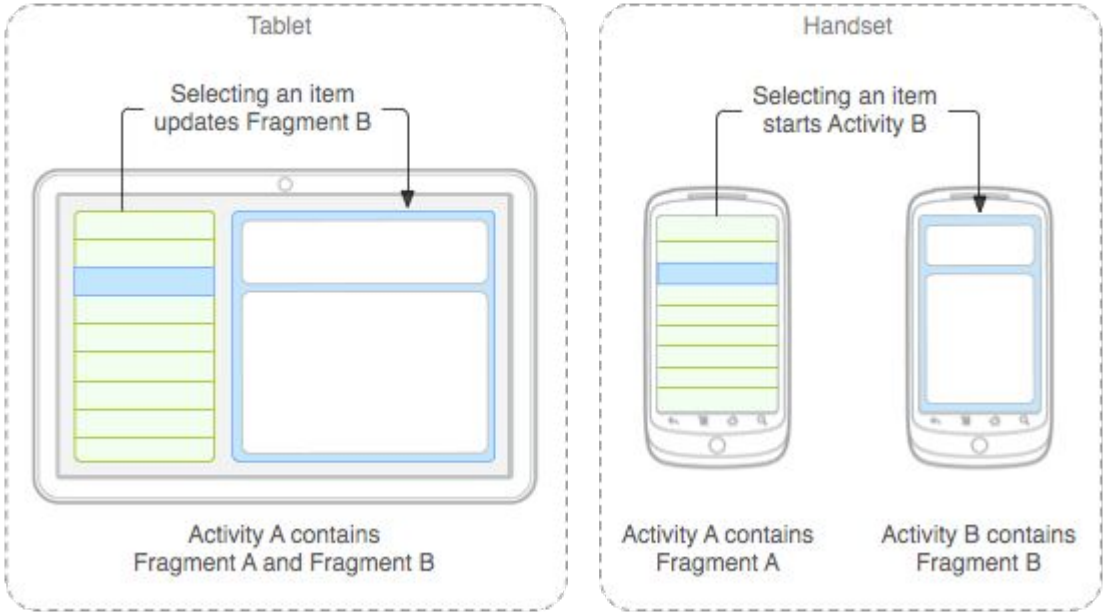
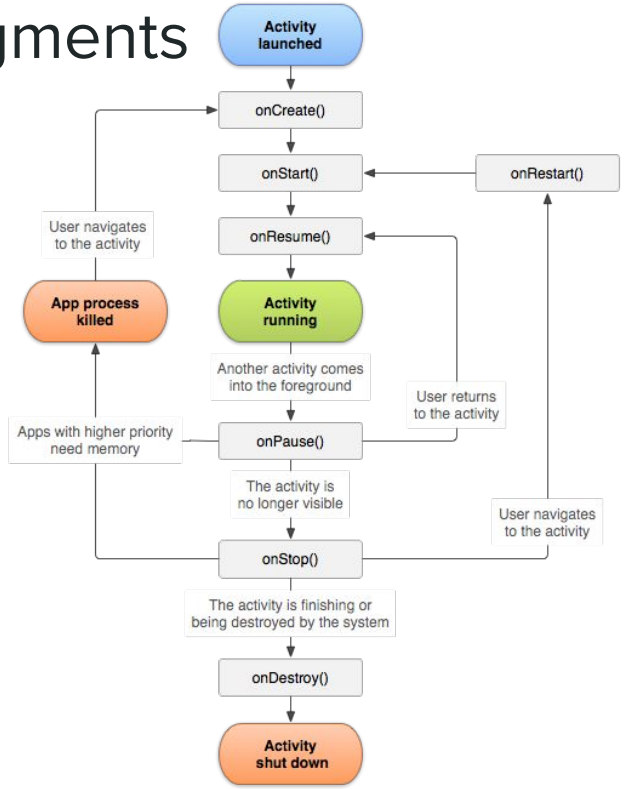
