



Do It! 안드로이드 앱 프로그래밍

개정6판

안드로이드 스튜디오 3.3 이상 기준

Mar. 2019

저자 : 정재곤



둘째 마당 - Chapter 05

프래그먼트

이지스퍼블리싱(주) 제공 강의 교안

저자 : 정재곤



이번 장에서는 무엇을 다룰까요?



프래그먼트로 부분 화면을 만드는 방법에 대해 알고 싶어요.



- 프래그먼트에 대해 알아보까요?
- 프래그먼트로 구성된 화면을 만들어보까요?
- 액션바와 탭을 사용하는 방법을 알아보까요?
- 바로가기 메뉴를 어떻게 붙이는지 알아보까요?





강의 주제

프래그먼트를 사용하는 방법에 대한 이해



1

프래그먼트란?

2

프래그먼트로 화면 만들기

3

액션바 사용하기

4

상단 탭과 하단 탭 만들기

5

바로가기 메뉴 만들기

1.

프래그먼트란?



- 안드로이드에서 화면을 구성하는 일반적인 방법은 각각의 화면을 액티비티로 만든 후 액티비티 간에 전환시키는 것
- 하나의 화면에서 두 가지 작업을 동시에 하고 싶다면?
- 예) 화면을 반으로 나눈 후 한 쪽에서는 트위터나 페이스북과 같은 SNS 기능을 사용하고 다른 한 쪽에서는 동영상을 보고 싶다면?
- 태블릿 같은 경우 한 쪽에는 메뉴 화면이 보이고, 메뉴 화면에서 하나의 메뉴를 선택하면 다른 한 쪽에 해당 기능을 실행한 화면이 보이도록 하면 자주 화면이 전환되는 불편함이 없어지고 넓은 화면을 잘 활용할 수 있게 됨
- 전체 화면 안에 부분 화면을 만들어 넣으면 화면을 전환하지 않아도 되니 불편하지도 않고 넓은 화면을 잘 활용할 수 있다
- **하나의 화면을 여러 부분으로 나눠서 보여주거나 각각의 부분 화면 단위로 바꿔서 보여주고 싶을 때 사용하는 것이 프래그먼트(Fragment)**
- 프래그먼트는 **하나의 화면 안에 들어가는 부분 화면**과 같아서 하나의 레이아웃처럼 보임
- 프래그먼트를 사용하는 가장 큰 목적은 **분할된 화면들을 독립적으로 구성**하고 그 상태를 관리하는 데 있다.



프래그먼트란?

- **프래그먼트 (Fragment)**

- 화면의 일정 영역을 독립적으로 처리하기 위해 만들어진 특별한 화면 구성 요소
- 태블릿의 대화면에서 화면 분할이 필요하게 되면서 만들어짐

- **프래그먼트의 기본 목적**

- 하나의 화면이 XML 레이아웃과 자바 소스로 구성된다는 점에 착안하여 **하나의 프래그먼트가 XML 레이아웃과 자바 소스로 구성되도록** 하고 **독립적으로 관리되도록** 하기 위함

