



안드로이드 앱 프로그래밍

Chapter 06

프래그먼트 이해하기



# 이번 장에서는 무엇을 다룰까요?



프래그먼트로 부분 화면을 만드는 방법에 대해 알고 싶어요.

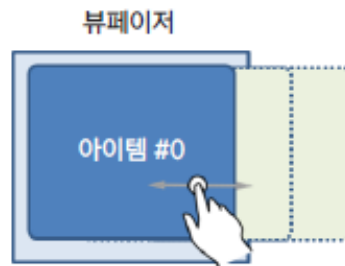
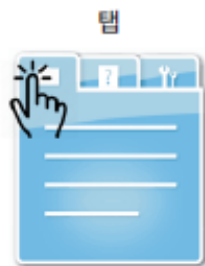
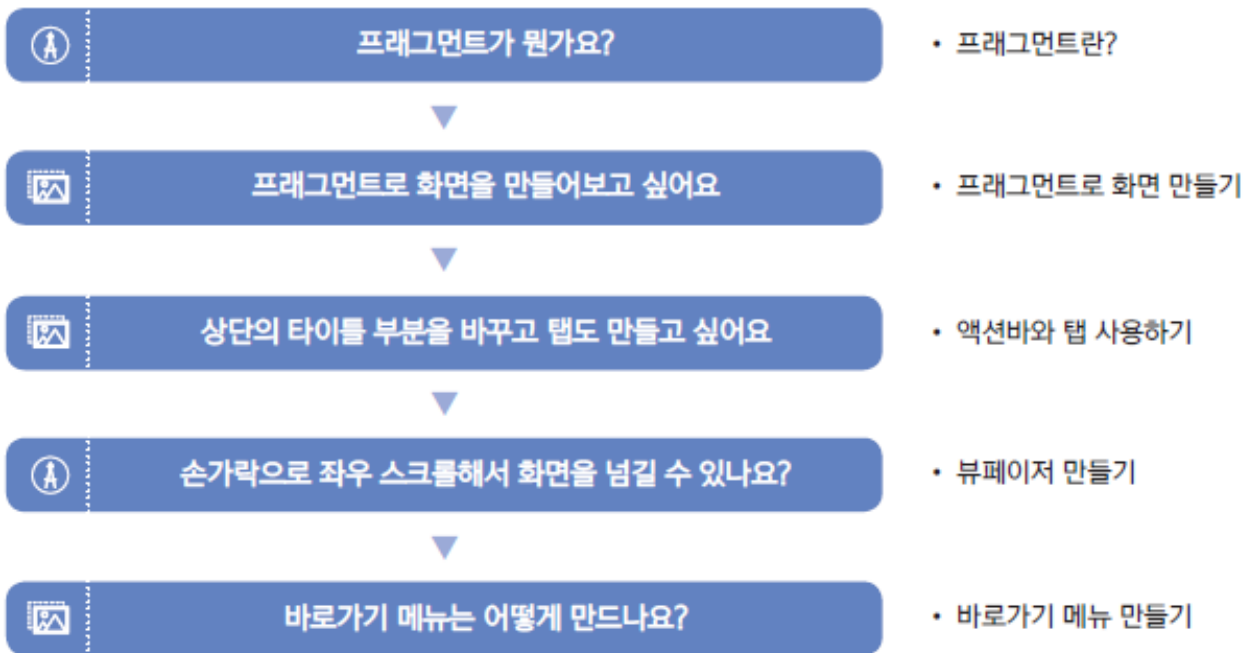


- 프래그먼트에 대해 알아보까요?
- 프래그먼트로 구성된 화면을 만들어볼까요?
- 액션바와 탭을 사용하는 방법을 알아보까요?
- 손가락으로 좌우 스크롤하여 화면을 넘기는 방법을 알아보까요?
- 바로가기 메뉴를 어떻게 붙이는지 알아보까요?





# 이번 장에서는 무엇을 다룰까요?





## 강의 주제

### 프래그먼트를 사용하는 방법에 대한 이해



- 1 프래그먼트란?
- 2 프래그먼트로 화면 만들기
- 3 액션바 사용하기
- 4 상단 탭과 하단 탭 만들기
- 5 뷰페이저 만들기
- 6 바로가기 메뉴 만들기

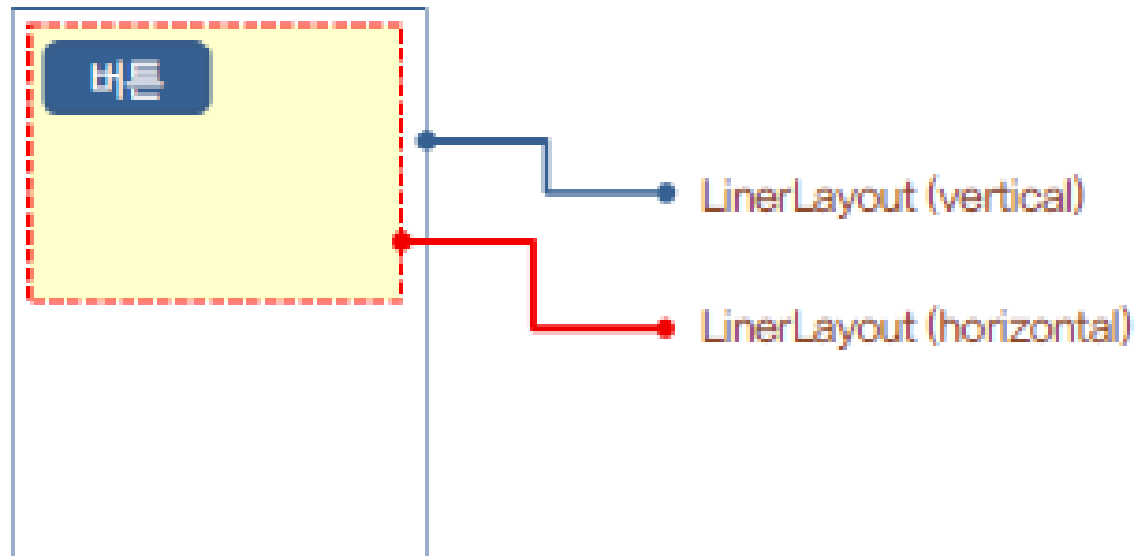
1.

프래그먼트란?



# 레이아웃 안에 레이아웃을 넣는 방식

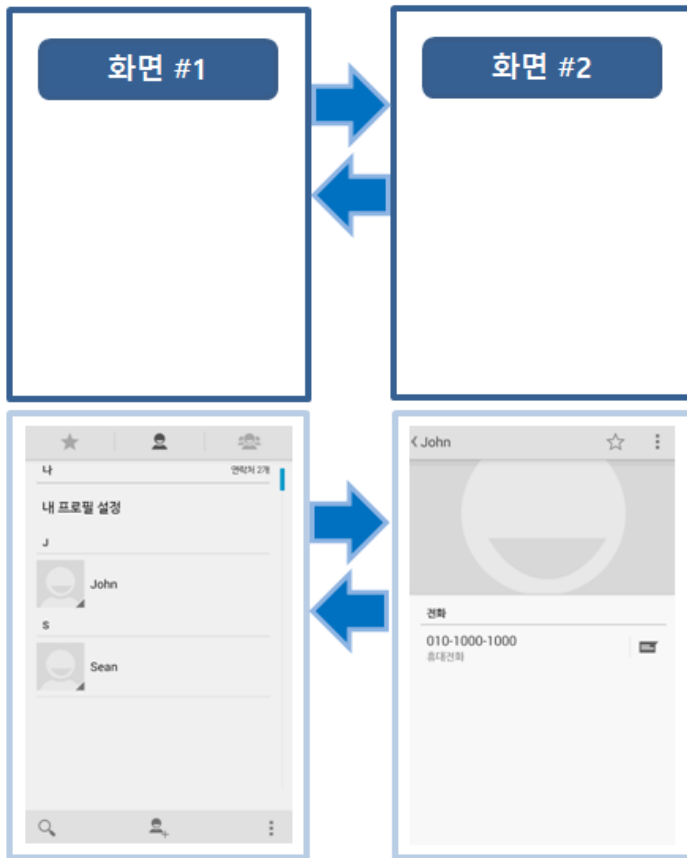
- 레이아웃 안에 부분 화면으로 레이아웃을 넣을 수 있음





# 한 화면에 여러 부분화면을 넣는 방식

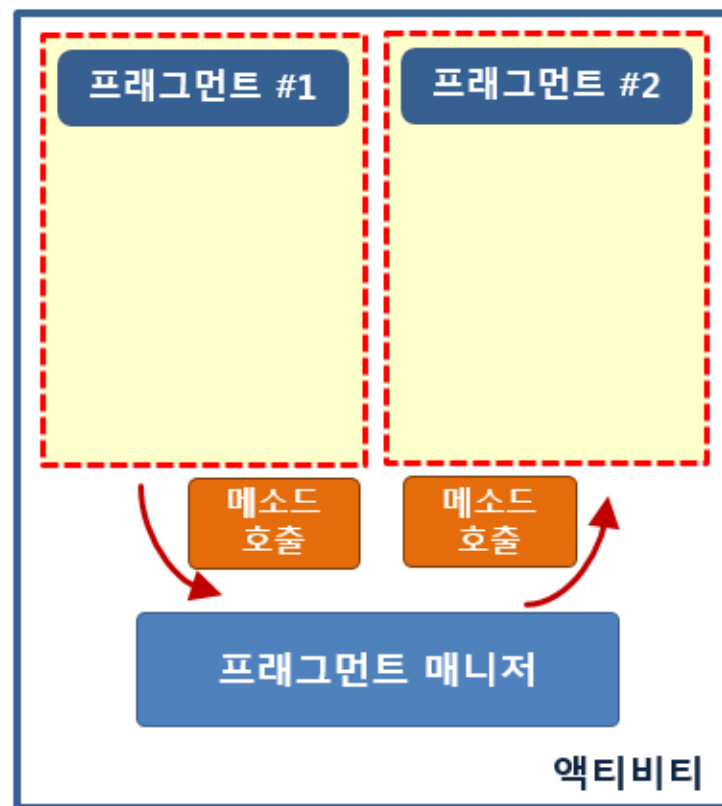
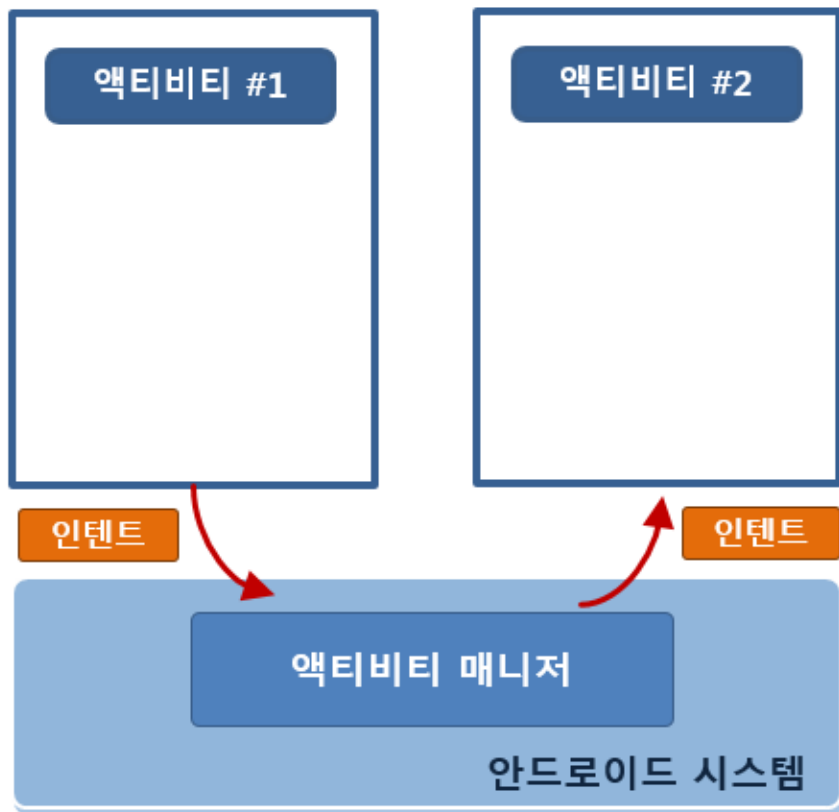
- 태블릿과 같이 큰 화면에서는 한 화면에 여러 부분화면을 넣는 것이 효율적임





