



# Mobile Apps Development

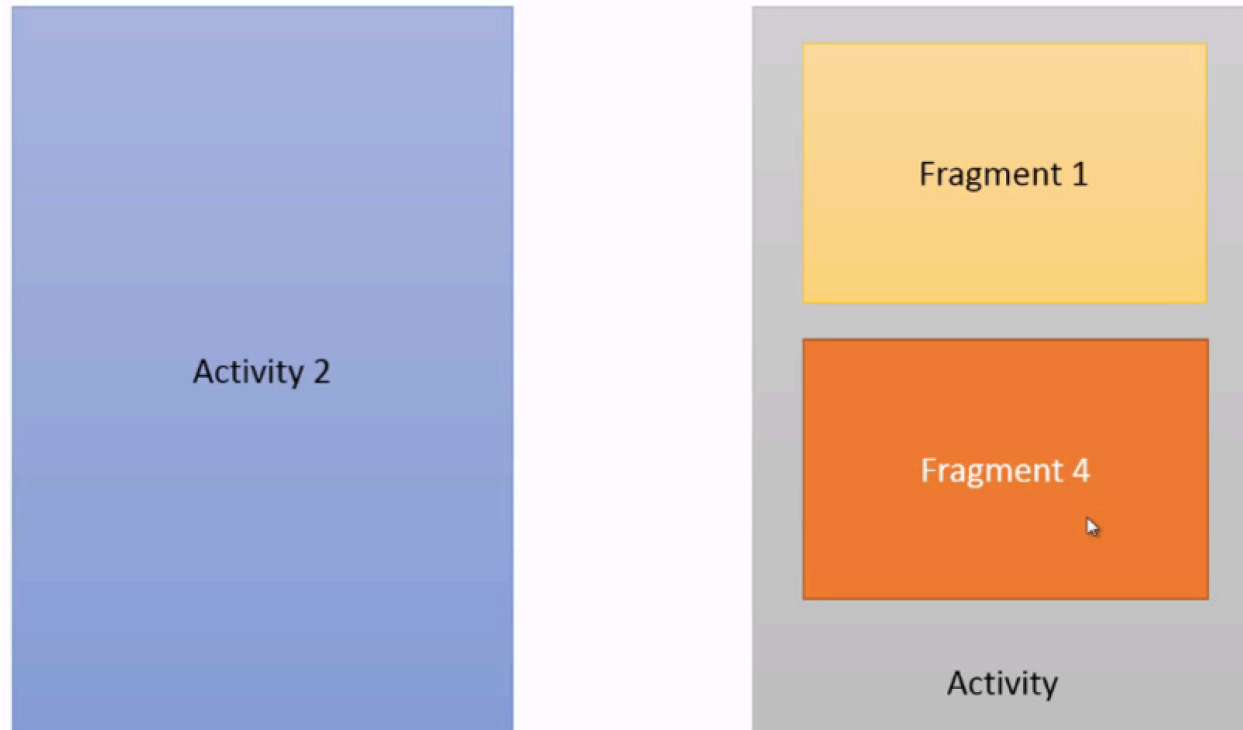
# Fragments

(Lesson 08)