- **기본 구성 방식**

- 액티비티가 동작하는 방식을 본떠 만들었음

- **사용되는 개념**

- 프래그먼트 매니저 : 프래그먼트를 관리하는 객체
- 트랜잭션 : 프래그먼트의 처리를 위해 만든 단위

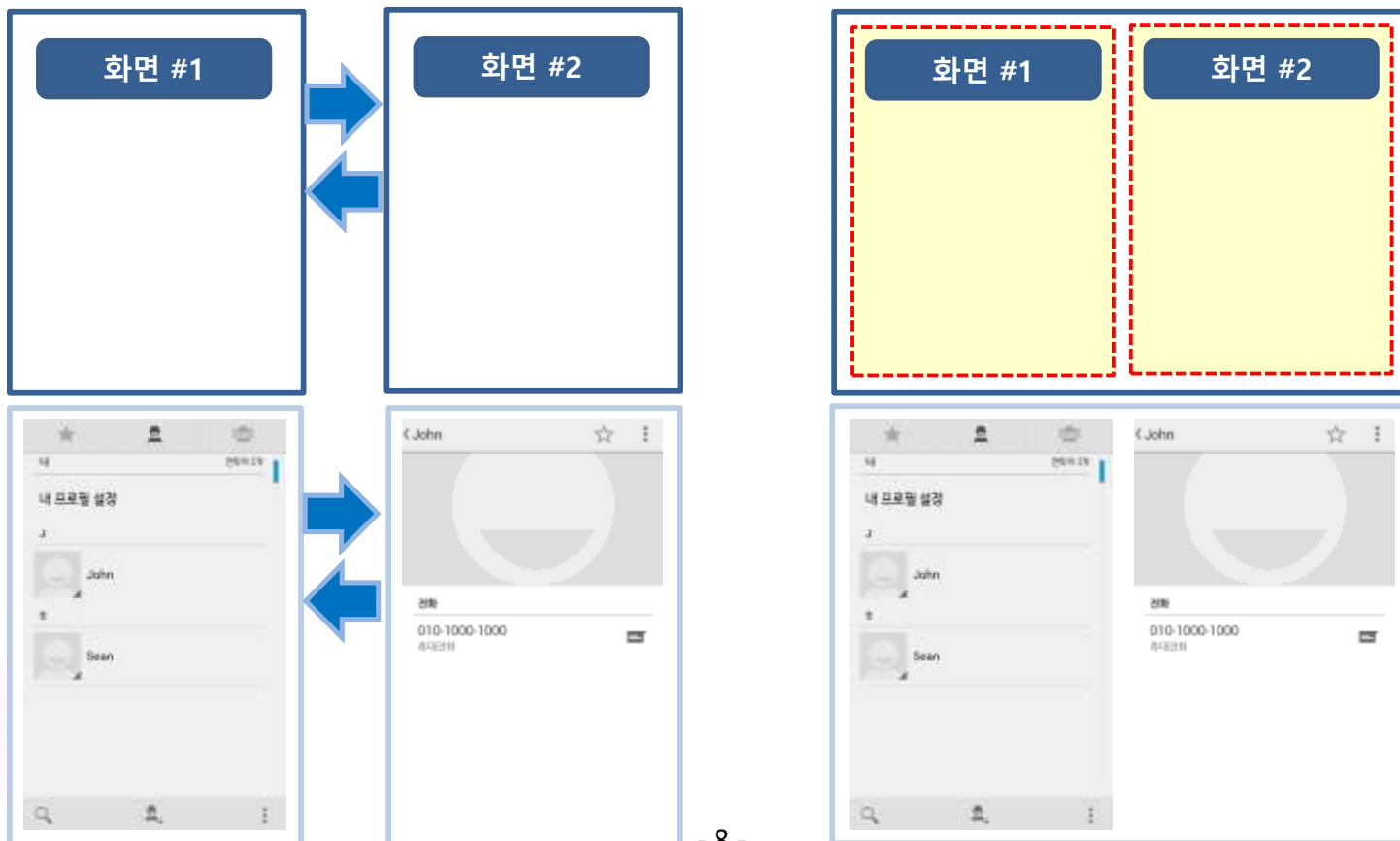


화면을 전환하는 경우와 화면을 분할하는 경우

• 화면 전환과 화면 분할

태블릿과 같이 큰 화면에서는 한 화면에 여러 부분화면을 넣는 것이 효율적임

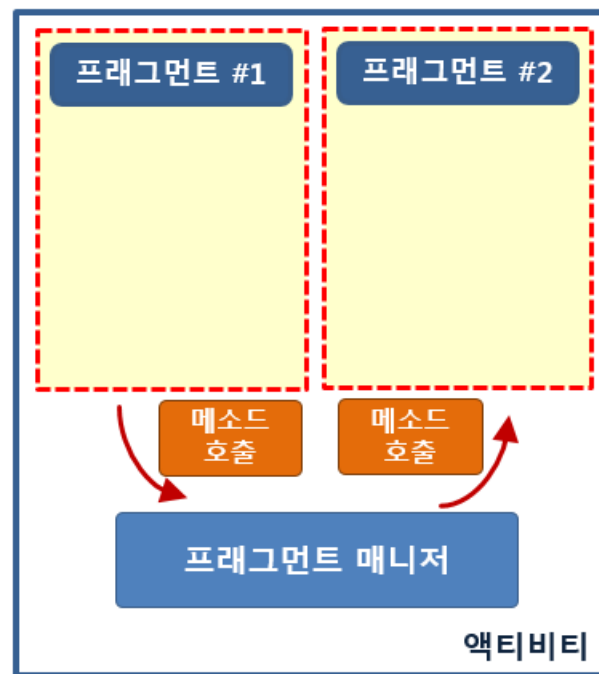
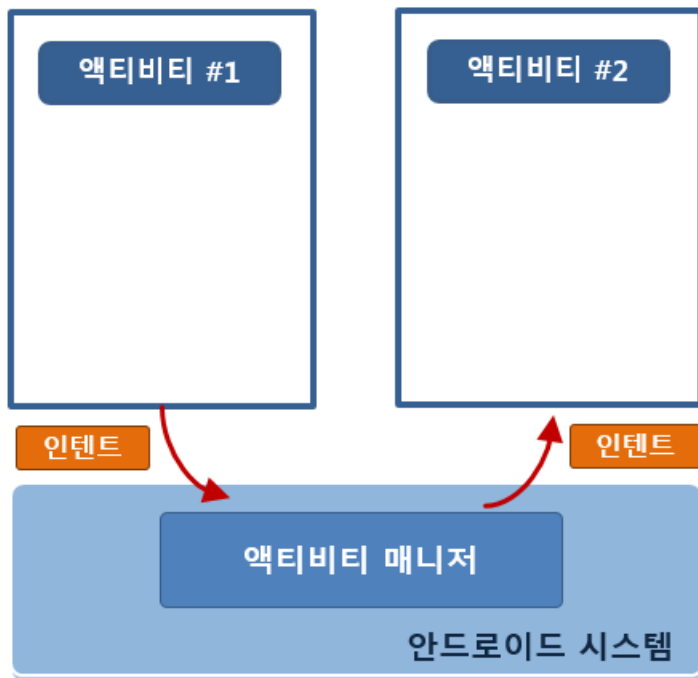
- (1) 두 개의 화면을 액티비티로 만들고 액티비티 간 전환
- (2) 하나의 액티비티 위에 프래그먼트를 두고 프래그먼트 간 전환
- ✓ 프래그먼트로 만들어 두면 스마트폰에서는 프래그먼트 간 화면 전환, 태블릿에서는 두 개의 프래그먼트를 하나의 액티비티 위에서 동시에 보여주는 화면 분할이 가능함