# 부분화면을 독립적으로 만들기

- 프래그먼트는 부분화면을 독립적으로 만들어주며 액티비티를 그대로 본떠 만든 것임

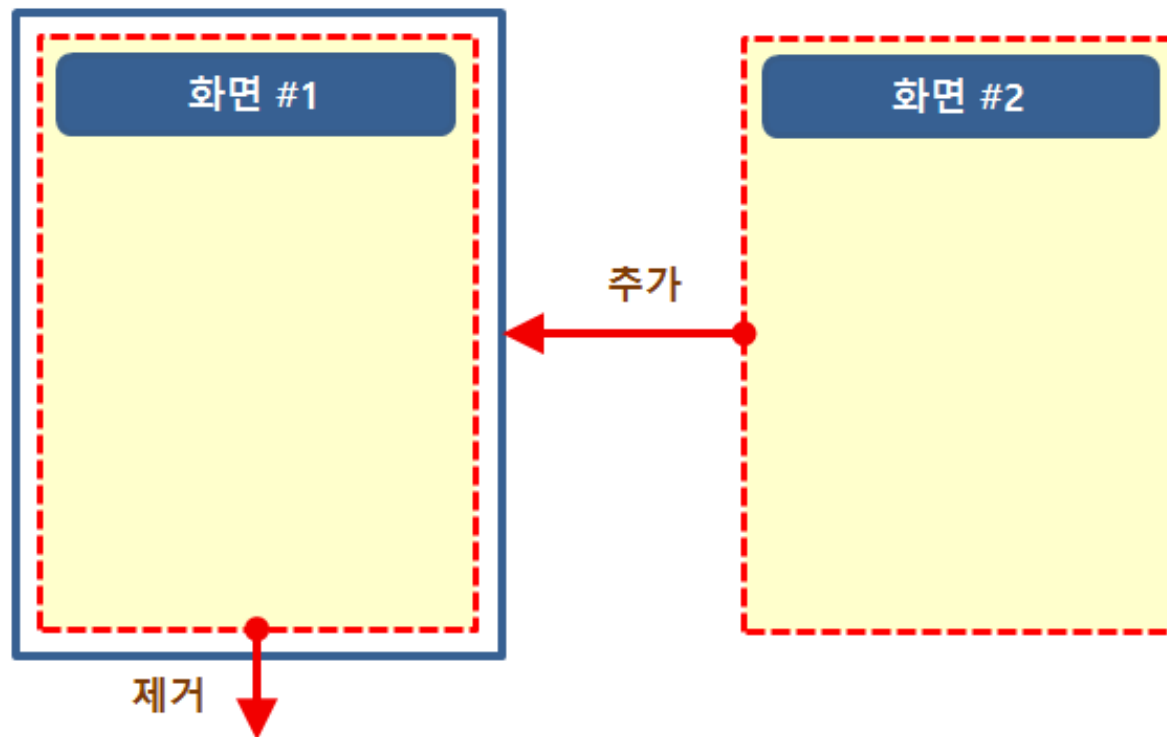






# 프래그먼트를 이용해 화면 전환하기

- 하나의 액티비티에 하나의 프래그먼트를 전체 화면으로 보여주면 시스템과 관계 없이 전체 화면을 전환하는 효과를 만들 수 있음

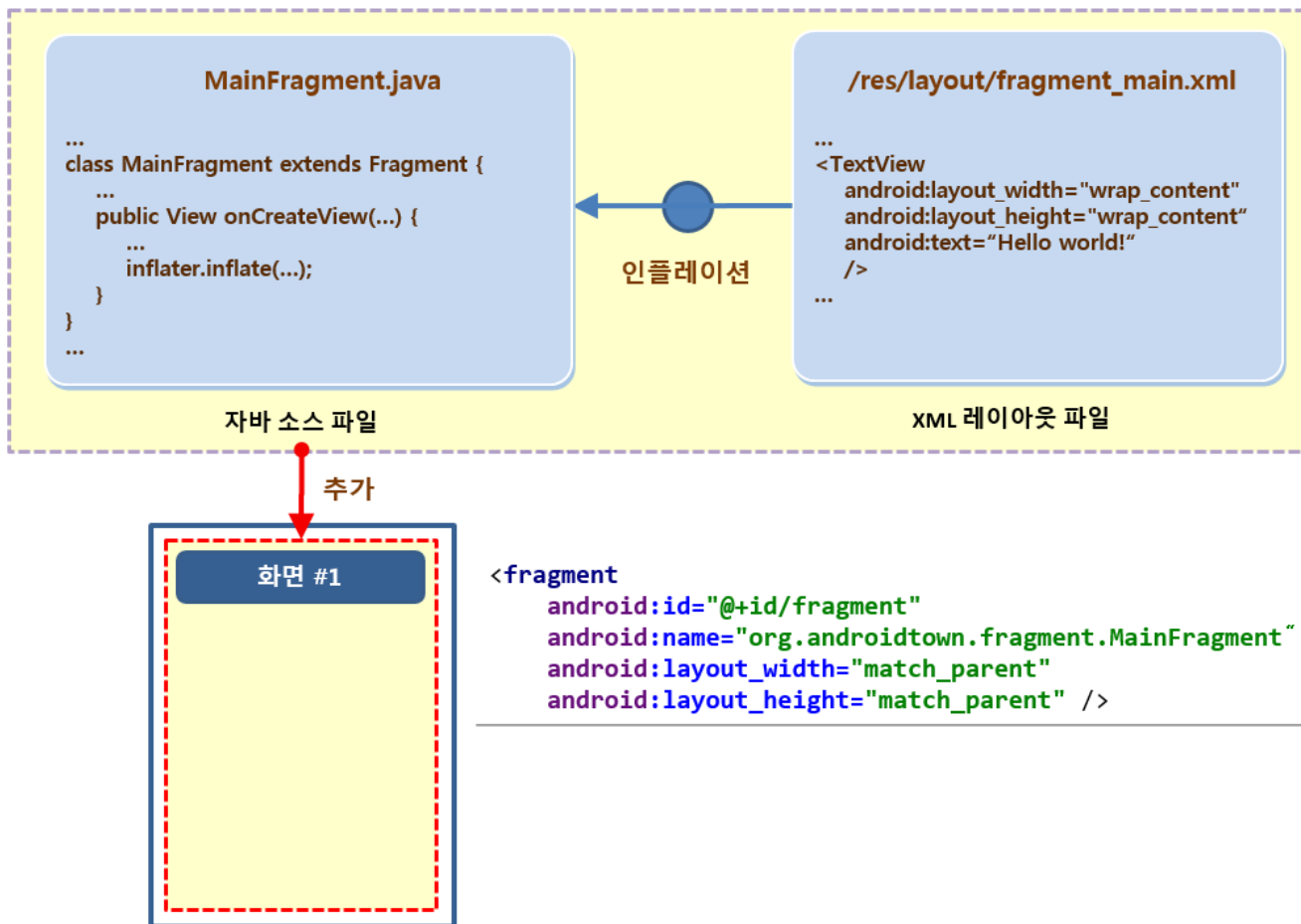




# 프래그먼트를 화면에 추가하기

- 프래그먼트는 액티비티 위에 올라가야 프래그먼트로 동작함
- 인플레이션을 위한 별도의 콜백 메소드를 제공함

하나의 프래그먼트





# 프래그먼트 만들어 화면에 추가하기

## • 프래그먼트 추가

The screenshot shows the Android Studio interface. On the left, the 'Project' and 'Resource Manager' tabs are visible. The 'Project' tab shows the project structure with 'app' as the selected module. The 'Resource Manager' tab shows the 'org.techtown.fragment' package. The 'New' menu is open, and 'Fragment' is selected. The 'Configure Component' dialog box is displayed on the right. It has a title bar 'New Android Component' and a close button. The main content area has a header 'Configure Component' with the Android Studio logo. Below the header, it says 'Creates a blank fragment that is compatible back to API level 4.' There is a preview of a fragment layout on the left. On the right, there are fields for 'Fragment Name' (MainFragment), 'Fragment Layout Name' (fragment\_main), 'Source Language' (Java), and 'Target Source Set' (main). There are also checkboxes for 'Create layout XML?' (checked), 'Include fragment factory methods?' (checked), and 'Include interface callbacks?' (checked). At the bottom, there are buttons for 'Previous', 'Next', 'Cancel', and 'Finish'.

Fragment Name: MainFragment

Fragment Layout Name: fragment\_main

Source Language: Java

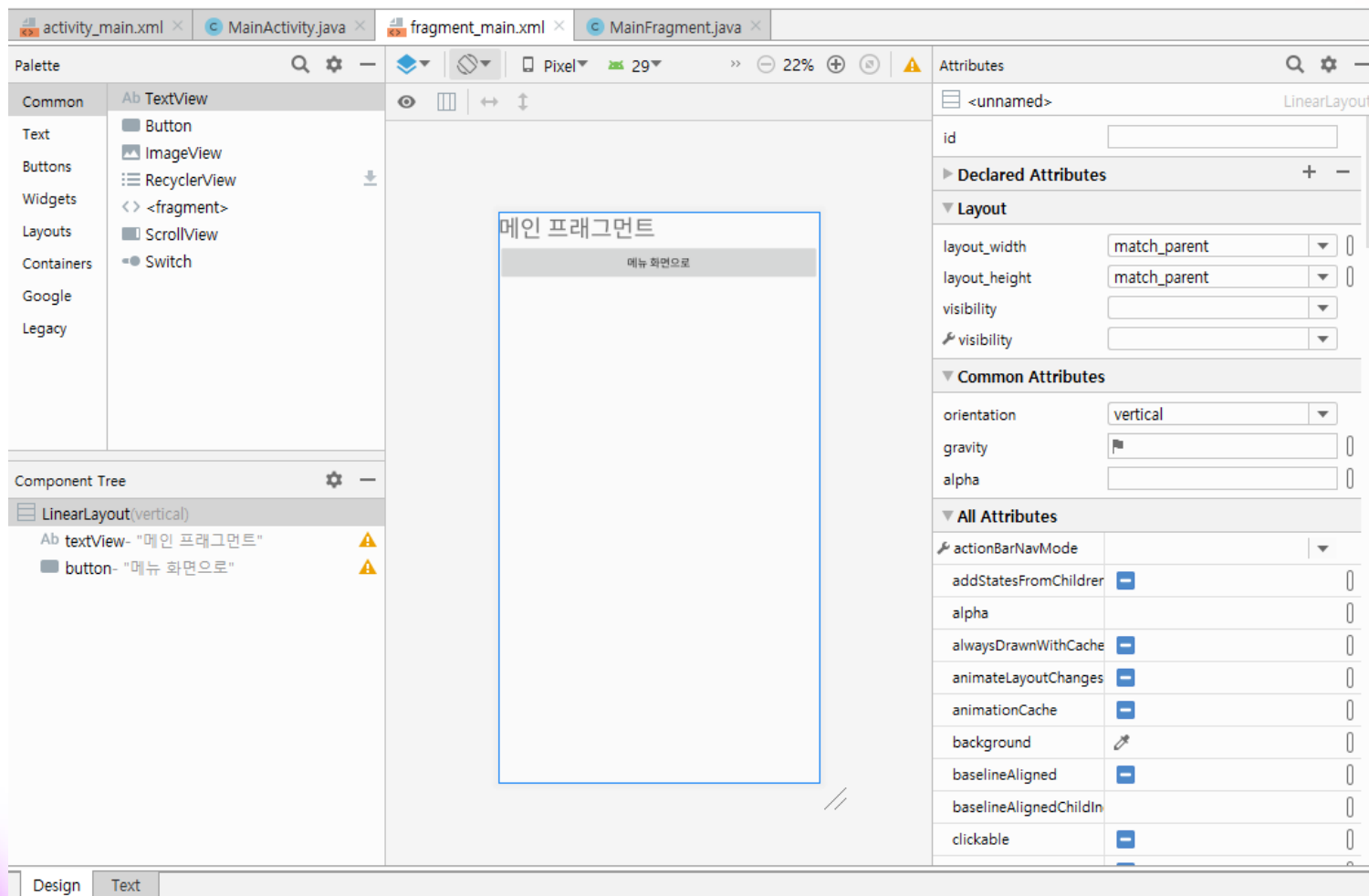
Target Source Set: main

Buttons: Previous, Next, Cancel, Finish



# 프래그먼트 만들어 화면에 추가하기

- 프래그먼트를 위한 화면 구성





# 프래그먼트 만들어 화면에 추가하기

- MainFragment.java 파일 만들고 프래그먼트 상속

참조파일 SampleFragment>/app/java/org.techtown.fragment/MainFragment.java

```
public class MainFragment extends Fragment {  
  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
        return inflater.inflate(R.layout.fragment_main, container, false);  
    }  
}
```



