Android Fragments

Android Fragment is the part of activity, it is also known as sub-activity. There can be more than one fragment in an activity. Fragments represent multiple screens inside one activity.

Android fragment lifecycle is affected by activity lifecycle because fragments are included in activity.

Each fragment has its own life cycle methods that is affected by activity life cycle because fragments are embedded in activity.

The FragmentManager class is responsible to make interaction between fragment objects.

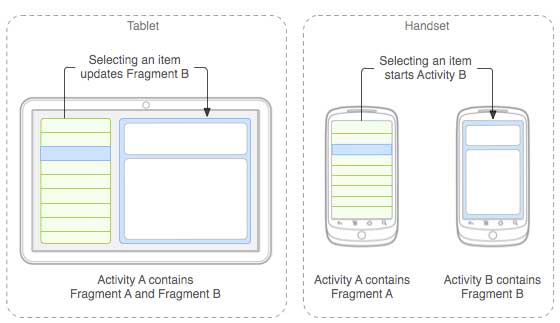
Following are important points about fragment −

* A fragment has its own layout and its own behaviour with its own life cycle call-backs.
* You can add or remove fragments in an activity while the activity is running.
* You can combine multiple fragments in a single activity to build a multi-pane UI.
* A fragment can be used in multiple activities.
* Fragment life cycle is closely related to the life cycle of its host activity which means when the activity is paused, all the fragments available in the activity will also be stopped.
* A fragment can implement a behaviour that has no user interface component.

You create fragments by extending **Fragment** class and You can insert a fragment into your activity layout by declaring the fragment in the activity's layout file, as a **<fragment>** element.

Prior to fragment introduction, we had a limitation because we can show only a single activity on the screen at one given point in time. So we were not able to divide device screen and control different parts separately. But with the introduction of fragment we got more flexibility and removed the limitation of having a single activity on the screen at a time. Now we can have a single activity but each activity can comprise of multiple fragments which will have their own layout, events and complete life cycle.

Following is a typical example of how two UI modules defined by fragments can be combined into one activity for a tablet design, but separated for a handset design.



The application can embed two fragments in Activity A, when running on a tablet-sized device. However, on a handset-sized screen, there's not enough room for both fragments, so Activity A includes only the fragment for the list of articles, and when the user selects an article, it starts Activity B, which includes the second fragment to read the article.

## Fragment Life Cycle

Android fragments have their own life cycle very similar to an android activity. This section briefs different stages of its life cycle.



#### **Fragment lifecycle**

Here is the list of methods which you can to override in your fragment class −

* **onAttach ()** The fragment instance is associated with an activity instance. The fragment and the activity is not fully initialized. Typically, you get in this method a reference to the activity which uses the fragment for further initialization work.
* **onCreate ()** The system calls this method when creating the fragment. You should initialize essential components of the fragment that you want to retain when the fragment is paused or stopped, then resumed.
* **onCreateView ()** The system calls this call back when it's time for the fragment to draw its user interface for the first time. To draw a UI for your fragment, you must return a **View** component from this method that is the root of your fragment's layout. You can return null if the fragment does not provide a UI.
* **onActivityCreated ()** The onActivityCreated () is called after the onCreateView () method when the host activity is created. Activity and fragment instance have been created as well as the view hierarchy of the activity. At this point, view can be accessed with the findViewById () method. example. In this method you can instantiate objects which require a Context object
* **OnStart ()** The onStart() method is called once the fragment gets visible.
* **onResume ()** Fragment becomes active.
* **onPause ()** The system calls this method as the first indication that the user is leaving the fragment. This is usually where you should commit any changes that should be persisted beyond the current user session.
* **onStop ()** Fragment going to be stopped by calling onStop ()
* **onDestroyView ()** Fragment view will destroy after call this method
* **onDestroy ()** onDestroy () called to do final clean-up of the fragment's state but Not guaranteed to be called by the Android platform.