**Lecturer: Vang Le**  
**Faculty of Information Technology**  
**Ton Duc Thang University**

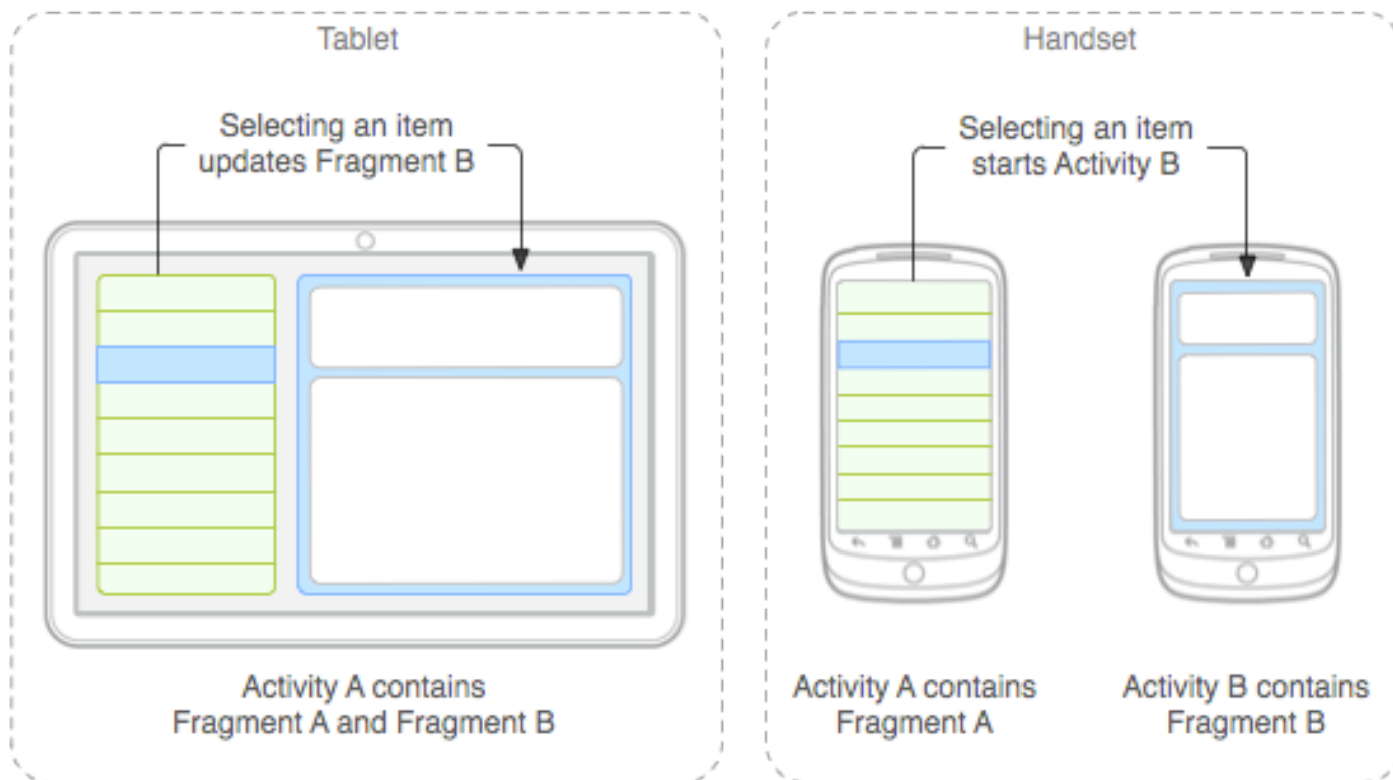
# Overview

- A fragment is a reusable class implementing a portion of an activity. A Fragment typically defines a part of a user interface.
- Fragments must be embedded in activities; they cannot run independently of activities.



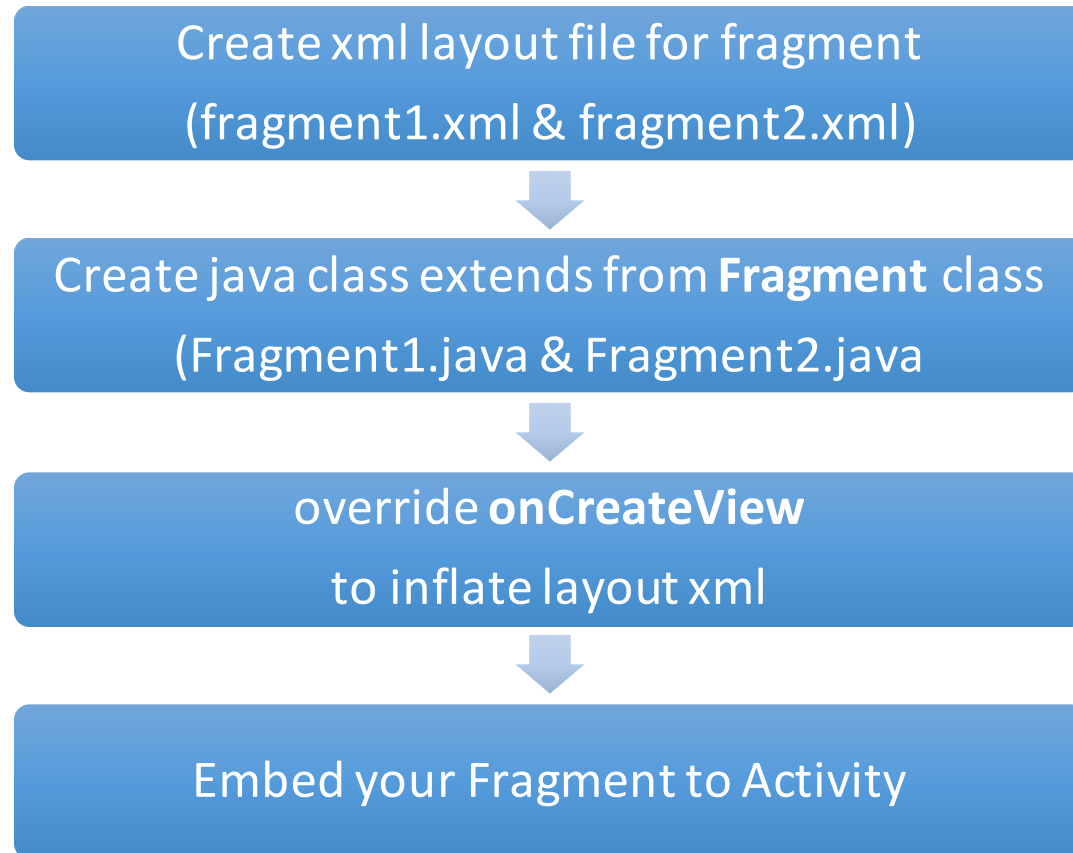
# Importance of Fragments

- Combine several fragments in a single Activity
- Reuse the same Fragment across several Activities
- Make better use of larger screen space on tablets
- Support difference layouts on portrait and landscape modes



# Usage

- A fragment, like an activity, has an XML layout file and a Java class that represents the Fragment controller.