액티비티와 프래그먼트의 동작 방식 비교

- 액티비티를 시스템의 액티비티 매니저에서 관리하듯이 **프래그먼트를 액티비티의 프래그먼트 매니저에서 관리함**
 - 프래그먼트 입장에서는 액티비티가 시스템 역할을 하므로 **프래그먼트는 항상 액티비티 위에 올라가 있어야 함**



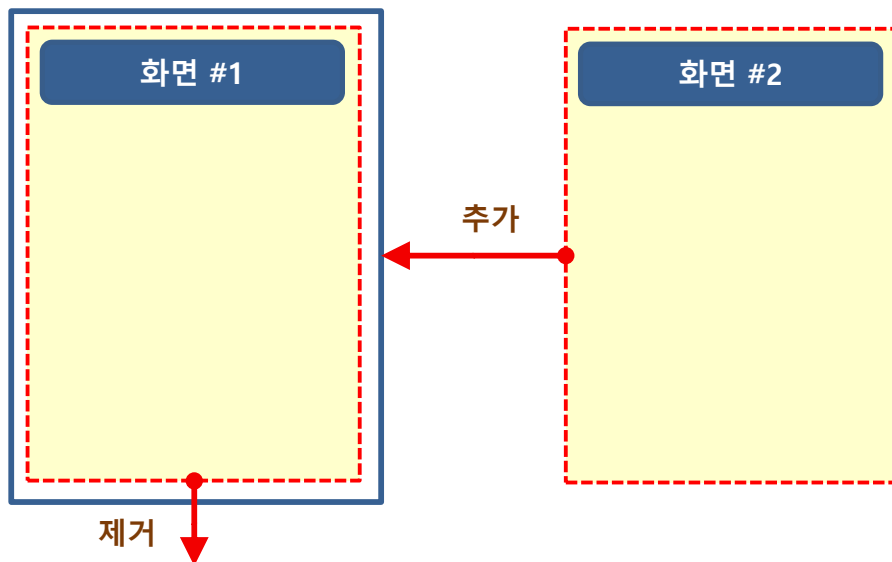


하나의 액티비티 안에서 프래그먼트 전환

• 프래그먼트 전환

- 프래그먼트 매니저와 트랜잭션을 이용해 추가(add)나 교체(replace) 가능
- 싱글 프래그먼트라고 부르며 액티비티 전환 없이 화면 전체가 전환되는 효과를 낼 수 있음

