



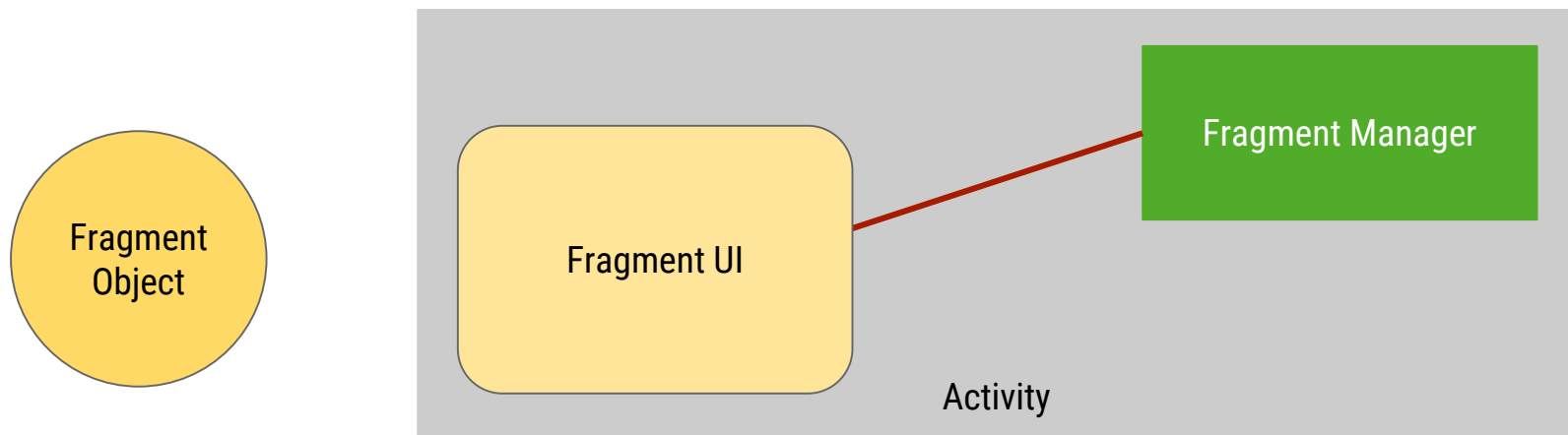
# Fragment Object States and Transactions

Fragment  
Fundamentals

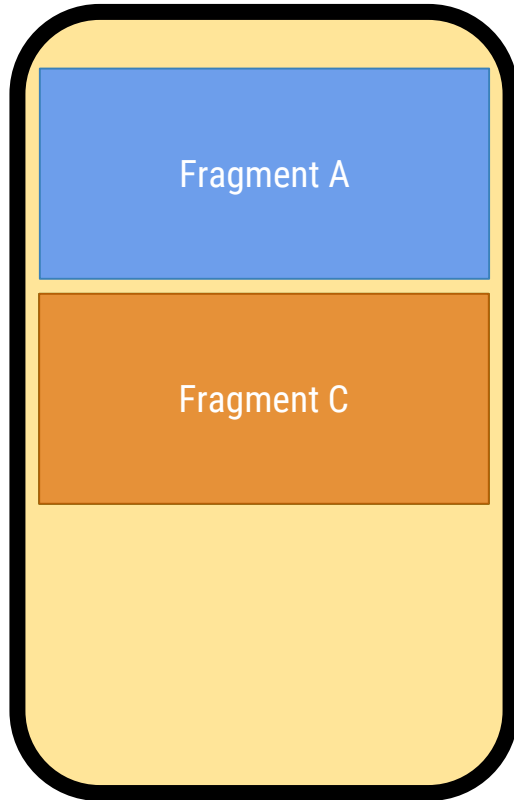
# Fragment States

Fragment exists as

- Java object
- Associated with Activity
- Associated with Activity + shown to the user as UI



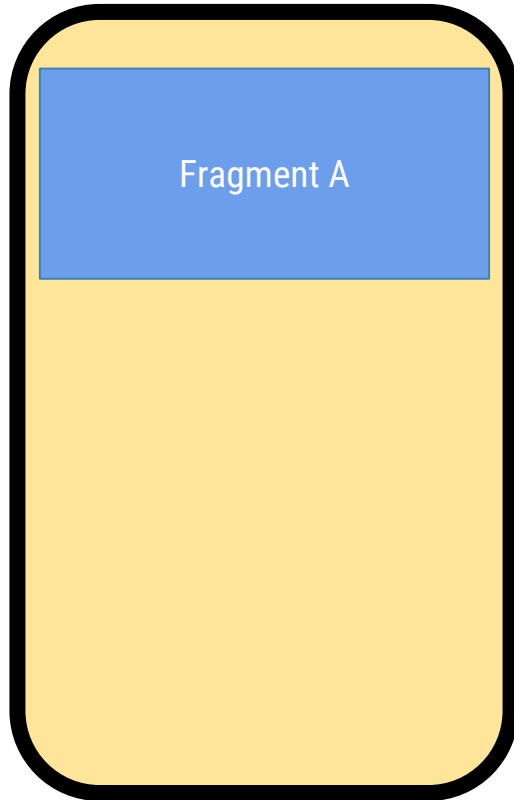
# What is a Fragment Transaction?