# Create XML Layout For Fragment

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical" >

  <TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="TextView" />

  <Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button" />

</LinearLayout>
```

fragment\_foo.xml

# Create Java Class For Fragment

```
public class FooFragment extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater,
                           ViewGroup parent, Bundle savedInstanceState) {
        // Defines the xml file for the fragment
        return inflater.inflate(R.layout.fragment_foo,
                               parent, false);
    }
    // This event is triggered soon after onCreateView().
    // Any view setup should occur here. E.g., view lookups.
    @Override
    public void onViewCreated(View view, Bundle savedInstanceState) {
        // Setup any handles to view objects here
        // EditText etFoo = (EditText) view.findViewById(R.id.etFoo);
    }
}
```

# Embedding a Fragment in an Activity (Statically)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

  <fragment
    android:name="com.example.FooFragment"
    android:id="@+id/fooFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />

</LinearLayout>
```

main\_activity.xml

# Embedding a Fragment in an Activity (Dynamically)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

  <FrameLayout
    android:id="@+id/your_placeholder"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
  </FrameLayout>
</LinearLayout>
```

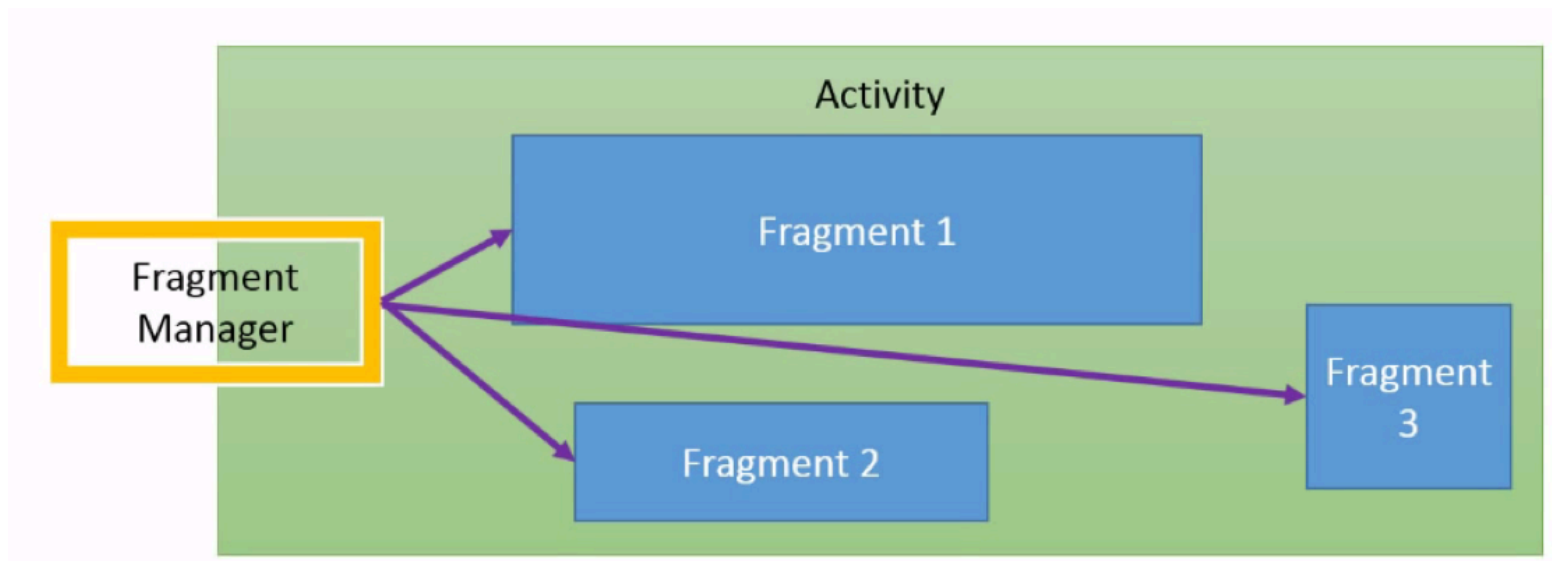
main\_activity.xml