하나의 액티비티에 하나의 프래그먼트를 전체 화면으로 보여주면 시스템과 관계없이 전체 화면을 전환하는 효과를 만들 수 있음

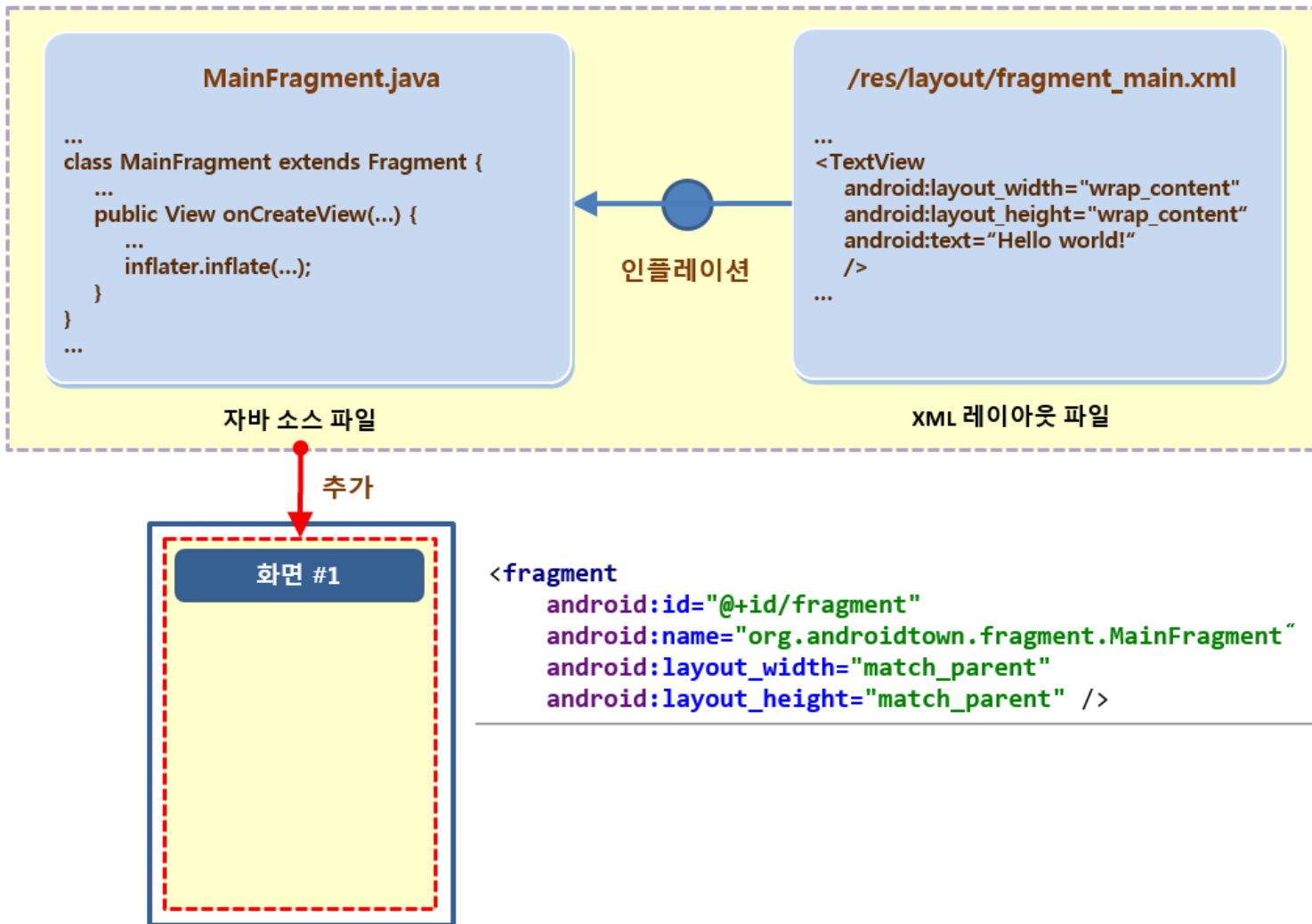




프래그먼트를 위한 레이아웃 인플레이션

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

하나의 프래그먼트





Fragment 클래스

public final [Activity](#) getActivity ()

이 프래그먼트를 포함하는 액티비티를 리턴함.

public final [FragmentManager](#) getFragmentManager ()

이 프래그먼트를 포함하는 액티비티에서 프래그먼트 객체들과 의사소통하는 프래그먼트 매니저를 리턴함.

public final [Fragment](#) getParentFragment ()

이 프래그먼트를 포함하는 부모가 프래그먼트일 경우 리턴함. 액티비티이면 null을 리턴함.

public final int getId ()

이 프래그먼트의 ID를 리턴함.



FragmentManager 클래스

public abstract [FragmentTransaction](#) beginTransaction ()

프래그먼트를 변경하기 위한 트랜잭션을 시작함.

public abstract [Fragment](#) findFragmentById (int id)

ID를 이용해 프래그먼트 객체를 찾음.

public abstract [Fragment](#) findFragmentByTag ([String](#) tag)

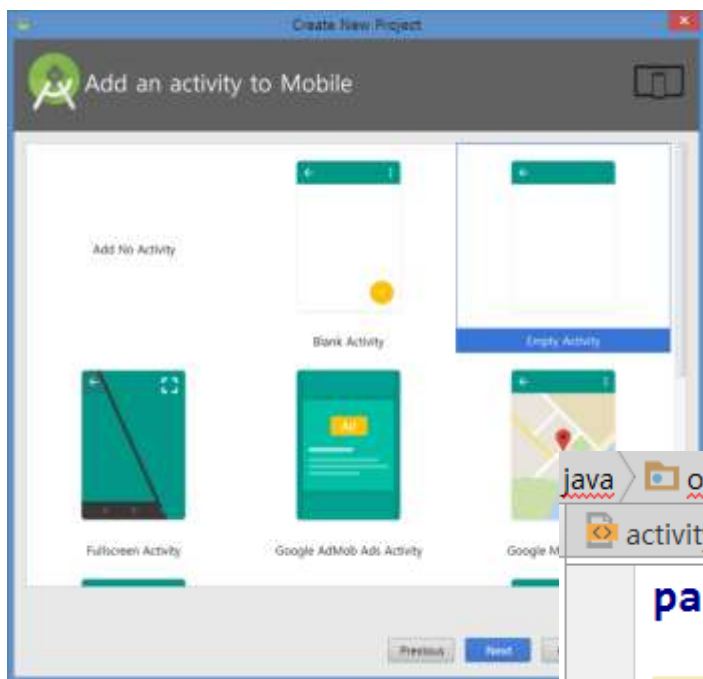
태그 정보를 이용해 프래그먼트 객체를 찾음.

public abstract boolean executePendingTransactions ()

트랜잭션은 commit() 메소드를 호출하면 실행되지만 비동기(asynchronous) 방식으로 실행되므로 즉시 실행하고 싶다면 이 메소드를 추가로 호출해야 함.