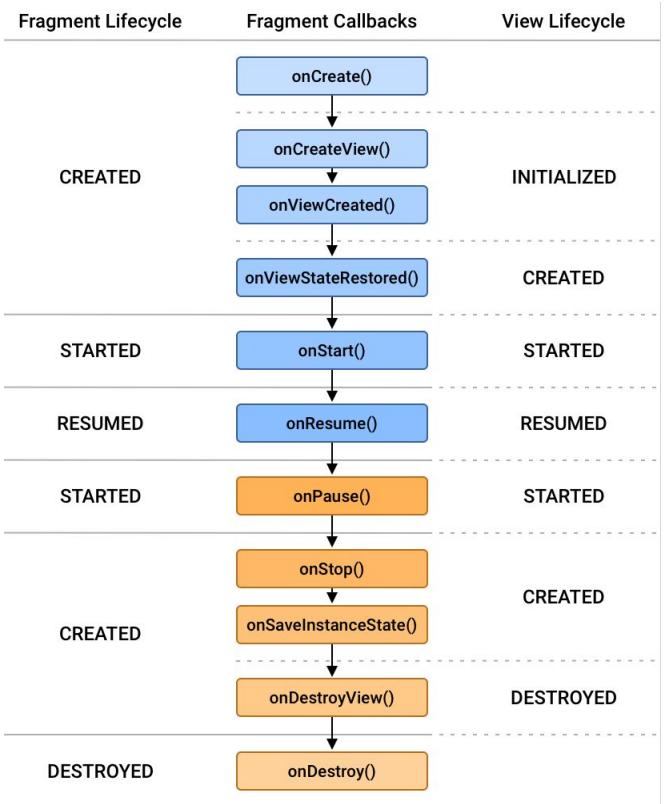
# Android Development

