bars and the navigation drawer. The Navigation component also ensures a consistent and predictable user experience by adhering to an established [set of principles](#).

The Navigation component consists of three key parts that are described below:

- **Navigation graph**: An XML resource that contains all navigation-related information in one centralized location. This includes all of the individual content areas within your app, called *destinations*, as well as the possible paths that a user can take through your app.
- **NavHost**: An empty container that displays destinations from your navigation graph. The Navigation component contains a default `NavHost` implementation, `NavHostFragment`, that displays fragment destinations.
- **NavController**: An object that manages app navigation within a `NavHost`. The `NavController` orchestrates the swapping of destination content in the `NavHost` as users move throughout your app.

As you navigate through your app, you tell the `NavController` that you want to navigate either along a specific path in your navigation graph or directly to a specific destination. The `NavController` then shows the appropriate destination in the `NavHost`.

The Navigation component provides a number of other benefits, including the following:

- Handling fragment transactions.
- Handling Up and Back actions correctly by default.
- Providing standardized resources for animations and transitions.
- Implementing and handling deep linking.
- Including Navigation UI patterns, such as navigation drawers and bottom navigation, with minimal additional work.
- [Safe Args](#) - a Gradle plugin that provides type safety when navigating and passing data between destinations.
- `ViewModel` support - you can scope a `ViewModel` to a navigation graph to share UI-related data between the graph's destinations.

In addition, you can use Android Studio's [Navigation Editor](#) to view and edit your navigation graphs.