프래그먼트 프로젝트 만들기



```
java > org > androidtown > fragment > MainFragment
activity_main.xml x MainActivity.java x fragment_main.xml x MainFragment.java x
package org.androidtown.fragment;

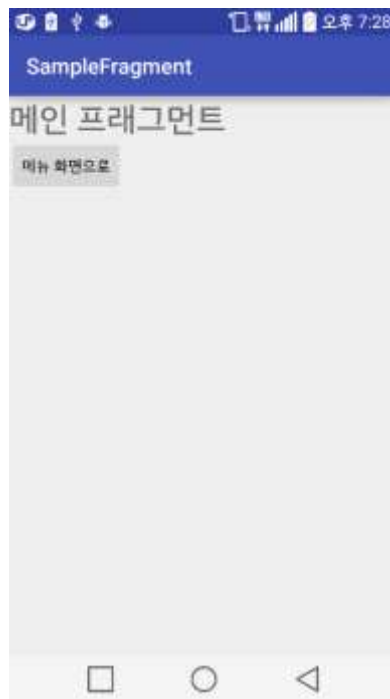
/**
 * Created by user on 2015-12-25
 */
public class MainFragment extends Fragment {
    |
}
```

? android.app.Fragment? (multiple choices...) Alt+Enter



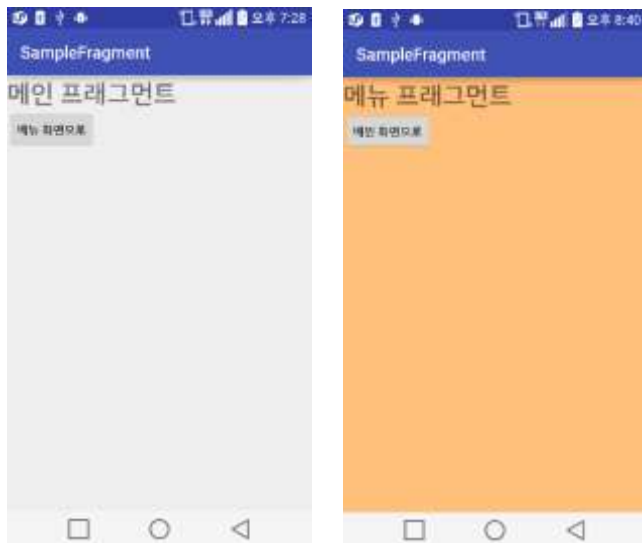
프래그먼트를 만들어 사용하는 과정

- (1) 프래그먼트를 위한 XML 레이아웃 만들기
- (2) 프래그먼트 클래스 만들기 (클래스 정의)
- (3) onCreateView() 메소드 안에서 인플레이션하기
- (4) 메인 액티비티를 위한 XML 레이아웃에 추가하거나 프래그먼트 매니저를 이용해 코드에서 추가하기





프래그먼트의 교체



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

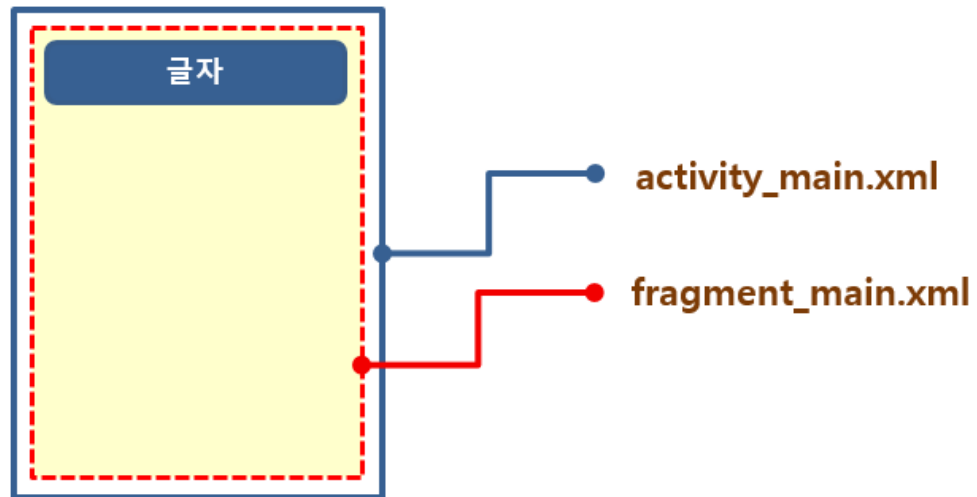
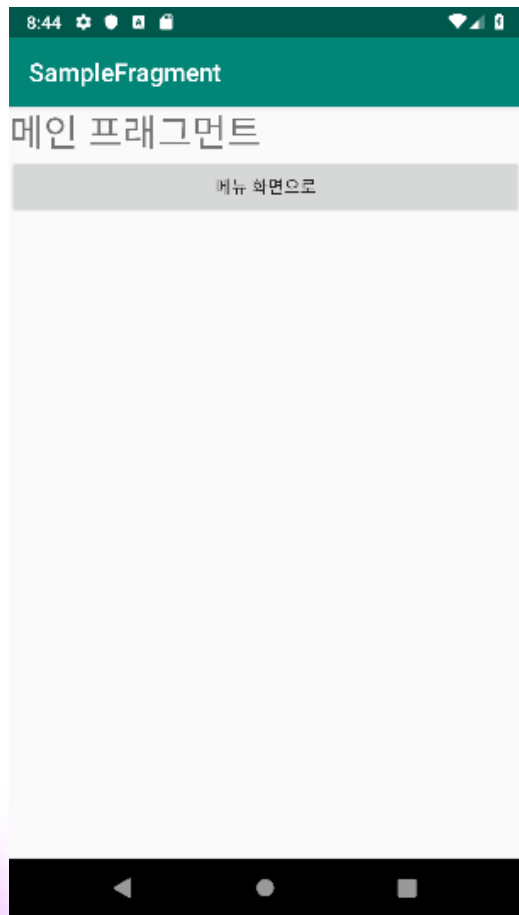
    Button button = (Button) rootView.findViewById(R.id.button);
    button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            MainActivity activity = (MainActivity) getActivity();
            activity.onFragmentChanged(0);
        }
    });