# 프래그먼트 만들어 화면에 추가하기

- 메인 액티비티의 XML 레이아웃에 추가

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/container"
    >

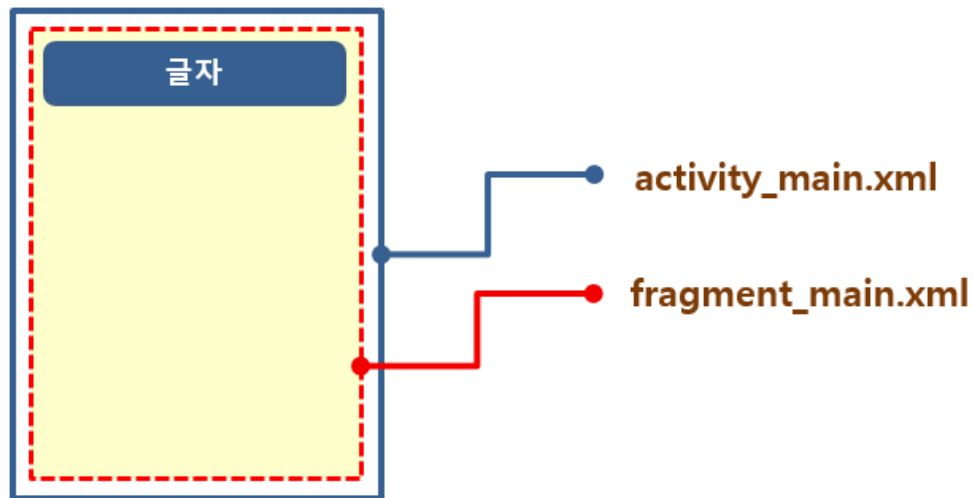
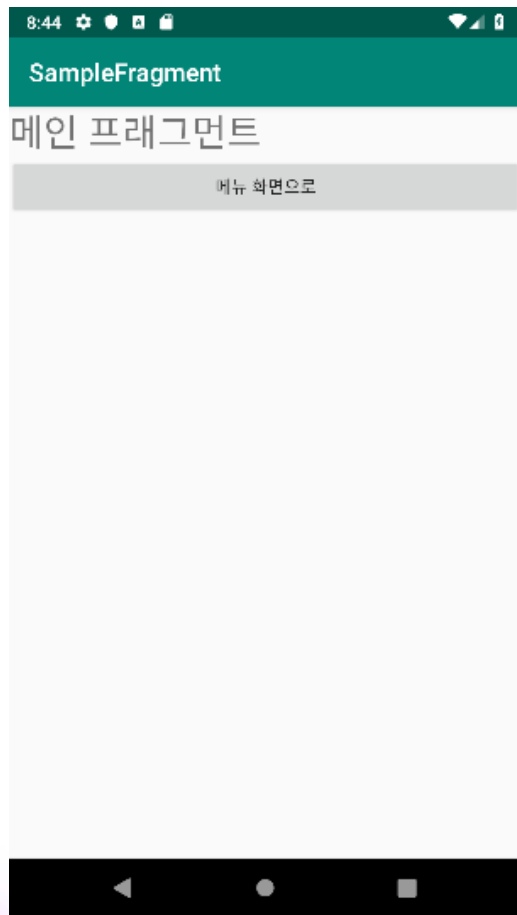
    <fragment
        android:id="@+id/mainFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:name="org.techtown.fragment.MainFragment" />

</RelativeLayout>
```



# 앱 실행 결과

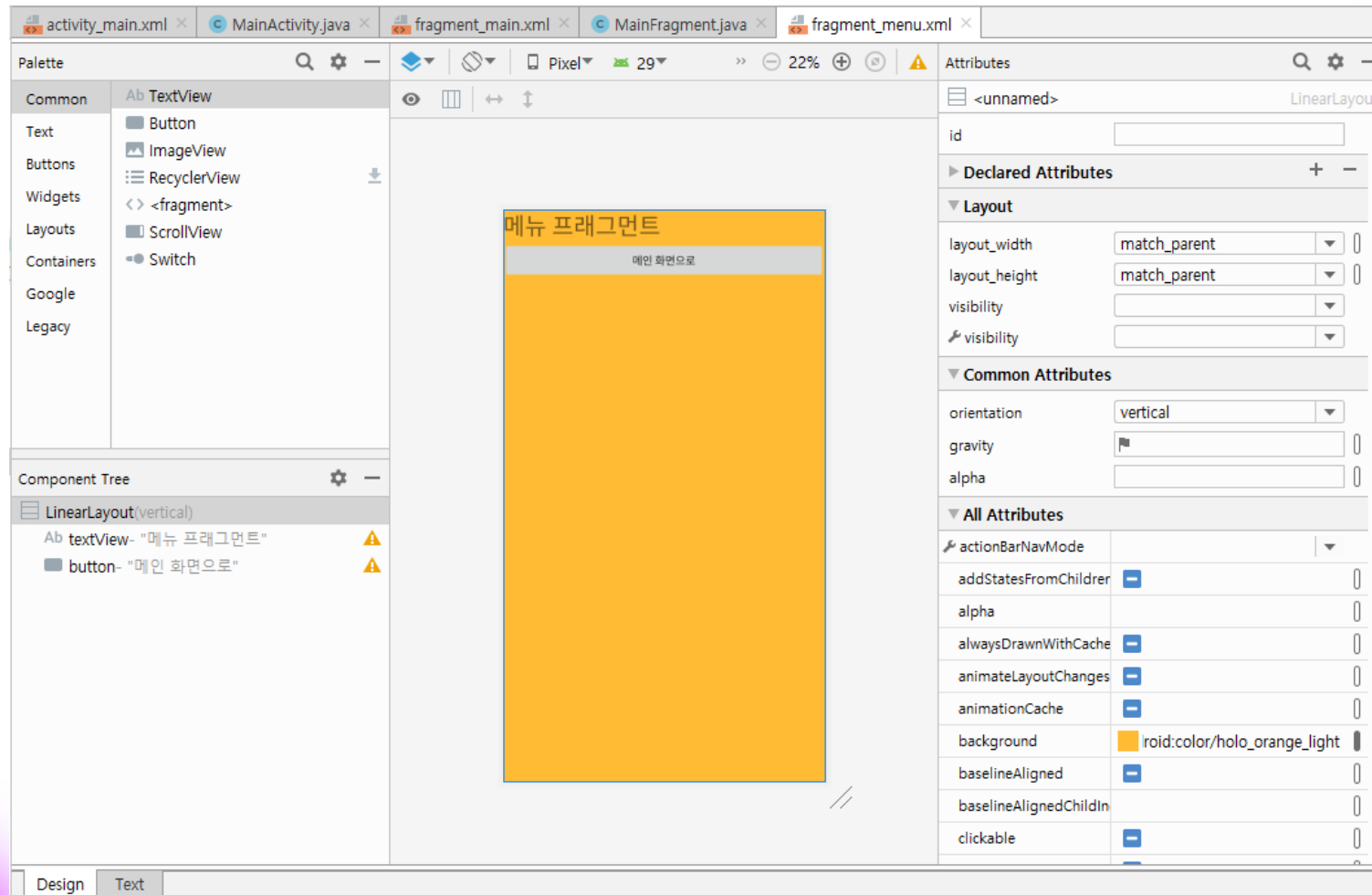
- 프래그먼트가 부분화면으로 추가됨





# 코드에서 프래그먼트 추가하기

- 메뉴 프래그먼트를 위한 XML 레이아웃 파일 만들고 화면 구성







# 코드에서 프래그먼트 추가하기

- 프래그먼트 소스 코드 입력

참조파일 SampleFragment>/app/java/org.techtown.fragment/MainFragment.java

```
public class MainActivity extends Fragment {

    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container,
                             @Nullable Bundle savedInstanceState) {
        ViewGroup rootView = (ViewGroup) inflater.inflate(R.layout.fragment_main,
                                                         container, false);

        Button button = rootView.findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                MainActivity activity = (MainActivity) getActivity();
                activity.onFragmentChanged(0);
            }
        });

        return rootView;
    }
}
```



# 코드에서 프래그먼트 추가하기

- 메인 액티비티 코드에서 프래그먼트 매니저 사용

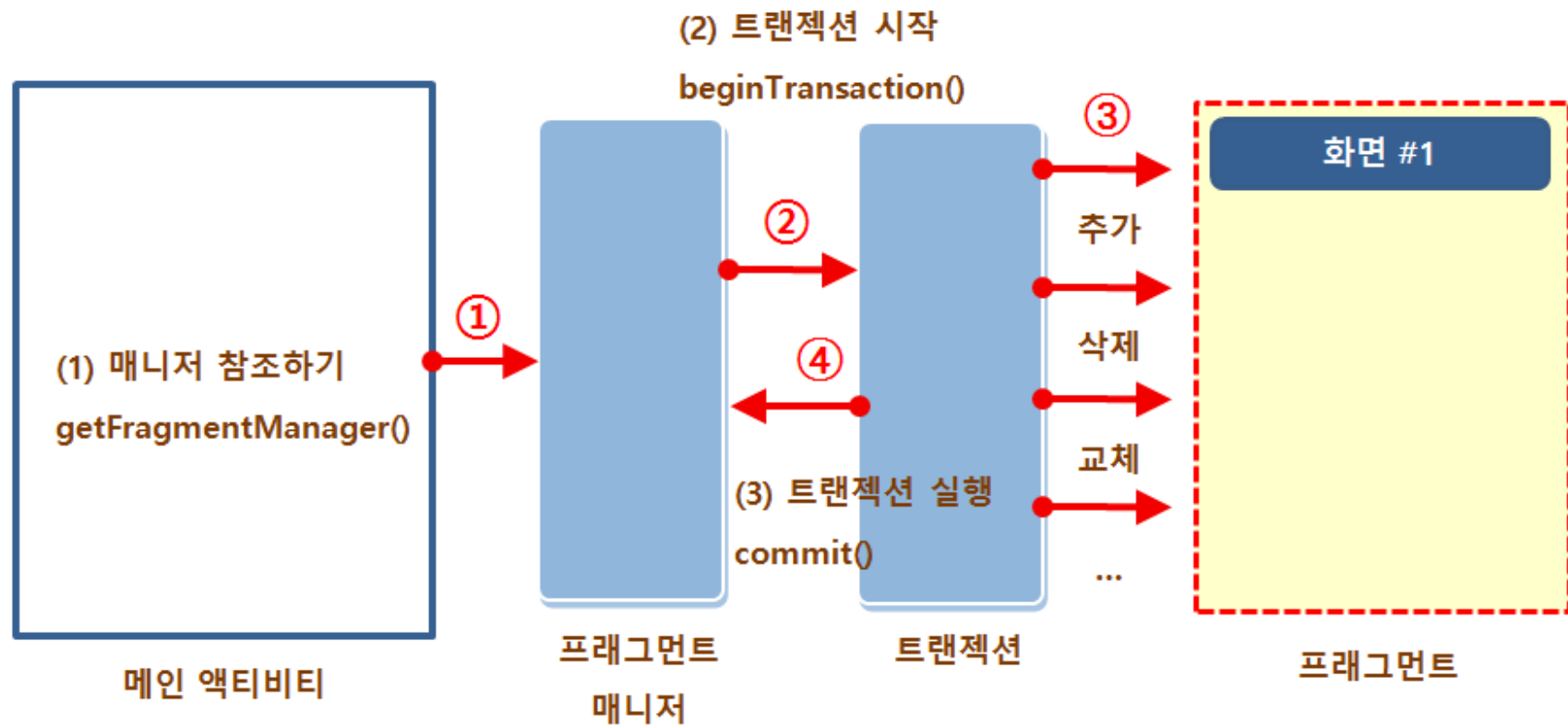
참조파일 SampleFragment>/app/java/org.techtown.fragment/MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
    MainFragment mainFragment;  
    MenuFragment menuFragment;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        mainFragment = (MainFragment) getSupportFragmentManager().findFragmentById(R.id.mainFragment);  
        menuFragment = new MenuFragment();  
    }  
  
    public void onFragmentChanged(int index) {  
        if (index == 0) {  
            getSupportFragmentManager().beginTransaction().replace(R.id.container,  
                                                                    menuFragment).commit();  
        } else if (index == 1) {  
            getSupportFragmentManager().beginTransaction().replace(R.id.container,  
                                                                    mainFragment).commit();  
        }  
    }  
}
```