```
FragmentManager ft;
ft = getSupportFragmentManager().beginTransaction();
ft.replace(R.id.your_placeholder, new FooFragment());
ft.commit();
```



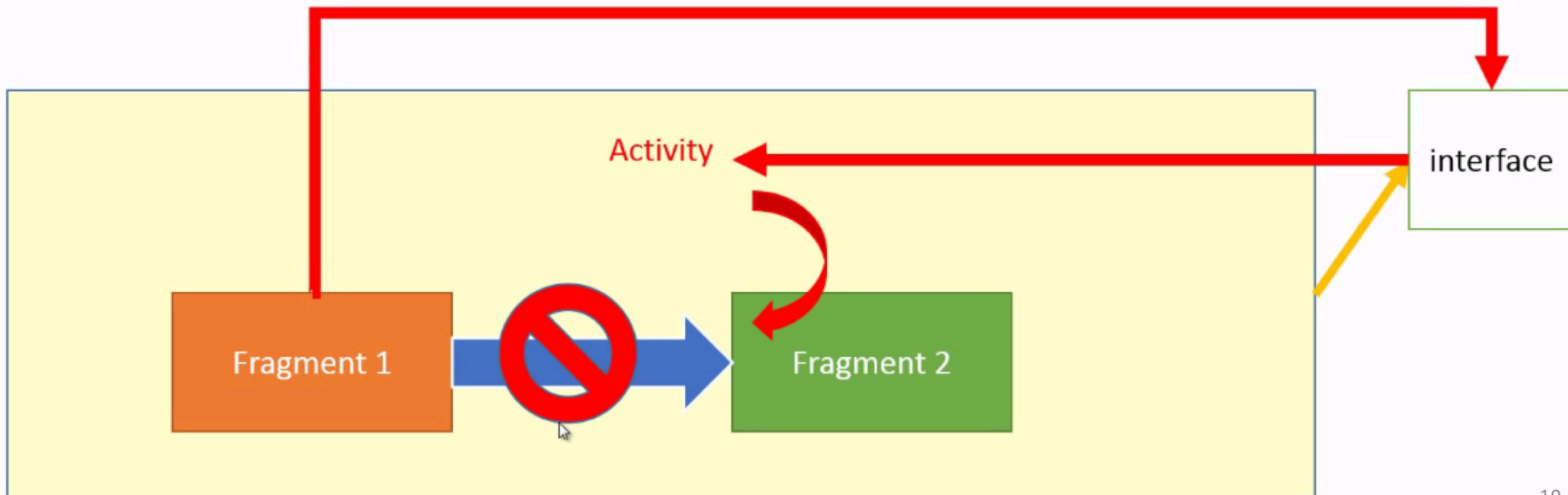
# Looking Up a Fragment Instance

- **ID** - Lookup a fragment by calling `findFragmentById` on the `FragmentManager`
- **Tag** - Lookup a fragment by calling `findFragmentByTag` on the `FragmentManager`
- **Pager** - Lookup a fragment by calling `getRegisteredFragment` on a `PagerAdapter`



# Communicating with Fragments

- Don't maintain direct references of fragments within each other
- Create an interface that contains the method which will act as an event carrier
- Let the Activity Implement the interface
- Let Fragment 1 use the interface to send the message
- Use the callback method in Activity to trigger changes in Fragment 2



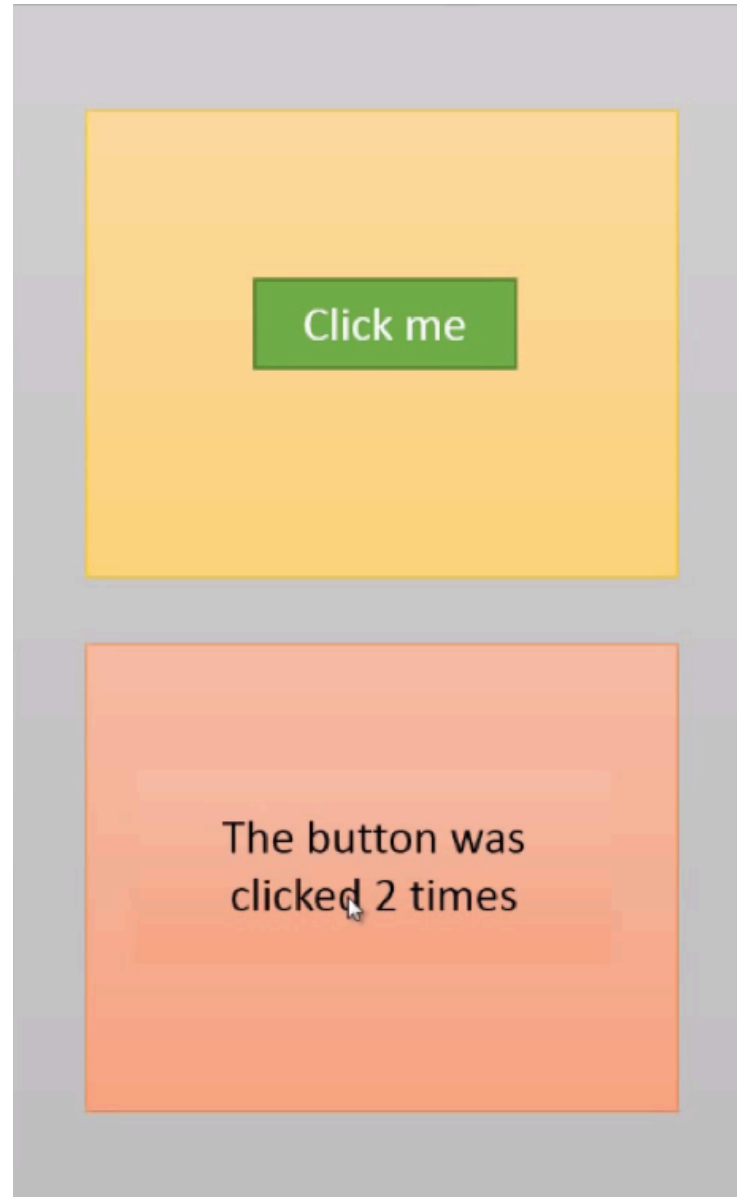
# Communicating with Fragments

Fragment1	MainActivity	Fragment2
Keep a reference to <i>MainActivity</i>	Keep a reference to both <i>Fragment1</i> , <i>Fragment2</i>	Keep a reference to <i>MainActivity</i>