    return rootView;
}
```




앱 실행 결과

- 프래그먼트가 부분화면으로 추가됨

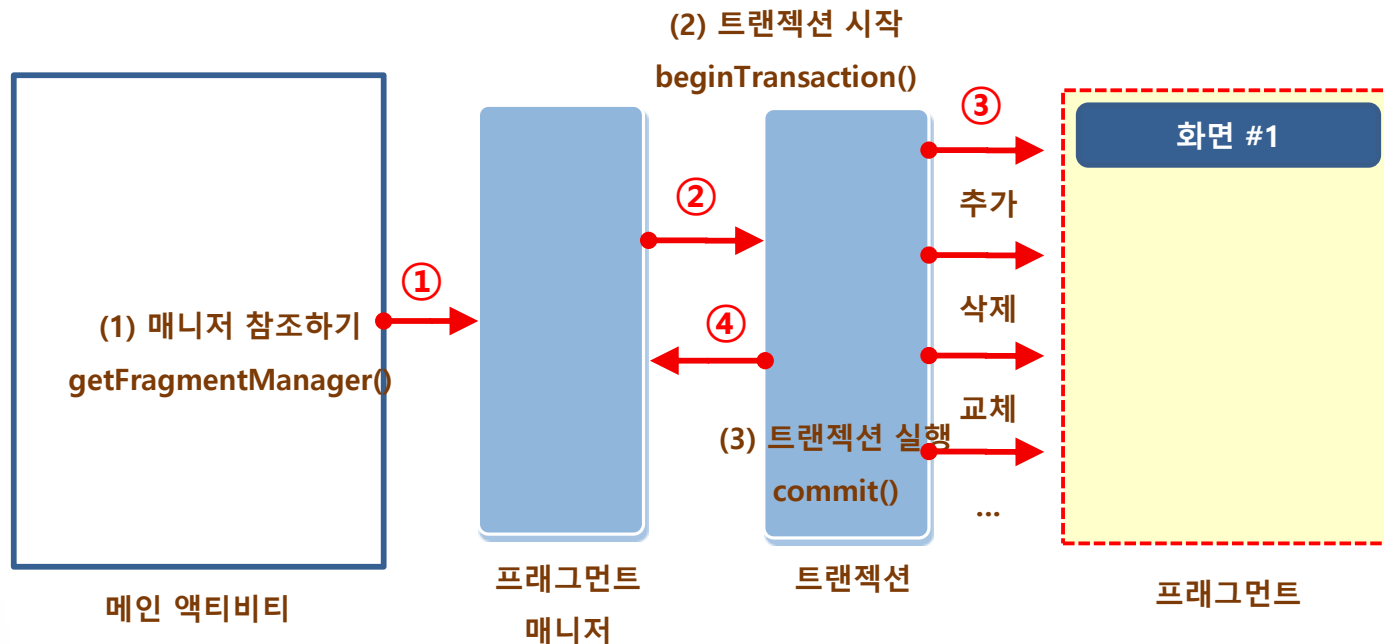




액티비티와 프래그먼트 간 의사소통

• 프래그먼트 처리 순서

- (1) 프래그먼트 매니저 객체 참조
- (2) 트랜잭션 시작
- (3) 프래그먼트의 추가, 삭제 또는 교체
- (4) 트랜잭션 커밋 (commit)





activity_main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <fragment
        android:id="@+id/mainFragment"
        android:name="com.example.fragmentapp.MainFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</RelativeLayout>
```



fragment_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainFragment" >
    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="메인 프래그먼트"
        android:textSize="30sp" />
    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="메뉴 화면으로"/>
</LinearLayout>
```



fragment_menu.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@android:color/holo_orange_light"
    android:orientation="vertical"
    tools:context=".MainFragment">
    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="메뉴 프래그먼트"
        android:textSize="30sp" />
    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="메인 화면으로" />
</LinearLayout>
```

```
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();  
        }  
    }  
}
```



MainFragment

```
public class MainFragment extends Fragment {  
    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;  
    }  
}
```