# 코드에서 프래그먼트 추가하기

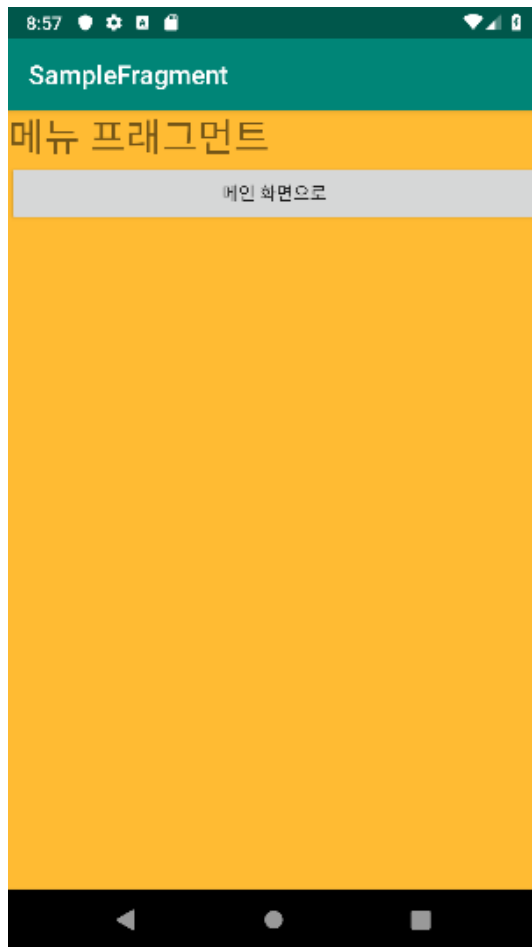
- 프래그먼트가 액티비티의 프래그먼트 매니저를 통해 의사소통하는 방식





## 앱 실행 결과

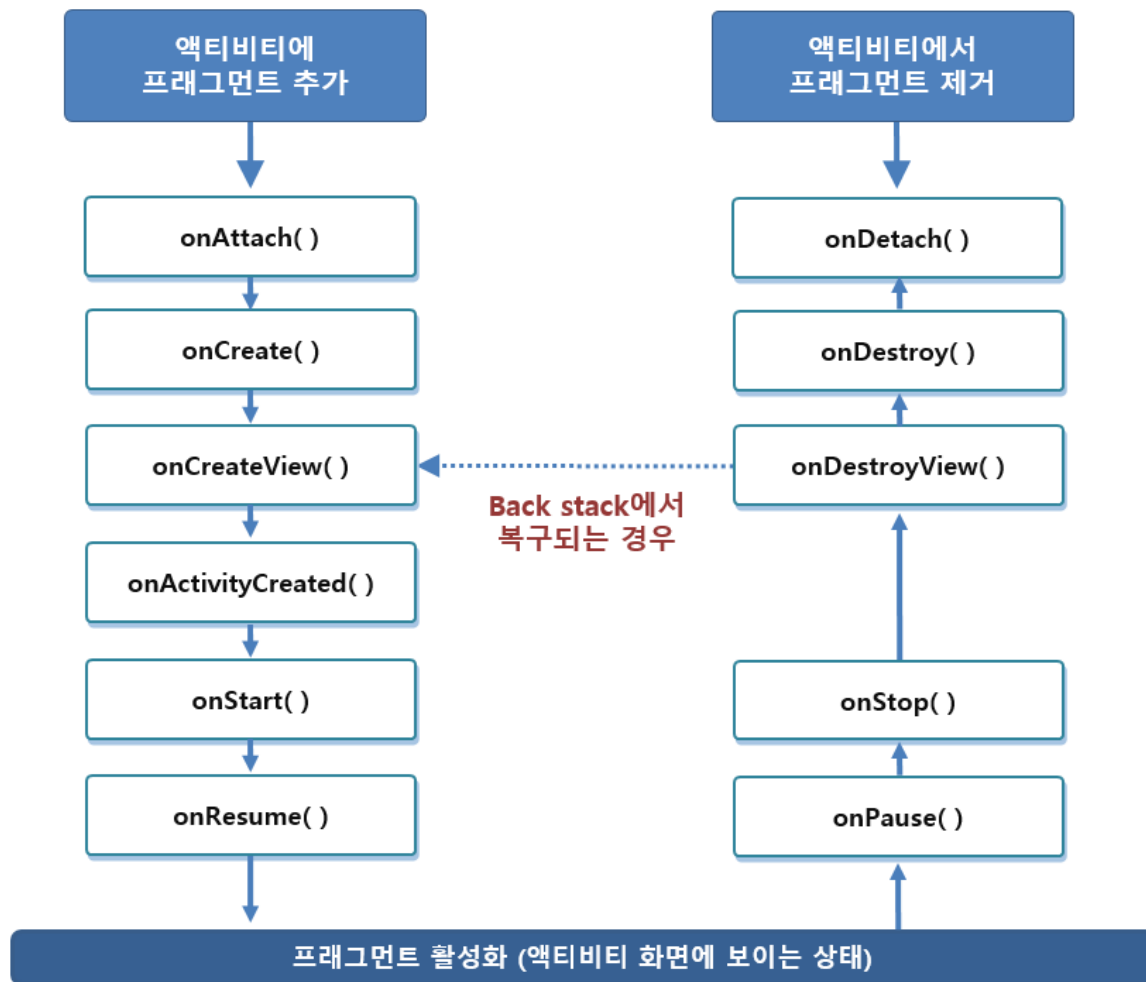
- 프래그먼트 안의 버튼 클릭 시 액티비티를 통해 프래그먼트 전환





# 프래그먼트의 수명주기

- 독립적인 상태 유지를 위해 수명주기 메소드를 제공함



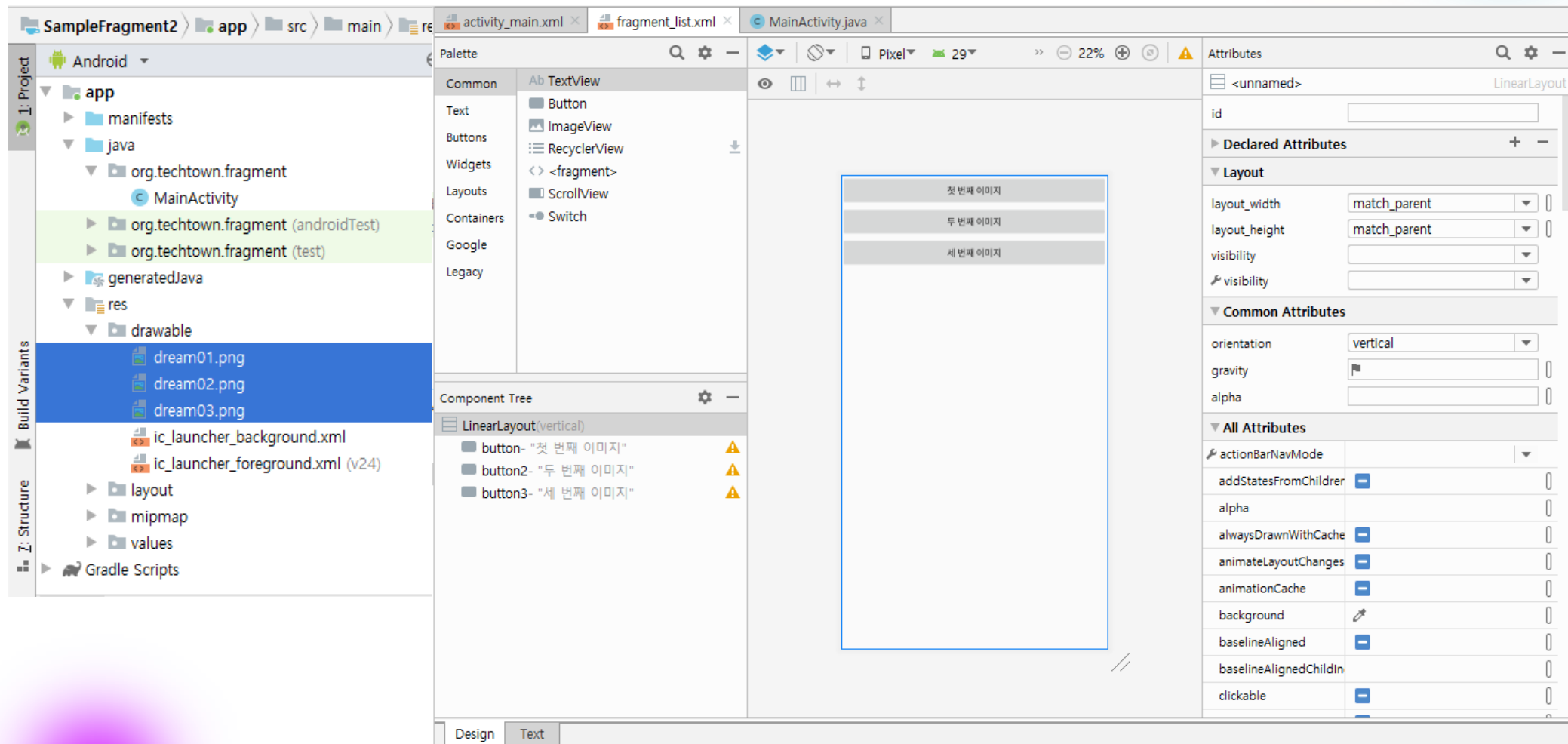
2.

프래그먼트로 화면 만들기



# 두 개의 프래그먼트로 구성된 이미지 뷰어

- 이미지 복사한 후 프래그먼트 구성





# 두 개의 프래그먼트로 구성된 이미지 뷰어

- 첫번째 프래그먼트에 코드 추가

참조파일 SampleFragment2>/app/java/org.techtown.fragment/ListFragment.java

```
public class ListFragment extends Fragment {

    public static interface ImageSelectionCallback {
        public void onImageSelected(int position);
    }

    public ImageSelectionCallback callback;

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);

        if (context instanceof ImageSelectionCallback) {
            callback = (ImageSelectionCallback) context;
        }
    }

    @Nullable
    @Override
```





# 두 개의 프래그먼트로 구성된 이미지 뷰어

- 첫번째 프래그먼트에 코드 추가

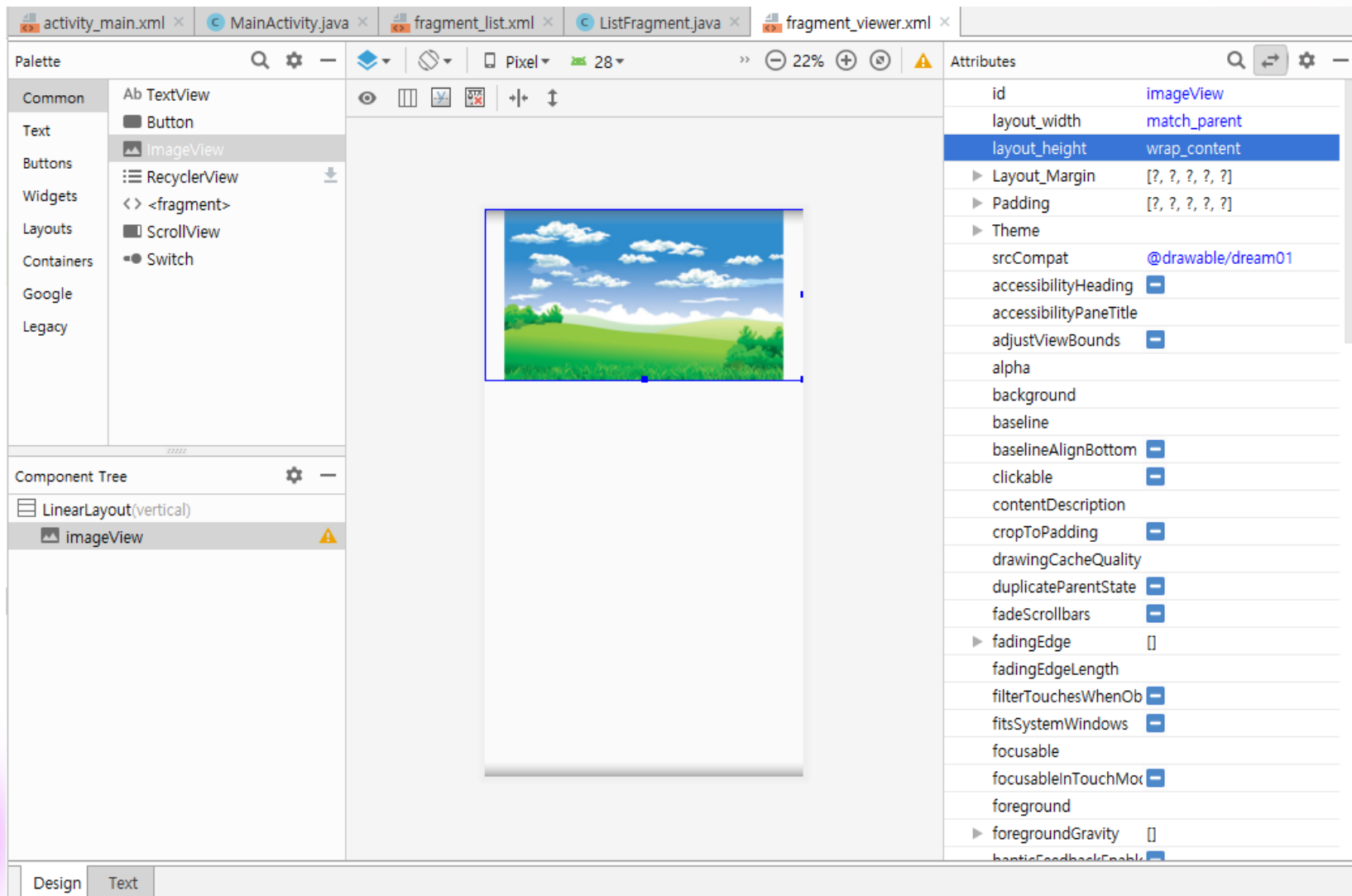
```
public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,
                        @Nullable Bundle savedInstanceState) {
    ViewGroup rootView = (ViewGroup) inflater.inflate(R.layout.fragment_list, container, false);
    Button button = rootView.findViewById(R.id.button);
    button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (callback != null) {
                callback.onImageSelected(0);
            }
        }
    });

    Button button2 = rootView.findViewById(R.id.button2);
    button2.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (callback != null) {
                callback.onImageSelected(1);
            }
        }
    });
}
```