# Communicating with Fragments

Fragment1	MainActivity	Fragment2
Keep a reference to <i>MainActivity</i>	Keep a reference to both <i>Fragment1</i> , <i>Fragment2</i>	Keep a reference to <i>MainActivity</i>
<pre>onAttach(Activity a){     refToMain = a; }</pre>	<pre>findFragmentById</pre>	<pre>onAttach(Activity a){     refToMain = a; }</pre>

# Exercise 01



# Navigating Between Fragments

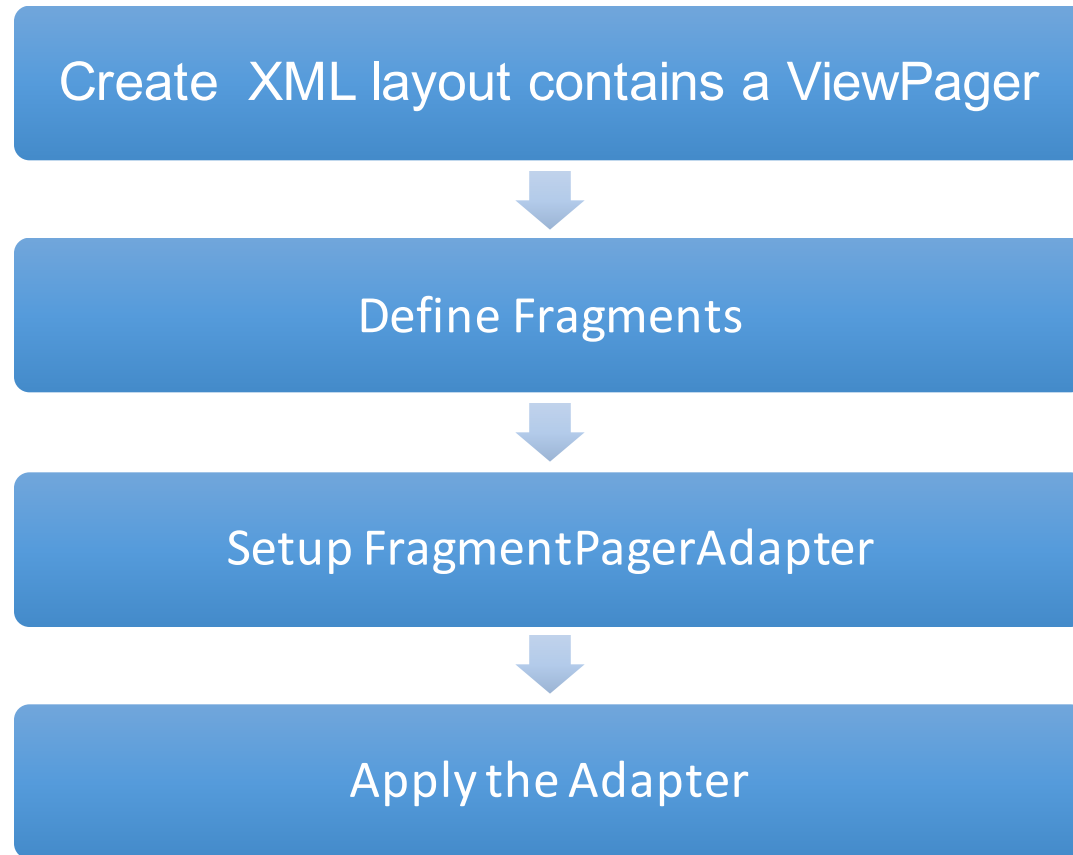
- TabLayout - Tabs at the top
- Fragment Navigation Drawer - Slide out navigation menu
- ViewPager - Swiping between fragments

# ViewPager



# Usage

- A fragment, like an activity, has an XML layout file and a Java class that represents the Fragment controller.





# Layout ViewPager

- A ViewPager is a layout which can be added to any layout XML file inside a root layout:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <android.support.v4.view.ViewPager
        android:id="@+id/vpPager"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
    </android.support.v4.view.ViewPager>
</LinearLayout>
```