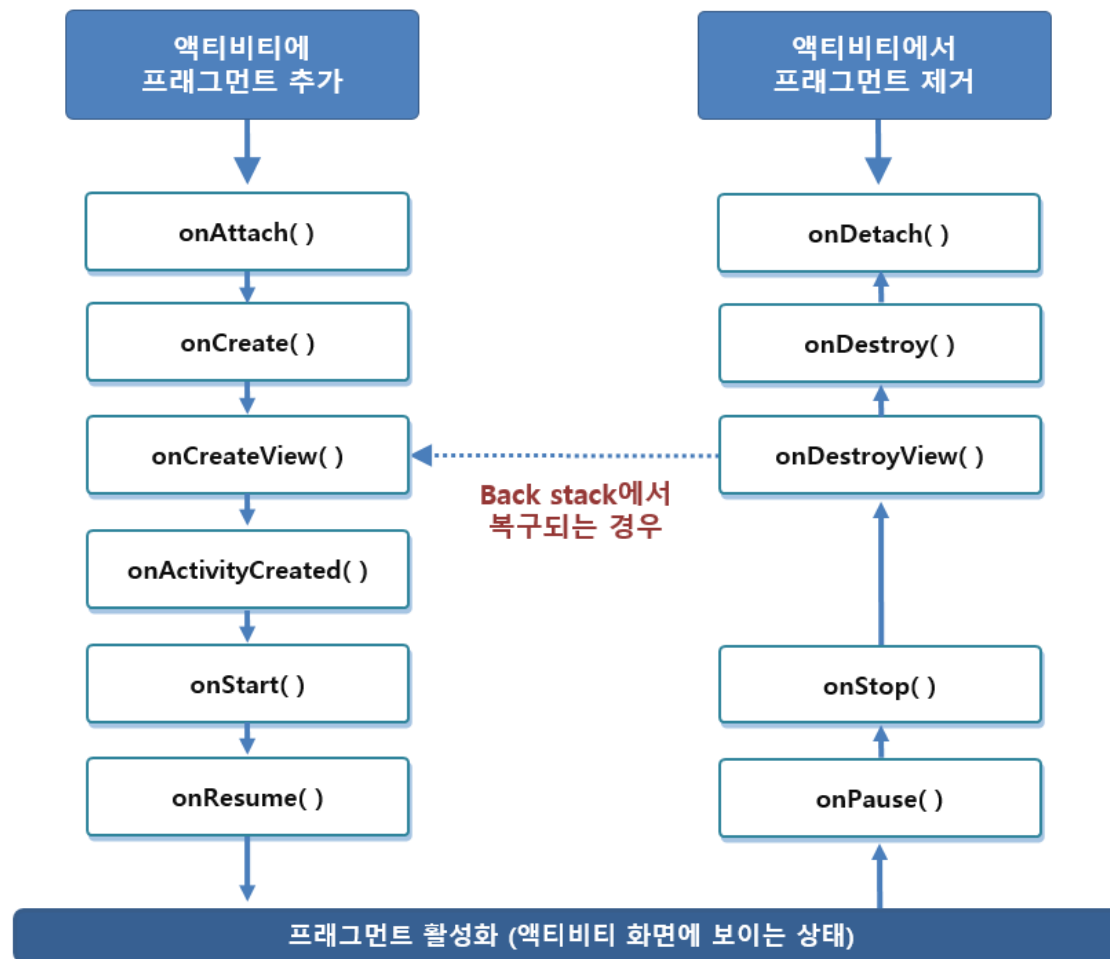


MenuFragment

```
public class MenuFragment extends Fragment {  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
        ViewGroup rootView  
            = (ViewGroup) inflater.inflate(R.layout.fragment_menu, 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(1);  
            }  
        });  
        return rootView;  
    }  
}
```