# 두 개의 프래그먼트로 구성된 이미지 뷰어

- 두번째 프래그먼트에 이미지뷰 추가





## 두 개의 프래그먼트로 구성된 이미지 뷰어

- 두번째 프래그먼트에 이미지뷰 추가

참조파일 SampleFragment2>/app/java/org.techtown.fragment/ViewerFragment.java

```
public class ViewerFragment extends Fragment {  
    ImageView imageView;  
  
    @Nullable  
    @Override  
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container,  
                             @Nullable Bundle savedInstanceState) {  
        ViewGroup rootView = (ViewGroup) inflater.inflate(R.layout.fragment_viewer,  
                                                            container, false);  
  
        imageView = rootView.findViewById(R.id.imageView);  
        return rootView;  
    }  
  
    public void setImage(int resId) {  
        imageView.setImageResource(resId);  
    }  
}
```



## 두 개의 프래그먼트로 구성된 이미지 뷰어

- 메인 액티비티의 XML 레이아웃에 두 개의 프래그먼트 추가

```
<fragment
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:name="org.techtown.fragment.ListFragment"
    android:id="@+id/listFragment"
/>

<fragment
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:name="org.techtown.fragment.ViewerFragment"
    android:id="@+id/viewerFragment"
/>
```



# 두 개의 프래그먼트로 구성된 이미지 뷰어

- 메인 액티비티 코드에서 버튼 클릭 시 이미지 설정

참조파일 SampleFragment2>/app/java/org.techtown.fragment/MainActivity.java

```
public class MainActivity extends AppCompatActivity
    implements ListFragment.ImageSelectionCallback {

    ListFragment listFragment;
    ViewerFragment viewerFragment;

    int[] images = {R.drawable.dream01, R.drawable.dream02, R.drawable.dream03};

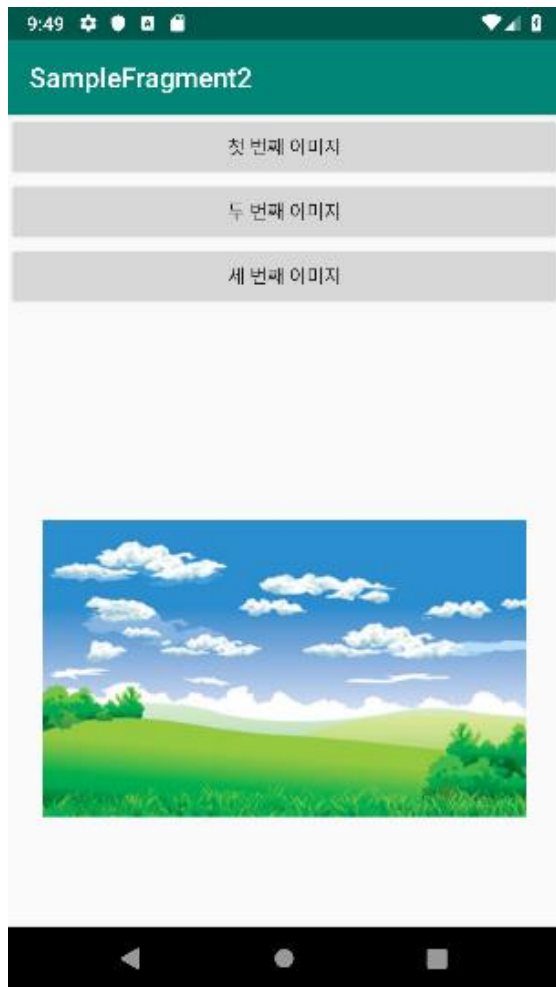
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        FragmentManager manager = getSupportFragmentManager();
        listFragment = (ListFragment) manager.findFragmentById(R.id.listFragment);
        viewerFragment = (ViewerFragment) manager.findFragmentById(R.id.viewerFragment);
    }

    @Override
    public void onImageSelected(int position) {
        viewerFragment.setImage(images[position]);
    }
}
```



## 앱 실행 결과



3.

액션바 사용하기



# 안드로이드의 메뉴

- [메뉴] 버튼은 안드로이드가 아이폰과 다른 특징을 보여주는 것 중의 하나
- [메뉴] 버튼을 누르면 숨어있던 메뉴가 보이게 되는데 이 메뉴를 애플리케이션에서 구현할 때는 옵션 메뉴(Option Menu)라고 부름
- 옵션 메뉴를 비롯해 안드로이드에서 제공하는 메뉴는 크게 두 가지임

## 옵션 메뉴

- 하드웨어 [메뉴] 버튼을 눌렀을 때 나타나는 메뉴로 각각의 화면마다 설정된 주요 메뉴임
- 옵션 메뉴는 최대 6개까지의 메뉴 아이템을 포함할 수 있으며,  
그 이상의 메뉴 아이템을 추가하면 "More"라는 메뉴 아이템으로 표시됨

## 컨텍스트 메뉴

- 화면을 길게 누르면 나타나는 메뉴로 텍스트뷰의 편집 상태를 바꾸거나 할 때 사용하는 메뉴임
- 뷰에 설정하여 나타나게 할 수 있음





## 메뉴 사용 방식의 기본

- onCreateOptionsMenu()와 onCreateContextMenu() 메소드 재정의
- 메뉴 선택 시의 이벤트 처리
  - ✓ 컨텍스트 메뉴의 아이템 선택 : onOptionsItemSelected() 호출
  - ✓ onOptionsItemSelected() 메소드와 onOptionsItemSelected() 메소드는 모두 applyMenuChoice() 메소드로 위임되어 상위 클래스로 전달됨

```
@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
    return (applyMenuChoice(item) || super.onOptionsItemSelected(item));  
}
```

```
@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
    return (applyMenuChoice(item) || super.onOptionsItemSelected(item));  
}
```

### [Reference]

```
void Activity.registerForContextMenu (View view)
```



# 메뉴를 XML에 정의

New Resource File

File name: menu\_main.xml

Source set: main

Directory name: menu

Available qualifiers:

Country Code

Network Code

Locale

Layout Direction

Smallest Screen Width

Screen Width

Screen Height

Size

Ratio

Orientation

UI Mode

Night Mode

Density

>>

<<

Chosen qualifiers:

Nothing to show

OK

Cancel

Help



# 메뉴를 XML에 정의

참조파일 SampleOptionsMenu>/app/res/menu/menu\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">

    <item android:id="@+id/menu_refresh"
          android:title="새로고침"
          android:icon="@drawable/menu_refresh"
          app:showAsAction="always"
        />

    <item android:id="@+id/menu_search"
          android:title="검색"
          android:icon="@drawable/menu_search"
          app:showAsAction="always"
        />

    <item android:id="@+id/menu_settings"
          android:title="설정"
          android:icon="@drawable/menu_settings"
          app:showAsAction="always"
        />

</menu>
```



## 메뉴에 기능 부여

@Override

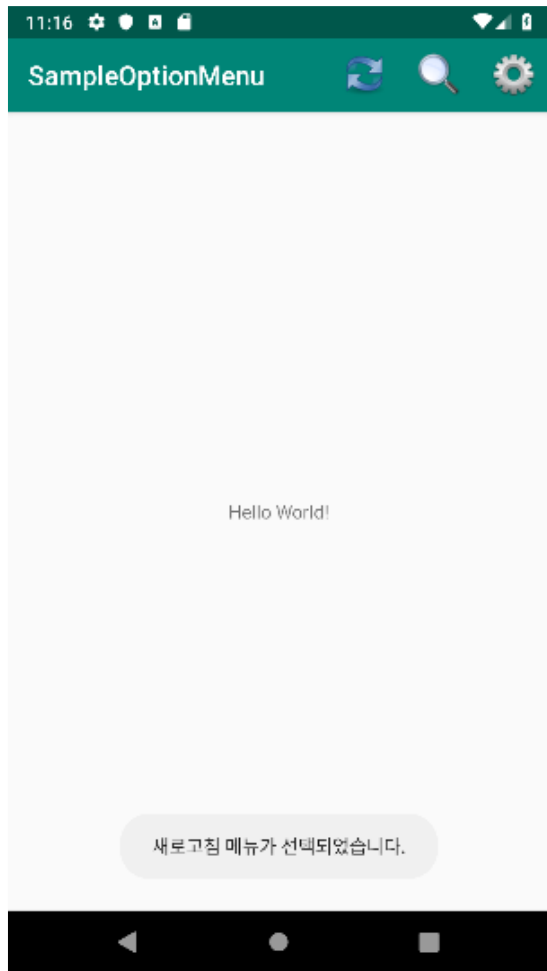
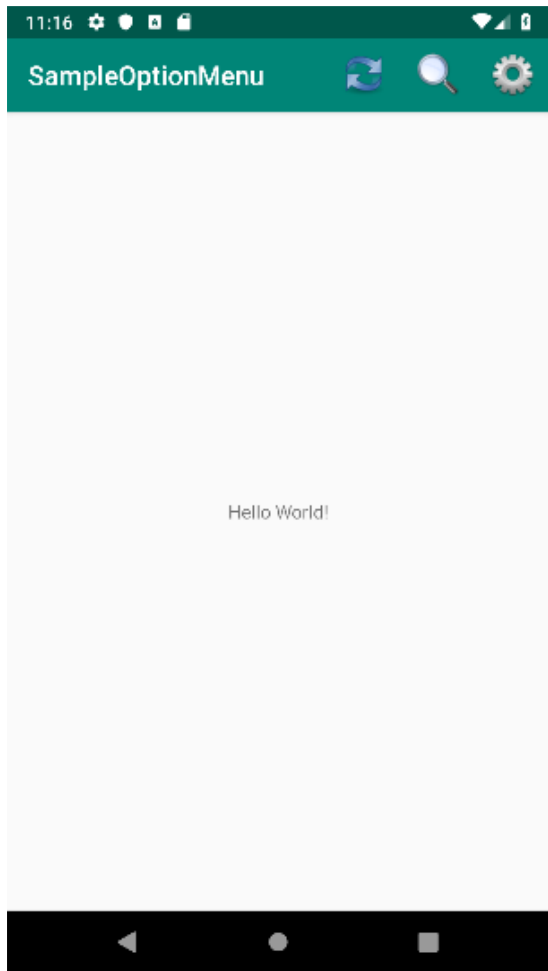
```
public boolean onCreateOptionsMenu(Menu menu) {  
    getMenuInflater().inflate(R.menu.menu_main, menu);  
    return true;  
}
```

@Override

```
public boolean onOptionsItemSelected(MenuItem item) {  
    int curId = item.getItemId();  
    switch(curId) {  
        case R.id.menu_refresh:  
            Toast.makeText(this, "새로고침 메뉴가 선택되었습니다.", Toast.LENGTH_SHORT).show();  
            break;  
        case R.id.menu_search:  
            Toast.makeText(this, "검색 메뉴가 선택되었습니다.", Toast.LENGTH_SHORT).show();  
            break;
```



# 메뉴에 기능 부여





## 액션바에 입력상자 넣기

참조파일 SampleActionBar2>/app/res/layout/search\_layout.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal" >
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="검색 :"
    android:textSize="16sp"
    android:textColor="#ffad8745" />
```