# Layout ViewPager

Page 1

Page 2

Page 3

Page with title Page # 2

```
<android.support.v4.view.ViewPager
    android:id="@+id/vpPager"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <android.support.v4.view.PagerTabStrip
        android:id="@+id/pager_header"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="top"
        android:paddingBottom="4dp"
        android:paddingTop="4dp" />

</android.support.v4.view.ViewPager>
```

# FragmentPagerAdapter

Page 1

Page 2

Page 3

Page with title: Page # 2

```
public class MyPagerAdapter
    extends FragmentPagerAdapter {

    ....
    @Override
    public int getCount() {
        return ???;
    }
    @Override
    public Fragment getItem(int position) {
        return ???;
    }
    @Override
    public CharSequence getPageTitle(int position) {
        return ???;
    }
    ...
}
```

# Apply the Adapter

Page 1

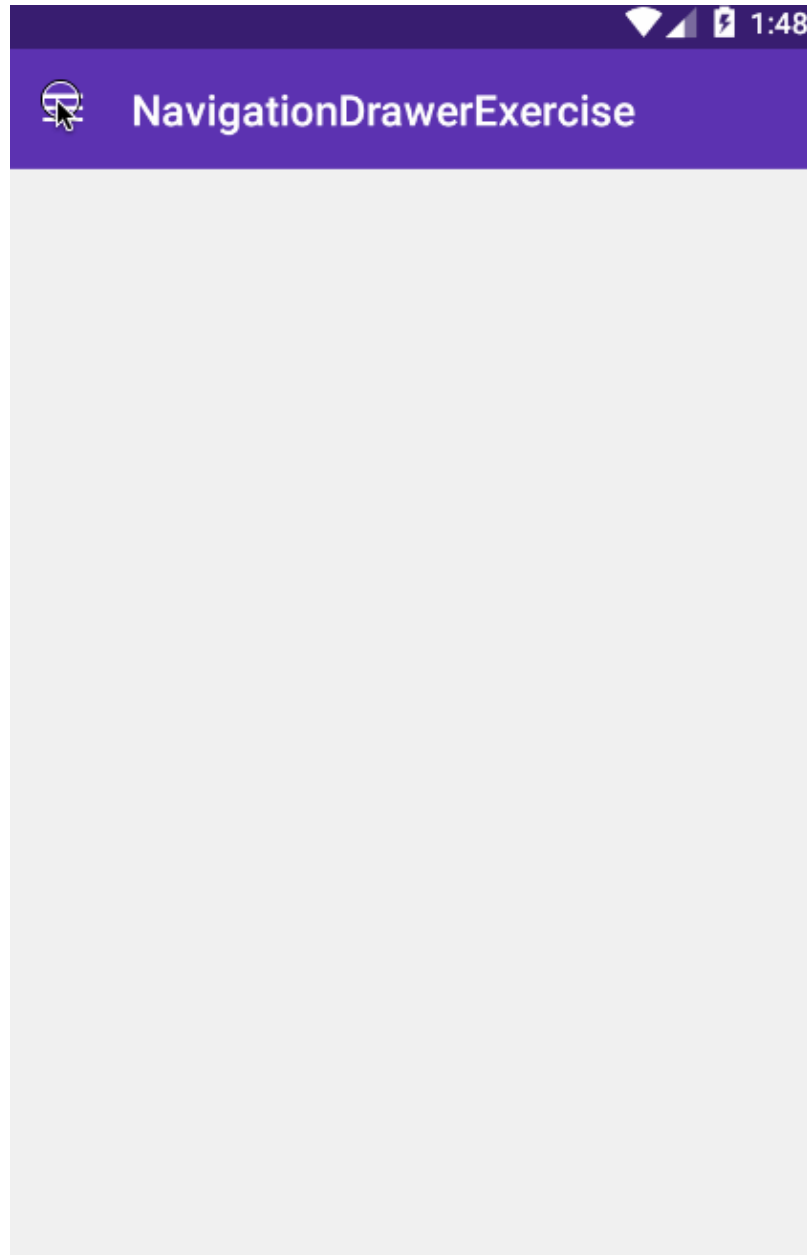
Page 2

Page 3

Page with title: Page # 2

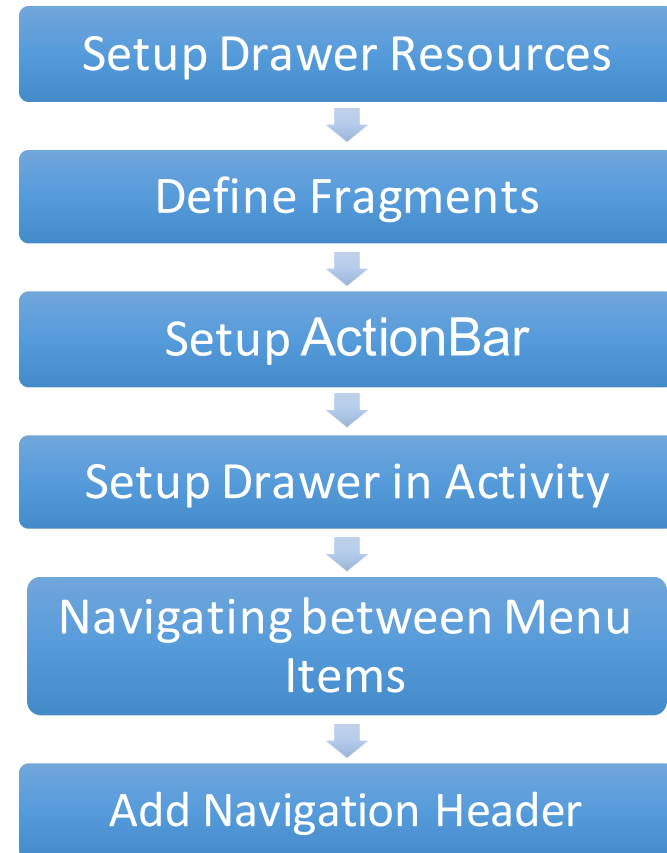
```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_home);
    ViewPager vpPager = (ViewPager) findViewById(
        R.id.vpPager);
    adapterViewPager = new MyPagerAdapter(
        getFragmentManager());
    ● → vpPager.setAdapter(adapterViewPager);
}
```