- Get the **FragmentManager**
- Begin a **FragmentTransaction**
- Add/Remove/Replace fragments
- Commit the transaction



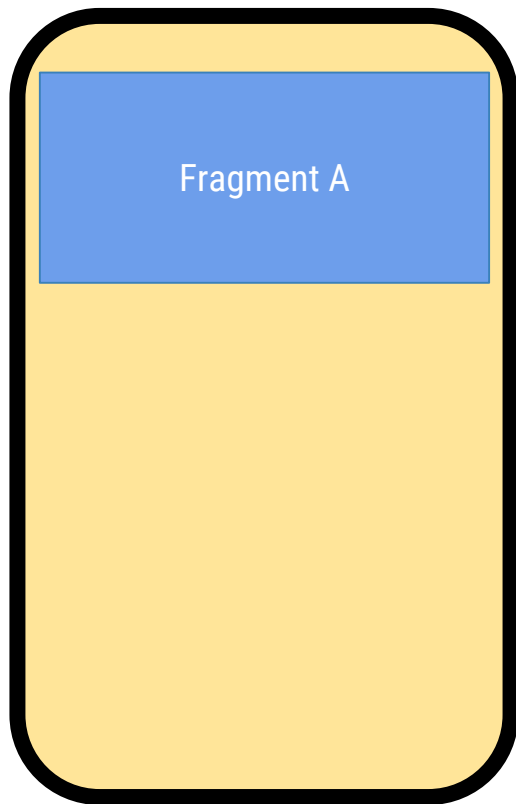
# Add a Fragment



- Create the Fragment object
- Begin the **FragmentTransaction**
- Add the Fragment
- **Commit** the transaction
- All the lifecycle methods execute inside the Fragment
- `add`(id of the layout, the fragment object, a string tag to get a reference to the fragment later)



# Remove a Fragment



- Get the Fragment object using ID or TAG
- Begin the **FragmentManager.beginTransaction()**
- Remove the fragment
- **Commit** the transaction
- All the lifecycle methods execute inside the Fragment
- `remove()`(the fragment object that you retrieved)
- Cannot use the Fragment object after removing it



# Replace a Fragment



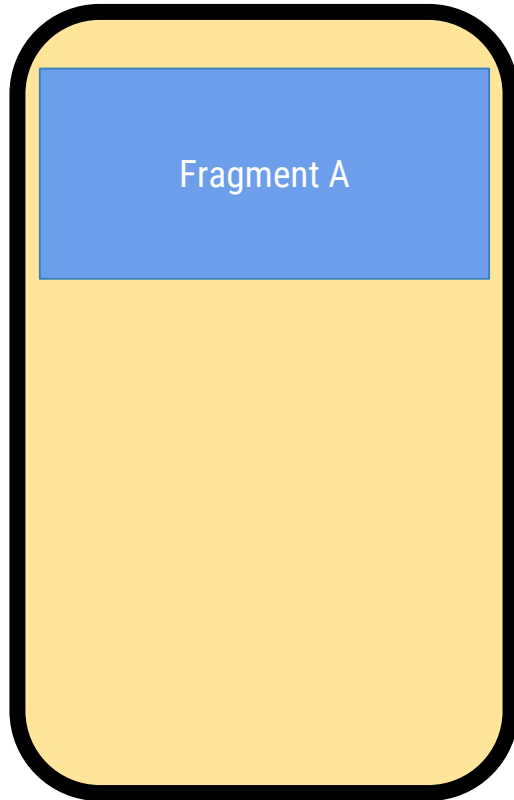
- Removes 1 fragment and adds another 1
- Calls destructive lifecycle methods on 1<sup>st</sup> Fragment and constructive ones on 2<sup>nd</sup>
- Create the new fragment object
- Specify the layout id where the old fragment is contained and call replace

```
<LinearLayout  
    android:id="@+id/container"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content">  
  
    <fragment android:id="@+id/myFrag"  
        class="sidespell.tech.FragmentA"/>  
</LinearLayout>
```

```
transaction.replace(R.id.container, newFragment);
```



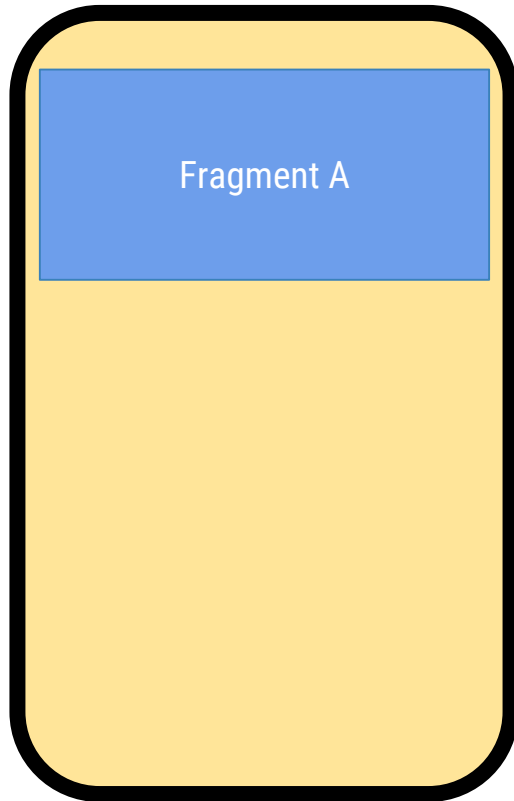
# Attach a Fragment



- Normal creation process: create the fragment object, associate with activity + create the UI
- `onAttach` is called in the Fragment to notify fragment has been attached to Activity.
- `transaction.attach()`: create the UI
- `onAttach` is not called since Fragment is already associated with Activity



# Detach a Fragment



- Normal destruction process: destroy the UI, remove association with activity, destroy fragment object
- `onDetach` is called in the Fragment to notify fragment has been disassociated to Activity.
- `transaction.detach()`: destroy the UI
- `onDetach` is not called since we are only destroying the UI