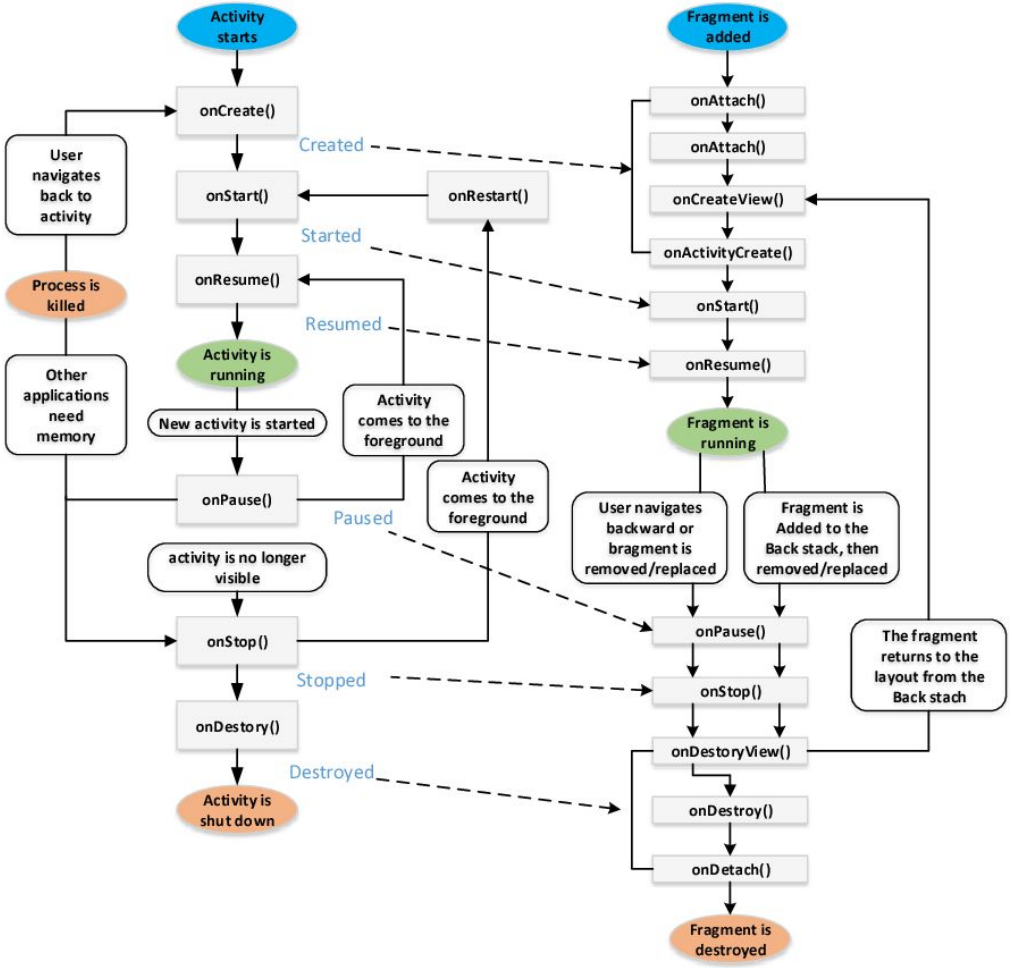
# Fragments


Tablet
Selecting an item
updates Fragment B
Activity A contains
Fragment A and Fragment B
Handset
Selecting an item
starts Activity B
Activity A contains
Fragment A
Activity B contains
Fragment B

# Fragments

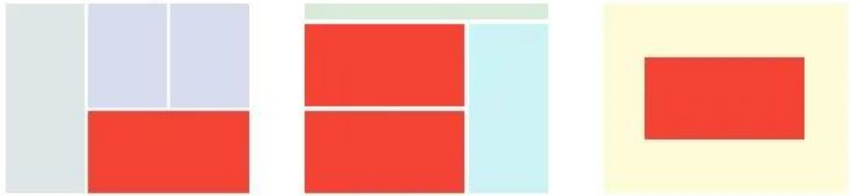

Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down


Fragment Lifecycle
Fragment Callbacks
View Lifecycle
onCreate()
onCreateView()
CREATED
INITIALIZED
onViewCreated()
onViewStateRestored()
CREATED
STARTED
onStart()
STARTED
RESUMED
onResume()
RESUMED
STARTED
onPause()
STARTED
onStop()
CREATED
CREATED
onSaveInstanceState()
onDestroyView()
DESTROYED
DESTROYED
onDestroy()


Activity starts
onCreate()
User navigates back to activity
Process is killed
Other applications need memory
onStart()
onRestart()
onResume()
Activity is running
New activity is started
Activity comes to the foreground
onPause()
Activity comes to the foreground
activity is no longer visible
onStop()
onDestory()
Activity is shut down
Created
Started
Resumed
Paused
Stopped
Destroyed
Fragment is added
onAttach()
onAttach()
onCreateView()
onActivityCreate()
onStart()
onResume()
Fragment is running
User navigates backward or bragment is removed/replaced
Fragment is Added to the Back stack, then removed/replaced
onPause()
onStop()
onDestoryView()
onDestroy()
onDetach()
Fragment is destroyed
The fragment returns to the layout from the Back stach

Modularity

Reusability



Adaptability

```kotlin
class ExampleFragment : Fragment() {

    override fun onCreateView(
            inflater: LayoutInflater,
            container: ViewGroup?,
            savedInstanceState: Bundle?
    ): View {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.example_fragment, container, false)
    }
}
```