# Fragment Navigation Drawer



# Usage

- How to set-up a basic material design style drawer filled with navigation items that switch different fragments into the content area



# Setup Drawer Resources

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <group android:checkableBehavior="single">
        <item
            android:id="@+id/nav_first_fragment"
            android:icon="@drawable/ic_one"
            android:title="First" />
        <item
            android:id="@+id/nav_second_fragment"
            android:icon="@drawable/ic_two"
            android:title="Second" />
        <item
            android:id="@+id/nav_third_fragment"
            android:icon="@drawable/ic_three"
            android:title="Third" />
    </group>
</menu>
```

res/menu/drawer\_view.xml

# Setup main xml


```
<android.support.v4.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/drawer"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <FrameLayout
        android:id="@+id/flContent"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
    <android.support.design.widget.NavigationView
        android:id="@+id/nv_view"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:background="@android:color/white"
        app:menu="@menu/drawer_view" />
</android.support.v4.widget.DrawerLayout>
```

res/layout/activity\_main.xml



# Setup Home Button

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_home);  
  
    ActionBar actionBar = getSupportActionBar();  
    actionBar.setHomeButtonEnabled(true);  
    actionBar.setHomeAsUpIndicator(R.drawable.ic_reorder);  
    actionBar.setDisplayHomeAsUpEnabled(true);  
}
```



```
@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
    DrawerLayout drawerLayout;  
    drawerLayout = (DrawerLayout)findViewById(R.id.drawer);  
    if(item.getItemId() == android.R.id.home) {  
        drawerLayout.openDrawer(GravityCompat.START);  
    }  
    return true;  
}
```

# Setup NavigationView

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_home);  
    ...  
    NavigationView navView;  
    navView = (NavigationView)findViewById(R.id.nv_view);  
    ● → navView.setNavigationItemSelectedListener(this);  
}
```

```
@Override  
public boolean onNavigationItemSelected(  
    @NonNull MenuItem item) {  
    ● → drawerLayout.closeDrawer(GravityCompat.START);  
  
    return true;  
}
```

# ActionBarDrawerToggle

```
protected void onCreate(Bundle savedInstanceState) {  
    toolbar = (Toolbar) findViewById(R.id.toolbar);  
    setSupportActionBar(toolbar);  
  
    mDrawer = (DrawerLayout) findViewById(R.id.drawer);  
    drawerToggle = ActionBarDrawerToggle(this,  
                                         mDrawer,  
                                         toolbar,  
                                         "open drawer",  
                                         "close drawer");  
  
    mDrawer.addDrawerListener(drawerToggle);  
    drawerToggle.syncState();  
}
```

# ActionBarDrawerToggle