① 문자열 표시를 위한 텍스트뷰

```
<EditText
    android:id="@+id/editText"
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="4dp"
    android:inputType="text"
    android:imeActionId="1337"
    android:imeOptions="actionDone"
    />
```

② 검색어를 입력하기 위한 입력상자

```
</LinearLayout>
```



# 액션바에 입력상자 넣기

참조파일 SampleActionBar2>/app/res/menu/menu\_main.xml

중략...

```

<item android:id="@+id/menu_search"
      android:title="검색"
      android:orderInCategory="102"
      app:showAsAction="always|withText"
      app:actionLayout="@layout/search_layout" />

```

→ 메뉴가 화면에 보이는 방식을 정의한 레이아웃 설정

참조파일 SampleActionBar2>/app/java/org.techtown.actionbar/MainActivity.java

중략...

```

public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_main, menu);
    View v = menu.findItem(R.id.menu_search).getActionView();
    if (v != null) {
        EditText editText = v.findViewById(R.id.editText);
        if (editText != null) {
            editText.setOnEditorActionListener(new TextView.OnEditorActionListener() {
                @Override
                public boolean onEditorAction(TextView v, int actionId, KeyEvent event) {
                    Toast.makeText(getApplicationContext(), "입력됨.", Toast.LENGTH_LONG).show();
                    return true;
                }
            });
        }
    }
}

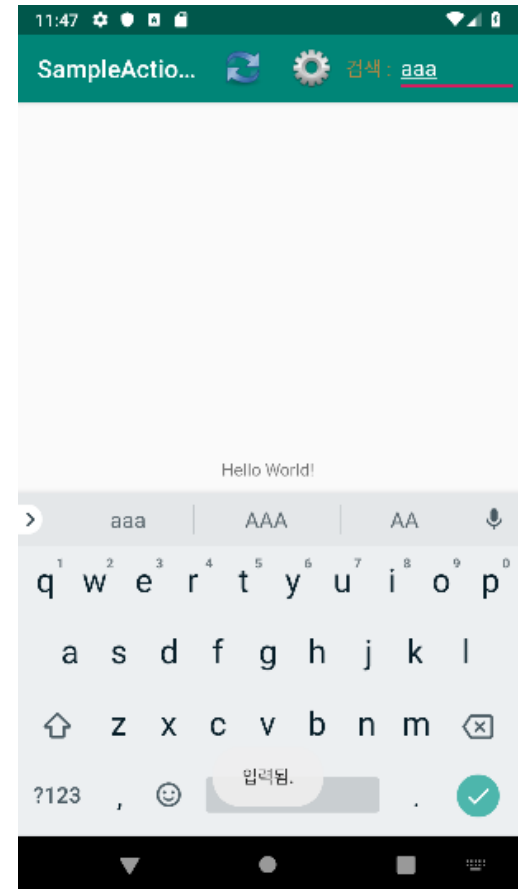
```

① XML로 정의된 메뉴 정보를 인플레이션하여 메모리에 로딩

→ ② 메뉴 아이템 중에서 검색을 위해 정의한 아이템을 뷰 객체로 참조

③ 검색을 위한 메뉴 아이템 안에 정의한 입력상자 객체 참조

→ ④ 입력상자 객체에 리스너 설정



4.

상단 탭과 하단 탭 만들기





## 상단 탭 보여주기

- 외부 라이브러리인 design 라이브러리 추가

참조파일 SampleTab>Gradle Scripts/build.gradle (Module: app)

중략...

```
dependencies {
```

중략...

```
    implementation 'com.android.support:design:28.0.0'
```

```
}
```



# 메인 액티비티의 XML 레이아웃 작성

참조파일 SampleTab>/app/res/layout/activity\_main.xml

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

    <android.support.design.widget.CoordinatorLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <android.support.design.widget.AppBarLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">

            <android.support.v7.widget.Toolbar
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:background="@color/colorPrimaryDark"
                android:theme="@style/ThemeOverlay.AppCompat.Dark"
                android:elevation="1dp"
                android:id="@+id/toolbar"
            >

                <TextView
                    android:id="@+id/titleText"
                    android:layout_width="wrap_content"
```



# 메인 액티비티의 XML 레이아웃 작성

```
        android:layout_height="wrap_content"
        android:textAppearance=
            "@style/Base.TextAppearance.Widget.AppCompat.Toolbar.Title"
        android:text="타이틀"
    />

</android.support.v7.widget.Toolbar>

<android.support.design.widget.TabLayout
    android:id="@+id/tabs"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:tabMode="fixed"
    app:tabGravity="fill"
    app:tabTextColor="@color/colorPrimary"
    app:tabSelectedTextColor="@color/colorAccent"
    android:elevation="1dp"
    android:background="@android:color/background_light" />

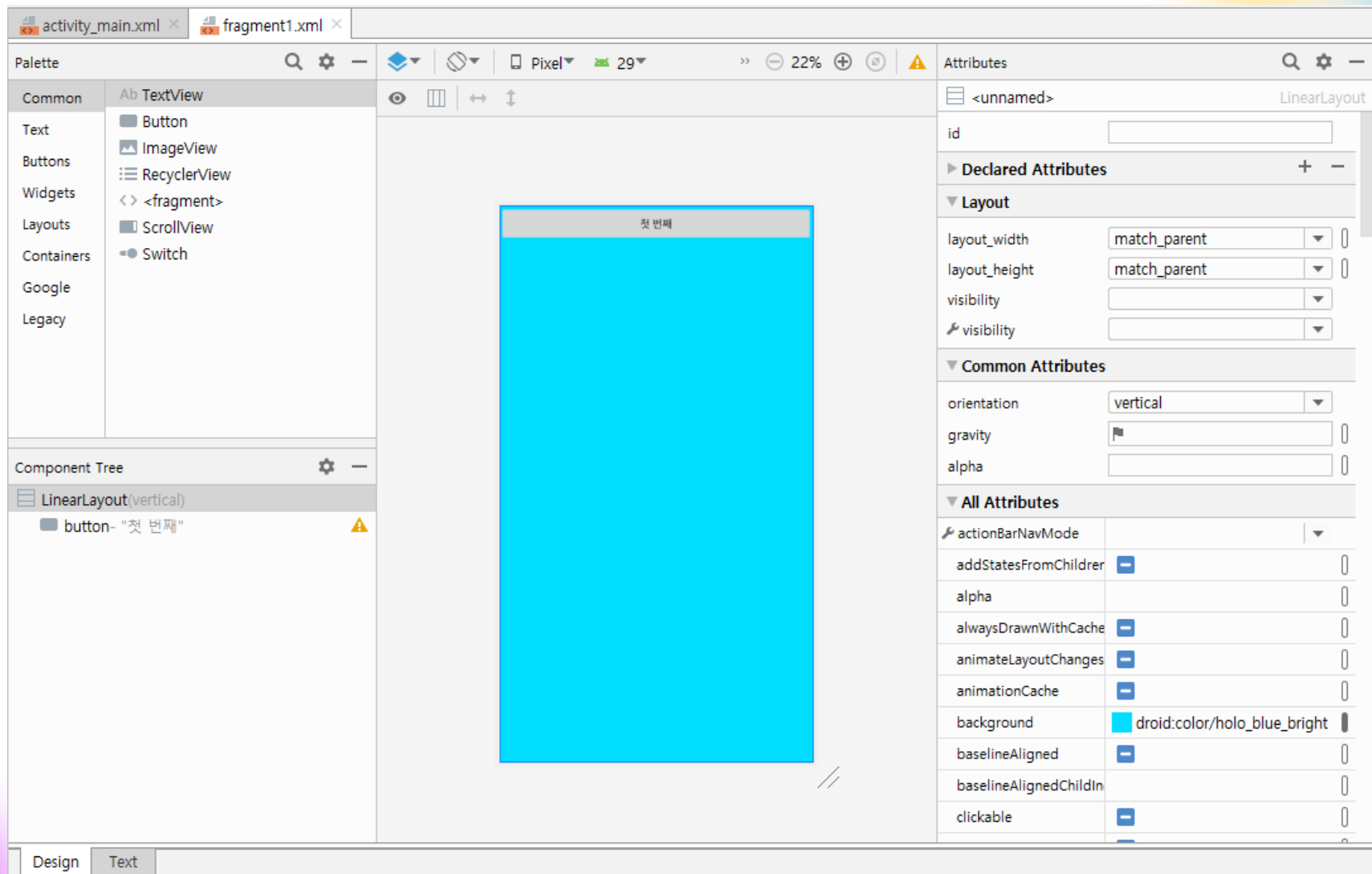
</android.support.design.widget.AppBarLayout>

<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    android:id="@+id/container">

    </FrameLayout>
</android.support.design.widget.CoordinatorLayout>
</RelativeLayout>
```



# 각 탭을 위한 프래그먼트 만들기





## 프래그먼트의 소스 코드

참조파일 SampleTab>/app/java/org.techtown.tab/Fragment1.java

```
public class Fragment1 extends Fragment {  
  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
        return inflater.inflate(R.layout.fragment1, container, false);  
    }  
}
```



## 탭 버튼 추가하고 동작 부여

참조파일 SampleTab>/app/java/org.techtown.tab/MainActivity.java

중략...

```
fragment1 = new Fragment1();
fragment2 = new Fragment2();
fragment3 = new Fragment3();

getSupportFragmentManager().beginTransaction().replace(R.id.container, fragment1).commit();

TabLayout tabs = findViewById(R.id.tabs);
tabs.addTab(tabs.newTab().setText("통화기록"));
tabs.addTab(tabs.newTab().setText("스팸기록"));
tabs.addTab(tabs.newTab().setText("연락처"));

tabs.addTabSelectedListener(new TabLayout.OnTabSelectedListener() {
    @Override
    public void onTabSelected(TabLayout.Tab tab) {
        int position = tab.getPosition();
        Log.d("MainActivity", "선택된 탭 : " + position);
    }
});
```



# 앱 실행 결과





# 하단 탭 보여주기

- 하단 탭 구성을 위한 XML 생성

참조파일 SampleTab2>/app/res/menu/menu\_bottom.xml

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

    <item
        android:id="@+id/tab1"
        app:showAsAction="ifRoom"
        android:enabled="true"
        android:icon="@android:drawable/ic_dialog_email"
        android:title="이메일" />

    <item
        android:id="@+id/tab2"
        app:showAsAction="ifRoom"
        android:icon="@android:drawable/ic_dialog_info"
        android:title="정보" />

    <item
        android:id="@+id/tab3"
        app:showAsAction="ifRoom"
        android:enabled="true"
        android:icon="@android:drawable/ic_dialog_map"
        android:title="위치" />

</menu>
```





## 하단 탭 보여주기

- BottomNavigationView 위젯으로 구성

참조파일 SampleTab2>/app/res/layout/activity\_main.xml

```
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <FrameLayout
        android:id="@+id/container"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_behavior="@string/appbar_scrolling_view_behavior" />

    <android.support.design.widget.BottomNavigationView
        android:id="@+id/bottom_navigation"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginEnd="@dp"
        android:layout_marginStart="@dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:itemBackground="@color/colorPrimary"
        app:itemIconTint="@drawable/item_color"
        app:itemTextColor="@drawable/item_color"
        app:menu="@menu/menu_bottom" />

</android.support.constraint.ConstraintLayout>
```