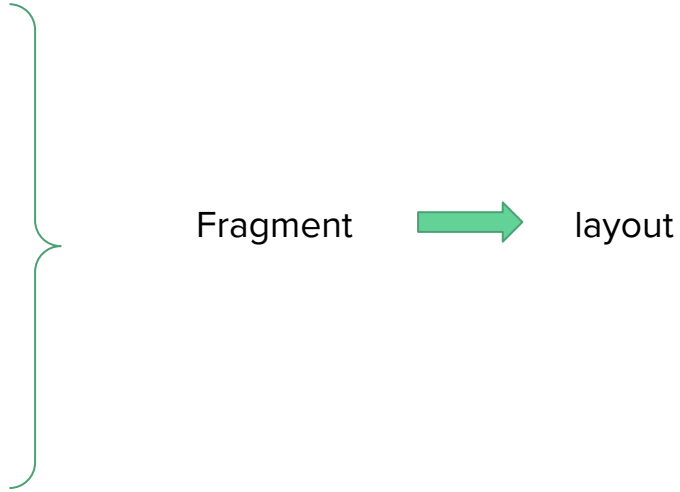
# Fragments

Layout

Fragment Manager

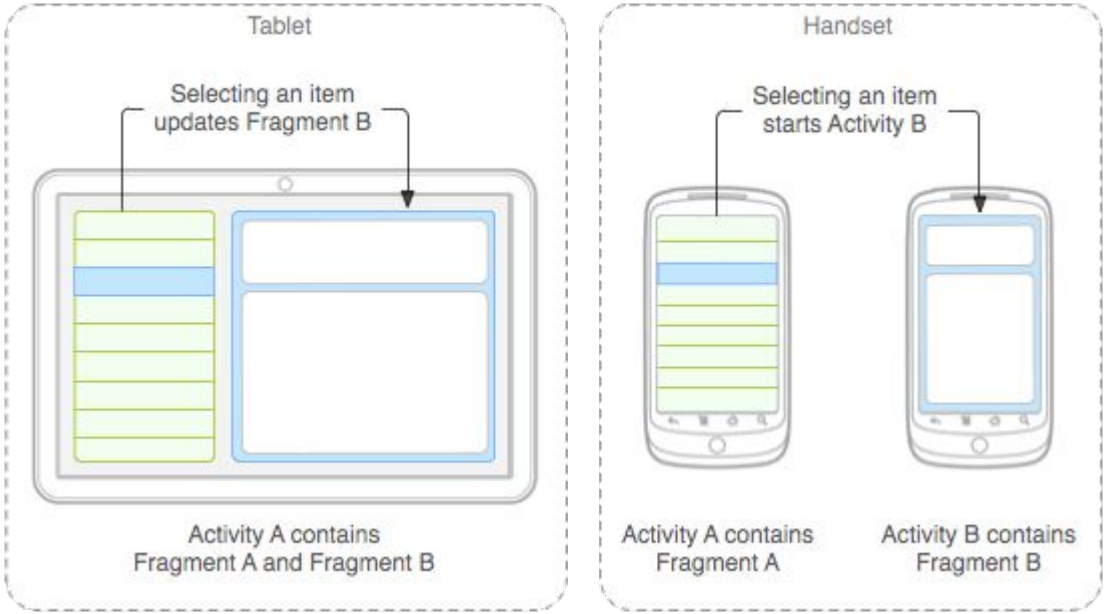
Fragment

Transaction



Fragment → layout

# Fragments


Tablet
Selecting an item
updates Fragment B
Activity A contains
Fragment A and Fragment B
Handset
Selecting an item
starts Activity B
Activity A contains
Fragment A
Activity B contains
Fragment B

# Fragments

```
dependencies {
    def fragment_version = "1.5.0"

    // Java language implementation
    implementation "androidx.fragment:fragment:$fragment_version"
    // Kotlin
    implementation "androidx.fragment:fragment-ktx:$fragment_version"
}
```

```
class ExampleFragment : Fragment(R.layout.example_fragment)
```

# Fragments

```
<androidx.fragment.app.FragmentContainerView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/fragment_container_view"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:name="com.example.ExampleFragment" />
```

```
<fragment
   android:name="com.example.examples.BlankFragment"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"></fragment>
```

# Fragments

```
class ExampleActivity : AppCompatActivity(R.layout.example_activity) {
      override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        if (savedInstanceState == null) {
            val bundle = bundleOf("some_int" to 0)
            supportFragmentManager.commit {
                setReorderingAllowed(true)
                add<ExampleFragment>(R.id.fragment_container_view, args = bundle)
            }
        }
    }
}
```

# Fragments

```
private var _binding: ResultProfileBinding? = null
// This property is only valid between onCreateView and
// onDestroyView.
private val binding get() = _binding!!

override fun onCreateView(
    inflater: LayoutInflater,
    container: ViewGroup?,
    savedInstanceState: Bundle?
): View? {
    _binding = ResultProfileBinding.inflate(inflater, container, false)
    val view = binding.root
    return view
}

override fun onDestroyView() {
    super.onDestroyView()
    _binding = null
}
```