```
<android.support.v4.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/drawer"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <FrameLayout
        android:id="@+id/flContent"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
    <android.support.design.widget.NavigationView
        android:id="@+id/nv_view"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:background="@android:color/white"
        app:menu="@menu/drawer_view" />
</android.support.v4.widget.DrawerLayout>
```

res/layout/activity\_main.xml

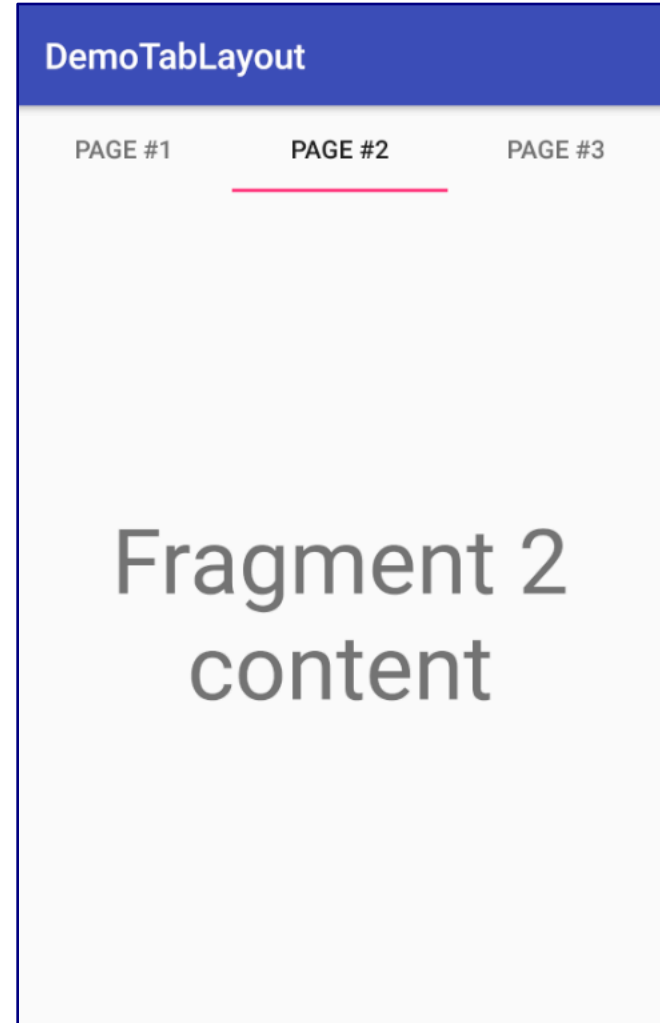
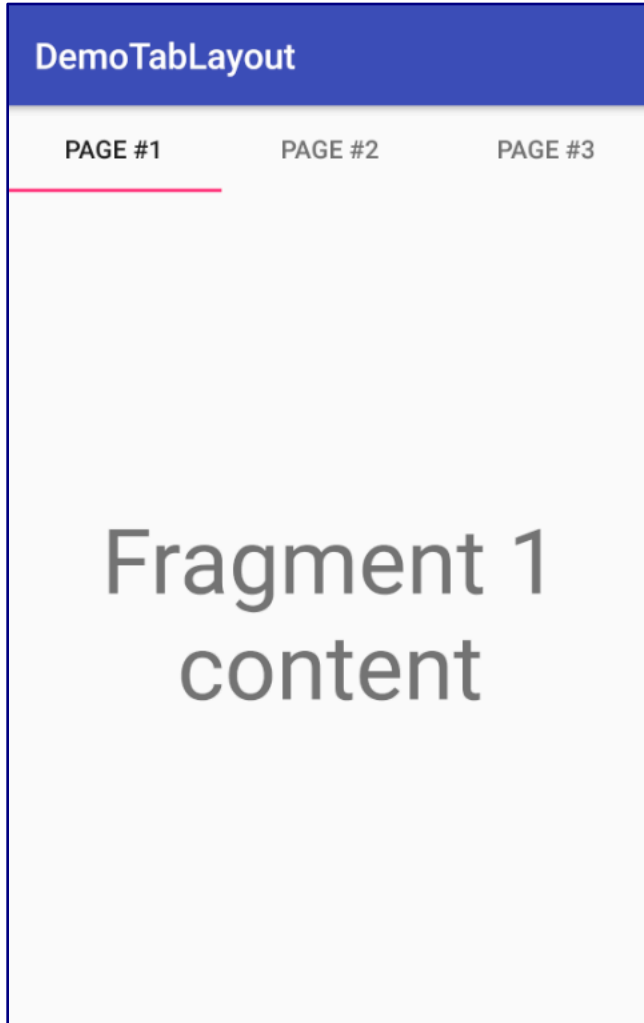
# ActionBarDrawerToggle

```
<android.support.v4.widget.DrawerLayout
    ...>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <include
            layout="@layout/toolbar"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />
        <FrameLayout
            android:id="@+id/flContent"
            android:layout_width="match_parent"
            android:layout_height="match_parent" />
    </LinearLayout>
    <android.support.design.widget.NavigationView
        ....>
</android.support.v4.widget.DrawerLayout>
```

# ActionBarDrawerToggle

```
<android.support.v7.widget.Toolbar  
  
xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    android:id="@+id/toolbar"  
    android:layout_height="wrap_content"  
    android:layout_width="match_parent"  
    android:fitsSystemWindows="true"  
    android:minHeight="?attr/actionBarSize"  
    app:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"  
    android:background="?attr/colorPrimaryDark">  
</android.support.v7.widget.Toolbar>
```

# TabLayout



# TabLayout

```
compile "com.android.support:design:25.3.0"
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout ... >

    <android.support.v4.view.ViewPager
        android:id="@+id/viewPager"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <android.support.design.widget.TabLayout
            android:id="@+id/sliding_tabs"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"/>

    </android.support.v4.view.ViewPager>
</LinearLayout>
```

res/layout/activity\_main.xml



# TabLayout

**@Override**

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    ViewPager viewPager = (ViewPager)findViewById(  
        R.id.viewPager);  
    MyPagerAdapter adapter = new MyPagerAdapter(  
        getSupportFragmentManager());  
    viewPager.setAdapter(adapter);  
  
    TabLayout tabLayout = (TabLayout)findViewById(  
        R.id.sliding_tabs);  
    tabLayout.setupWithViewPager(viewPager);  
}
```

# Exercise 02

DemoMasterDetailFlow
Email subject number 1
Email subject number 2
Email subject number 3
Email subject number 4
Email subject number 5
Email subject number 6
Email subject number 7
Email subject number 8
Email subject number 9
Email subject number 10

← DemoMasterDetailFlow
Email subject number 1 Email content number 1 sender 1 29/03/2017 21:01:52

# Exercise 02

DemoMasterDetailFlow	
Email subject number 1	Email subject number 2 Email content number 2 sender 2
Email subject number 2	30/03/2017 14:04:53
Email subject number 3	
Email subject number 4	
Email subject number 5	
Email subject number 6	
Email subject number 7	
Email subject number 8	
Email subject number 9	
Email subject number 10	
Email subject number 11	
Email subject number 12	
Email subject number 13	
Email subject number 14	

## Exercise 02



Watch Now



My Library



My Wishlist



Shop