# 하단 탭 보여주기

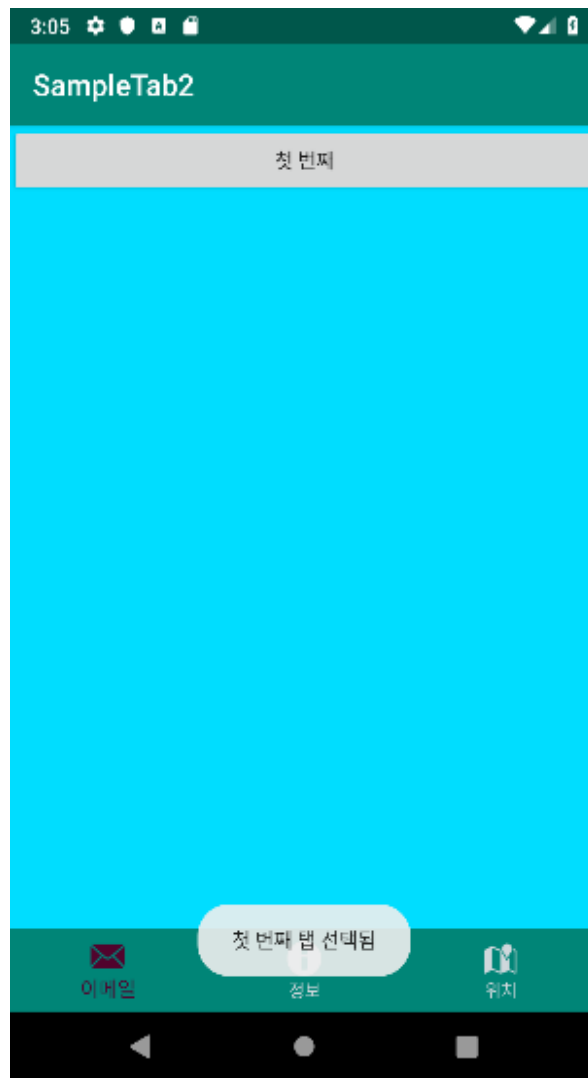
- 코드 구성

```
BottomNavigationView bottomNavigation = findViewById(R.id.bottom_navigation);
bottomNavigation.setOnNavigationItemSelectedListener(
    new BottomNavigationView.OnNavigationItemSelectedListener() {
        @Override
        public boolean onNavigationItemSelected(@NonNull MenuItem item) {
            switch (item.getItemId()) {
                case R.id.tab1:
                    Toast.makeText(getApplicationContext(), "첫 번째 탭 선택됨", Toast.LENGTH_LONG).show();
                    getSupportFragmentManager().beginTransaction().replace(R.id.container, fragment1).commit();

                    return true;
                case R.id.tab2:
                    Toast.makeText(getApplicationContext(), "두 번째 탭 선택됨", Toast.LENGTH_LONG).show();
                    getSupportFragmentManager().beginTransaction().replace(R.id.container, fragment2).commit();
            }
        }
    }
);
```



# 하단 탭 보여주기



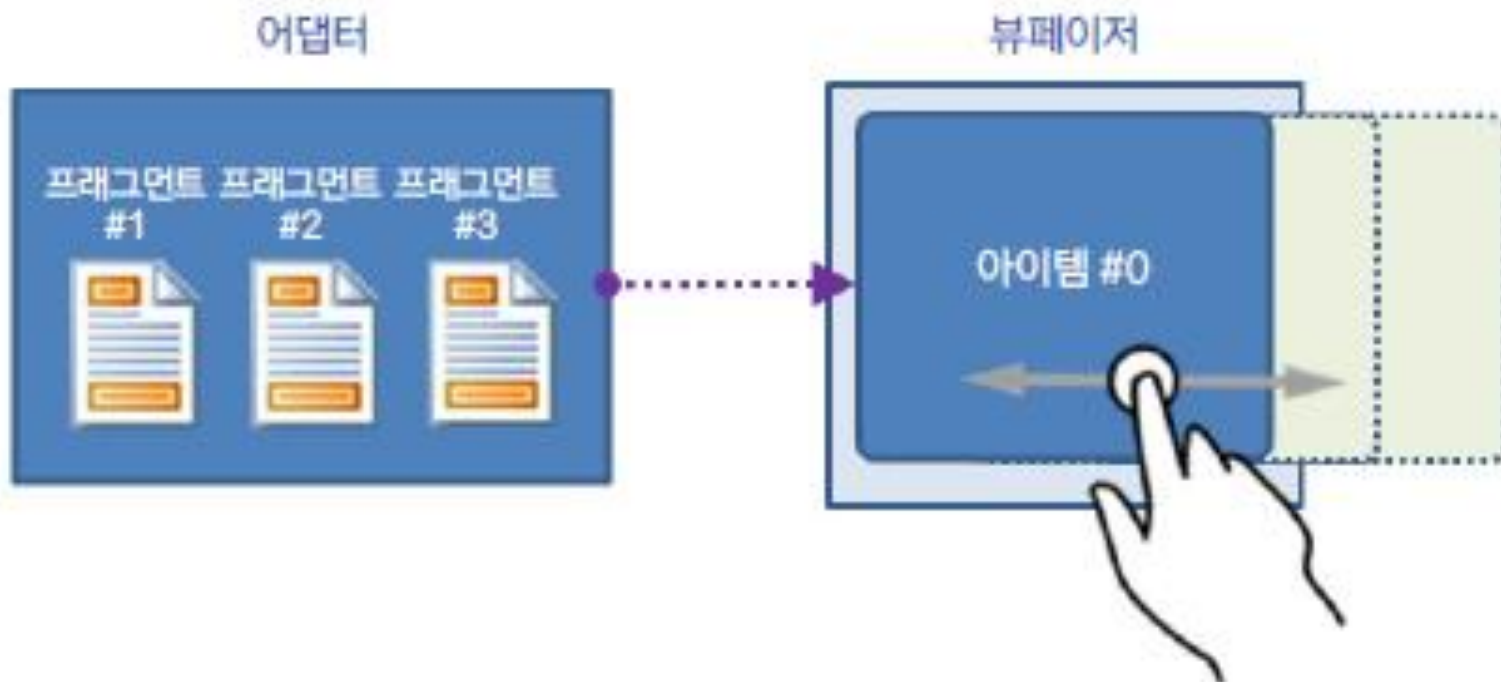
5.

뷰페이지 만들기



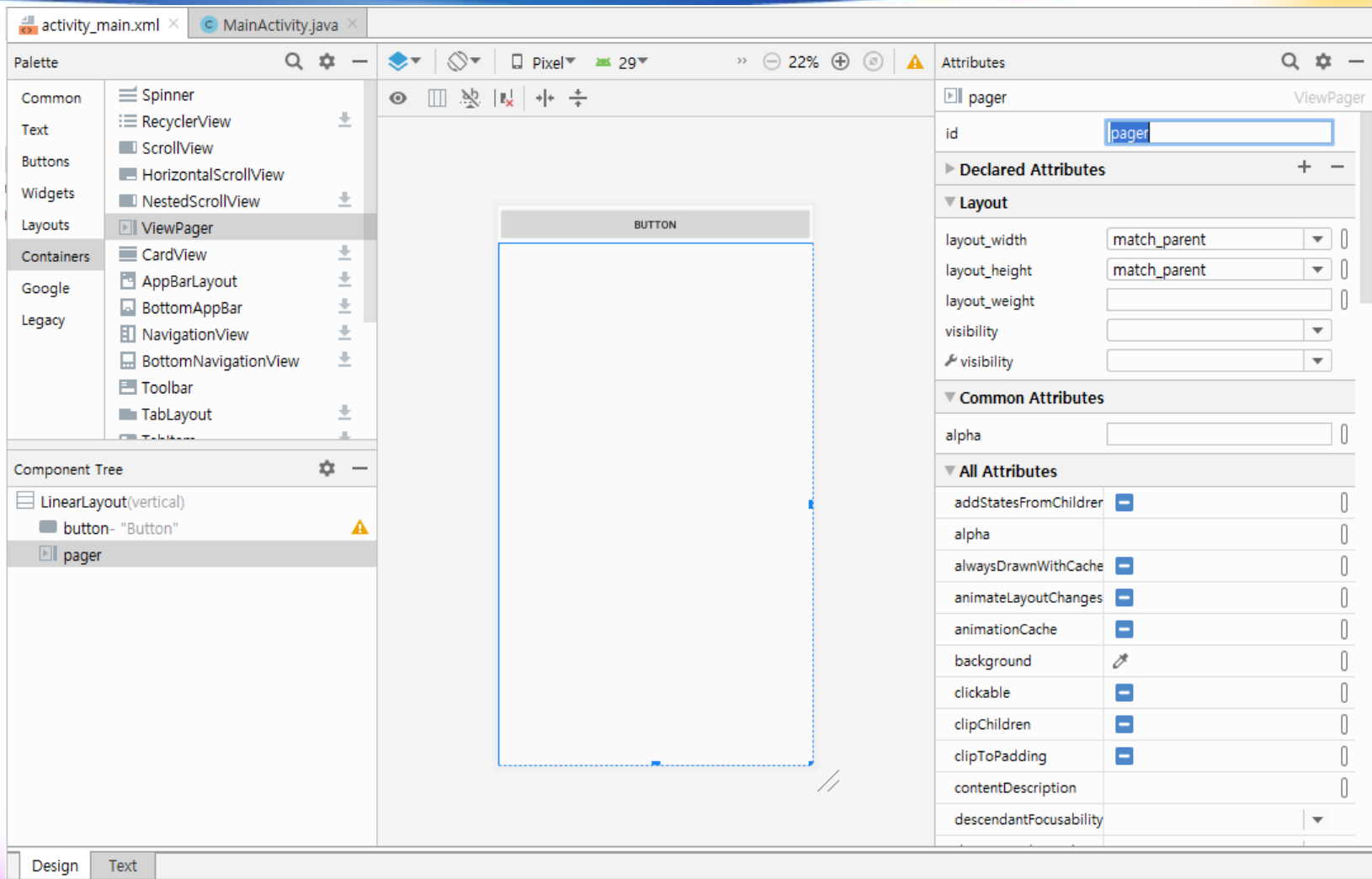
# 뷰페이저

- 손가락으로 좌우 스크롤하여 프래그먼트 화면 넘김





# 레이아웃에 뷰페이지 위젯 추가





## 뷰페이저를 위한 어댑터 생성

참조파일 SamplePager>/app/java/org.techtown.pager/MainActivity.java

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    class MyPagerAdapter extends FragmentStatePagerAdapter {
        ArrayList<Fragment> items = new ArrayList<Fragment>();
        public MyPagerAdapter(FragmentManager fm) {
            super(fm);
        }

        public void addItem(Fragment item) {
            items.add(item);
        }
    }
}
```



# 어댑터에 프래그먼트 추가

참조파일 SamplePager>/app/java/org.techtown.pager/MainActivity.java

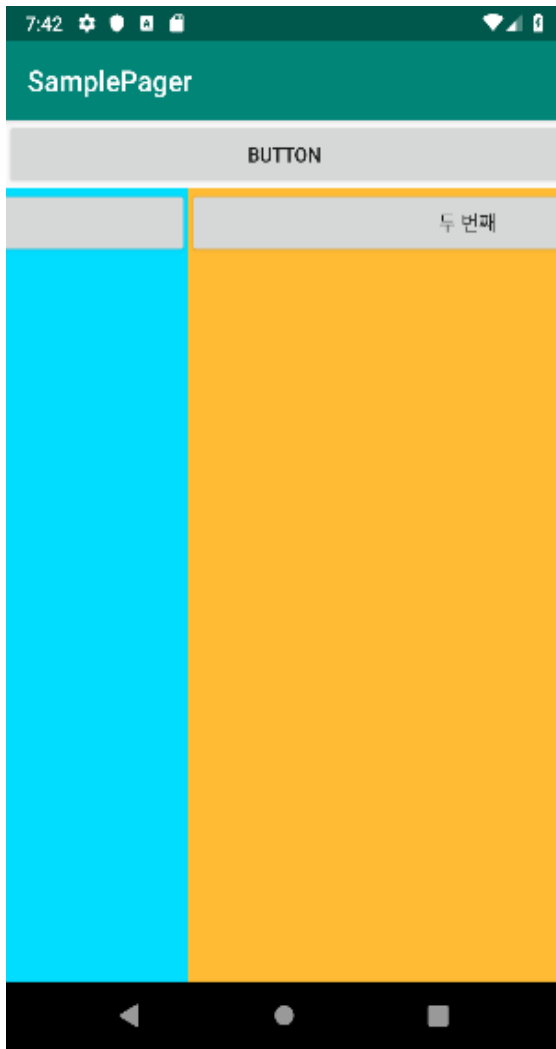
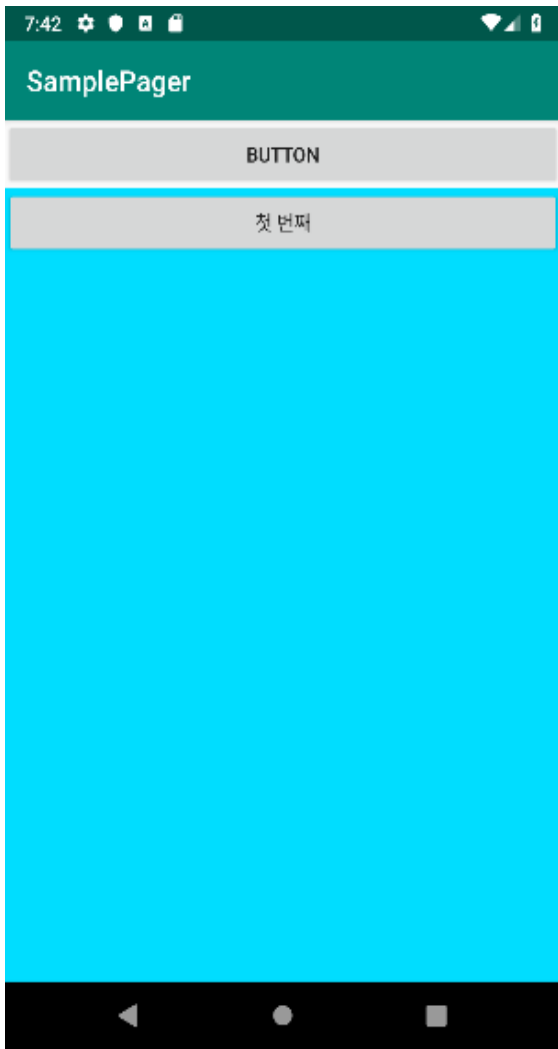
```
public class MainActivity extends AppCompatActivity {  
    ViewPager pager;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        pager = findViewById(R.id.pager);  
        pager.setOffscreenPageLimit(3);  
  
        MyPagerAdapter adapter = new MyPagerAdapter(getSupportFragmentManager());  
  
        Fragment1 fragment1 = new Fragment1();  
        adapter.addItem(fragment1);  
  
        Fragment2 fragment2 = new Fragment2();  
        adapter.addItem(fragment2);  
  
        Fragment3 fragment3 = new Fragment3();  
        adapter.addItem(fragment3);  
  
        pager.setAdapter(adapter);  
    }  
}
```

중략...





# 뷰페이저





## 특정 프래그먼트 화면을 보여주기

참조파일 SamplePager>/app/java/org.techtown.pager/MainActivity.java

중략...

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    Button button = findViewById(R.id.button);
    button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            pager.setCurrentItem(1);
        }
    });

    pager = findViewById(R.id.pager);
    pager.setOffscreenPageLimit(3);
```

중략...



