 参考网址：

[Android Fragment 真正的完全解析（上）](http://blog.csdn.net/lmj623565791/article/details/37970961)

## onAttach(Activity activity)

Called when a fragment is first attached to its activity.

## onCreate(@Nullable Bundle savedInstanceState)

调用以执行片段的初始创建。 这在onAttach之后和onCreateView之前调用。

注意

当片段的活动仍在创建过程中时可以调用此方法。 因此，您不能依赖于此时初始化的活动的内容视图层次结构。 如果您想在创建活动后进行工作，请参阅onActivityCreated（Bundle）。

任何恢复的子片段将在基础Fragment.onCreate方法返回之前创建。

## onCreateView

被称为片段实例化其用户界面视图。 这是可选的，非图形片段可以返回null（这是默认实现）。 这将在onCreate（Bundle）和onActivityCreated（Bundle）之间调用。

如果从这里返回一个视图，当视图被释放时，将在onDestroyView中调用。

## onViewCreated

onCreateView（LayoutInflater，ViewGroup，Bundle）返回后立即调用，但在任何已保存的状态恢复到视图之前。 这使得子类有机会在他们知道它们的视图层次已经完全创建后初始化它们自己。 然而，片段的视图层次结构在这一点上不附加到它的父代。

## onActivityCreated

当Activity的onCreate方法返回时调用

在片段的活动已创建并且此片段的视图层次实例化时调用。 一旦这些片段到位，它可以用于执行最终初始化，例如检索视图或恢复状态。 它对于使用setRetainInstance（boolean）保留其实例的片段也很有用，因为此回调告诉片段何时与新活动实例完全关联。 这在onCreateView和beforeViewStateRestored（Bundle）之后调用。

## onStart

片段对用户可见时调用。 这通常绑定到包含Activity的生命周期的Activity.onStart。

## onResume

当片段对用户可见且正在运行时调用。 这通常绑定到包含Activity的生命周期的Activity.onResume。

## onPause

片段不再恢复时调用。 这通常绑定到包含Activity的生命周期的Activity.onPause。

## onStop

当Fragment不再启动时调用。 这通常绑定到包含Activity的生命周期的Activity.onStop

## onDestroyView

当以前由onCreateView创建的视图已从片段中分离时调用。 下一次需要显示片段时，将创建一个新视图。 这在onStop（）之后和onDestroy（）之前调用。 无论onCreateView是否返回非空视图，它都会被调用。 在内部它是在视图的状态已经保存之后但在它的父对象被删除之前调用的。

## onDestroy

当片段不再使用时调用。 这在onStop（）和onDetach（）之后调用。

## onDetach

当片段不再连接到其活动时调用。 这在onDestroy（）之后调用。

