**Fragment**

A fragment defines and manages its own layout, has its own lifecycle, and can handle its own input events. Fragments cannot live on their own--they must be hosted by an activity or another fragment. The fragment's view hierarchy becomes part of, or attaches to, the host's view hierarchy.

**Fragment Lifecycle**

Each Fragment instance has its own lifecycle. When a user navigates and interacts with your app, your fragments transition through various states in their lifecycle as they are added, removed, and enter or exit the screen.

To manage lifecycle, Fragment implements LifecycleOwner, exposing a Lifecycle object that you can access through the getLifecycle() method.

Each possible Lifecycle state is represented in the Lifecycle.State enum.

* INITIALIZED
* CRATED
* STARTED
* RESUMED
* DESTROYED

By building Fragment on top of Lifecycle, you can use the techniques and classes available for Handling Lifecycles with Lifecycle-Aware Components. For example, you might display the device's location on the screen using a lifecycle-aware component. This component could automatically start listening when the fragment becomes active and stop when the fragment moves to an inactive state.

As an alternative to using a LifecycleObserver, the Fragment class includes callback methods that correspond to each of the changes in a fragment's lifecycle. These include onCreate(), onStart(), onResume(), onPause(), onStop(), and onDestroy().

A fragment's view has a separate Lifecycle that is managed independently from that of the fragment's Lifecycle. Fragments maintain a LifecycleOwner for their view, which can be accessed using getViewLifecycleOwner() or getViewLifecycleOwnerLiveData(). Having access to the view's Lifecycle is useful for situations where a Lifecycle-aware component should only perform work while a fragment's view exists, such as observing LiveData that is only meant to be displayed on the screen.

This topic discusses the Fragment lifecycle in detail, explaining some of the rules that determine a fragment's lifecycle state and showing the relationship between the Lifecycle states and the fragment lifecycle callbacks.