# 타이틀 스트립 보여주기

참조파일 SamplePager>/app/res/layout/activity\_main.xml

중략...

```
android:text="두 번째 화면 보여주기" />
```

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

```
<android.support.v4.view.PagerTitleStrip
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="top"
    android:background="#55cedf"
    android:textColor="#FFFFFF"
    android:paddingTop="5dp"
    android:paddingBottom="5dp">
```

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

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

중략...



# 타이틀 스트립 보여주기

참조파일 SamplePager>/app/java/org.techtown.pager/MainActivity.java

중략...

```
class MyPagerAdapter extends FragmentStatePagerAdapter {
```

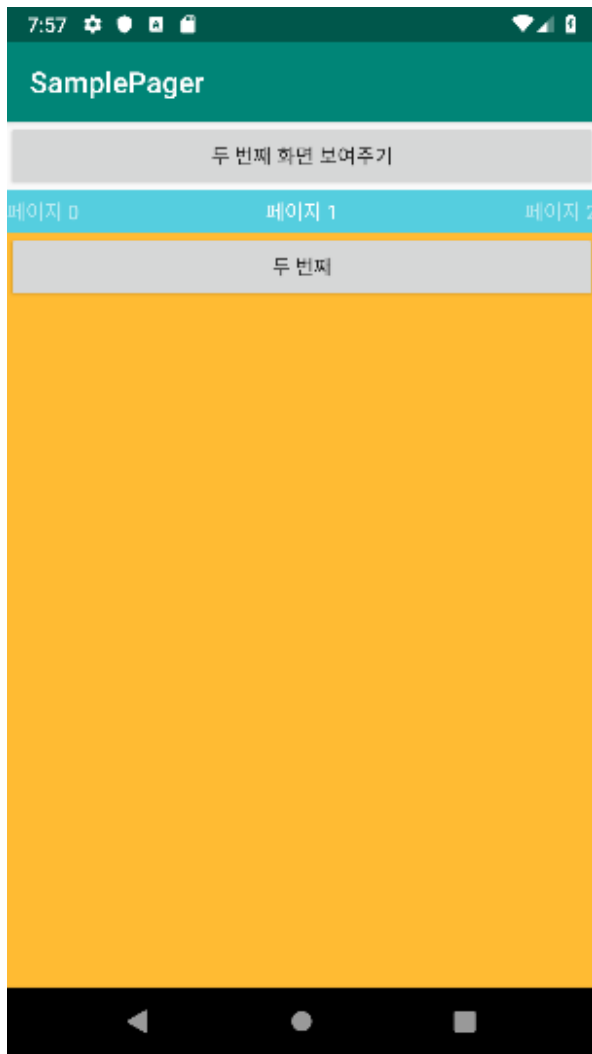
중략...

```
    @Nullable
    @Override
    public CharSequence getPageTitle(int position) {
        return "페이지" + position;
    }
}
```

중략...



# 타이틀 스트립 보여주기



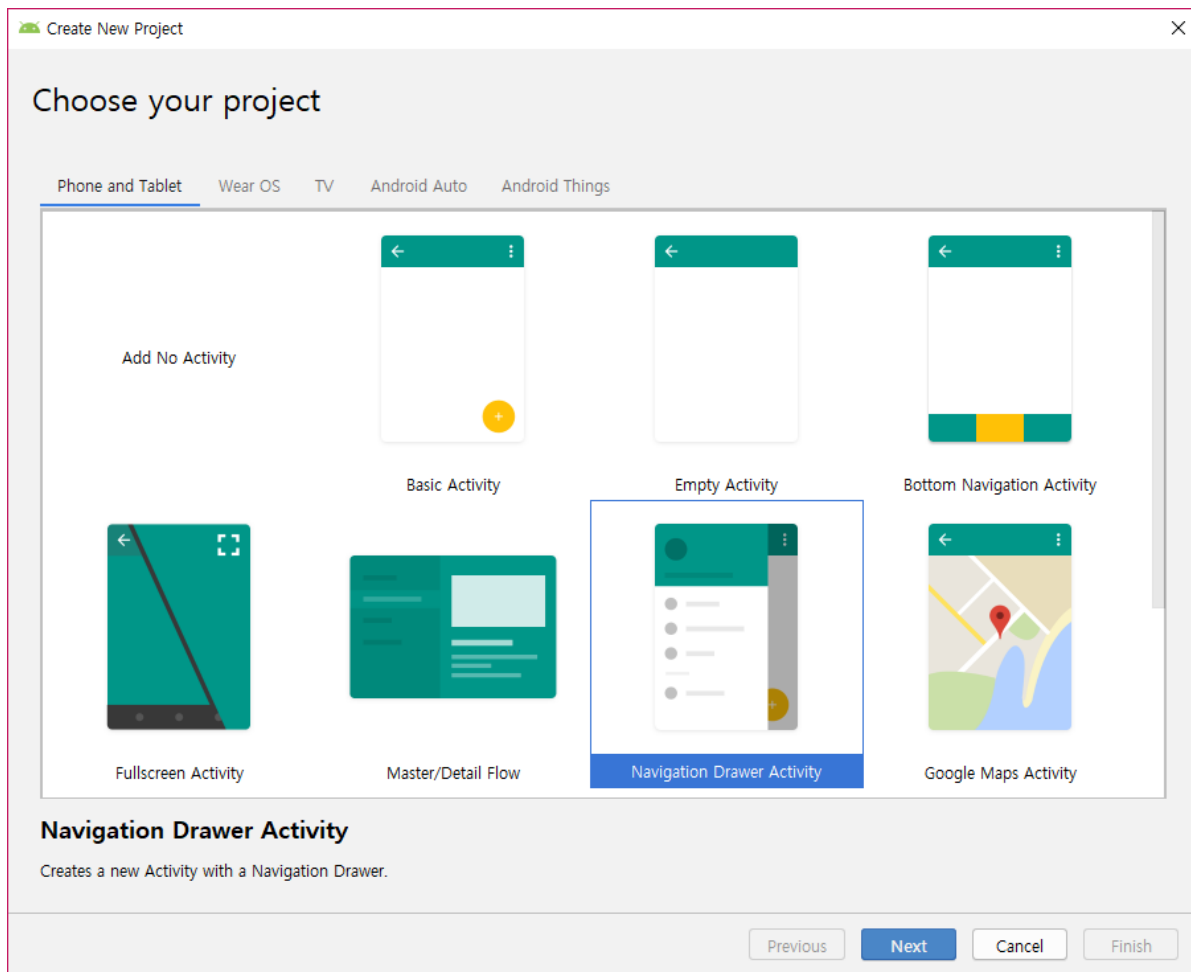
6.

바로가기 메뉴 만들기



# 바로가기 메뉴

- 좌측 상단의 햄버거 모양 아이콘을 눌렀을 때 나타나는 화면으로 기본 화면 유형에 있음





# 바로가기 메뉴

- 액티비티의 스타일이 NoActionBar로 설정되어야 기본 액션바가 없어짐

참조파일 SampleDrawer>/manifests/AndroidManifest.xml

중략...

```
<activity
    android:name=".MainActivity"
    android:label="@string/app_name"
    android:theme="@style/AppTheme.NoActionBar">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

중략...

SampleDrawer라고 회색으로 표시되나요? 안드로이드 스튜디오  
오에서 숨김 또는 대체했기 때문입니다. 회색 영역을 클릭하면 원문  
코드가 나타납니다.

참조파일 SampleDrawer>/app/res/values/styles.xml

중략...

```
<style name="AppTheme.NoActionBar">
    <item name="windowActionBar">false</item>
    <item name="windowNoTitle">true</item>
</style>
```

중략...





# 바로가기 메뉴

- 상단 액션바 설정

참조파일 SampleDrawer>/app/res/layout/activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:openDrawer="start">

    <android.support.design.widget.CoordinatorLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <android.support.design.widget.AppBarLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:theme="@style/AppTheme.AppBarOverlay">

            <android.support.v7.widget.Toolbar
                android:id="@+id/toolbar"
                android:layout_width="match_parent"
                android:layout_height="?attr/actionBarSize"
                android:background="?attr/colorPrimary"
                app:popupTheme="@style/AppTheme.PopupOverlay" />

        </android.support.design.widget.AppBarLayout>
```



## 바로가기 메뉴

- 상단 액션바 설정

```
<FrameLayout
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior">

</FrameLayout>

</android.support.design.widget.CoordinatorLayout>

<android.support.design.widget.NavigationView
    android:id="@+id/nav_view"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_gravity="start"
    android:fitsSystemWindows="true"
    app:headerLayout="@layout/nav_header_main"
    app:menu="@menu/activity_main_drawer" />

</android.support.v4.widget.DrawerLayout>
```



## 바로가기 메뉴

- 코드에서 NavigationDrawer 설정

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    toolbar = findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    DrawerLayout drawer = findViewById(R.id.drawer_layout);
    ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
        this, drawer, toolbar, R.string.navigation_drawer_open, R.string.navigation_drawer_close);
    drawer.addDrawerListener(toggle);
    toggle.syncState();

    NavigationView navigationView = findViewById(R.id.nav_view);
    navigationView.setNavigationItemSelectedListener(this);

    fragment1 = new Fragment1();
    fragment2 = new Fragment2();
    fragment3 = new Fragment3();

    getSupportFragmentManager().beginTransaction().add(R.id.container, fragment1).commit();
}
```



## 바로가기 메뉴

- 바로가기 메뉴의 메뉴아이템 클릭 시 실행될 코드 추가

```
@Override
public boolean onNavigationItemSelected(MenuItem item) {
    int id = item.getItemId();

    if (id == R.id.menu1) {
        Toast.makeText(this, "첫 번째 메뉴 선택됨.", Toast.LENGTH_LONG).show();
        onFragmentSelected(0, null);
    } else if (id == R.id.menu2) {
        Toast.makeText(this, "두 번째 메뉴 선택됨.", Toast.LENGTH_LONG).show();
        onFragmentSelected(1, null);
    } else if (id == R.id.menu3) {
        Toast.makeText(this, "세 번째 메뉴 선택됨.", Toast.LENGTH_LONG).show();
        onFragmentSelected(2, null);
    }

    DrawerLayout drawer = findViewById(R.id.drawer_layout);
    drawer.closeDrawer(GravityCompat.START);
    return true;
}
```



# 바로가기 메뉴

- 바로가기 메뉴의 메뉴 XML 구성

참조파일 SampleDrawer>/app/res/menu/activity\_main\_drawer.xml

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

    <group android:checkableBehavior="single">
        <item
            android:id="@+id/menu1"
            android:icon="@drawable/ic_menu_camera"
            android:title="Import" />
        <item
            android:id="@+id/menu2"
            android:icon="@drawable/ic_menu_gallery"
            android:title="Gallery" />
        <item
            android:id="@+id/menu3"
            android:icon="@drawable/ic_menu_slideshow"
            android:title="Slideshow" />
    </group>

</menu>
```



# 바로가기 메뉴