프래그먼트의 수명주기

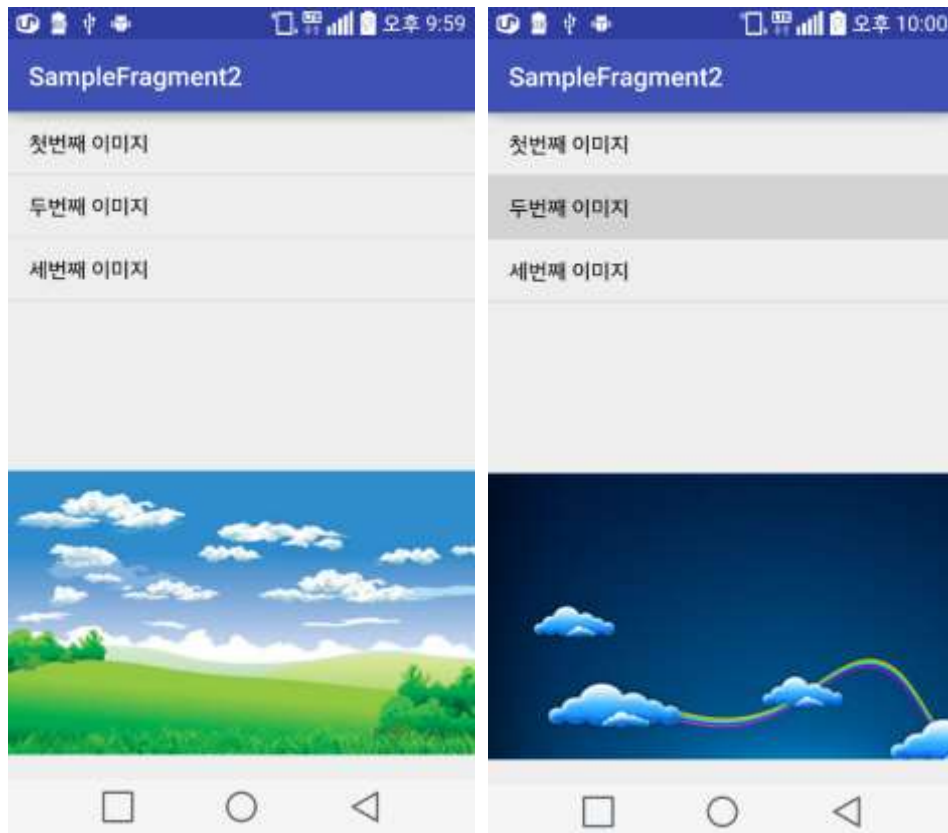


2.

프래그먼트로 화면 만들기



한 화면에 두 개의 프래그먼트 넣기



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

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



액티비티의 코드에서 프래그먼트 전환

```
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]);  
}
```



activity_main.xml

```
<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:id="@+id/listFragment"
        android:name="com.example.fragmentapp2.ListFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1" />
    <fragment
        android:id="@+id/viewerFragment"
        android:name="com.example.fragmentapp2.ViewerFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1" />
</LinearLayout>
```



fragment_list.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">
    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="첫 번째 이미지" />
    <Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="두 번째 이미지" />
    <Button
        android:id="@+id/button3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="세 번째 이미지" />
</LinearLayout>
```



fragment_viewer.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/dream01" />
</LinearLayout>
```

```
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]);
}
}
```



```
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);  
    }  
}
```

```
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);
```

onAttach() 메서드는 프래그먼트가 액티비티 위에 올라오는
시점에 호출됨

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

```
@Nullable
```

```
@Override
```

```
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);
        }
    }
});
Button button3 = rootView.findViewById(R.id.button3);
button3.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        if (callback != null) {
            callback.onImageSelected(2);
        }
    }
});
return rootView;
}
```



ListFragment

- onAttach() 메서드는 프래그먼트가 액티비티 위에 올라오는 시점에 호출됨
- 프래그먼트에서 해당 액티비티를 참조하고 싶다면 onAttach() 메서드로 전달되는 파라미터를 참조하거나 getActivity() 메서드를 호출하여 반환되는 객체를 참조할 수 있다.
- 그리고 그 객체를 변수에 할당하면 프래그먼트 클래스 안에서 자유롭게 액티비티 객체를 참조할 수 있게 됨
- onAttach() 메서드가 호출될 때 callback 변수에 객체가 할당되었는데 그 자료형이 Activity가 아니라 ImageSelectionCallback이다
- 왜?
- 화면에서 선택된 버튼에 따라 다른 프래그먼트의 이미지를 바꿔주려면 액티비티 쪽으로 데이터를 전달해야 하므로 액티비티에 onImageSelected() 메서드를 정의한 후 그 메서드를 호출하도록 함
- 그런데 매번 액티비티마다 다른 이름의 메서드를 만들면 프래그먼트가 올라간 액티비티가 다른 액티비티로 변경되었을 때 해당 액티비티가 무엇인지 매번 확인해야 하는 번거로움이 있다.
- 이 때문에 인터페이스를 하나 정의한 후 액티비티가 이 인터페이스를 구현하도록 하는 것이 좋다
- 여기에서는 ImageSelectionCallback 인터페이스를 정의했으며 onImageSelected() 메서드는 이 인터페이스 안에 정의했다



ListFragment

- 만약 MainActivity가 이 인터페이스를 구현하도록 했다면 이 프래그먼트에서는 액티비티 객체를 참조할 때 인터페이스 타입으로 참조한 후 `onImageSelected()` 메서드를 호출할 수 있다.
- `onAttach()` 메서드안에서는 MainActivity 객체를 참조한 후 `ImageSelectionCallback` 타입으로 된 `callback` 변수에 할당함
- 그러면 버튼이 클릭되었을 때 `callback` 변수에 할당된 액티비티 객체의 `onImageSelected()` 메서드를 호출할 수 있다

ViewerFragment

- `setImage()` 메서드를 만들어 액티비티에서 이 프래그먼트에 있는 이미지뷰에 이미지를 설정할 수 있도록 한다

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
- 옵션메뉴는 액션바에 포함되어 보이도록 만들어져 있다 (액션바 - 앱의 제목이 보이는 위쪽 부분)

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

옵션 메뉴와 컨텍스트 메뉴는 각각의 액티비티마다 설정할 수 있으므로
액티비티에 추가하고 싶은 경우에
onOptionsItemSelected()와 **onCreateContextMenu()** 메소드 재정의

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

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

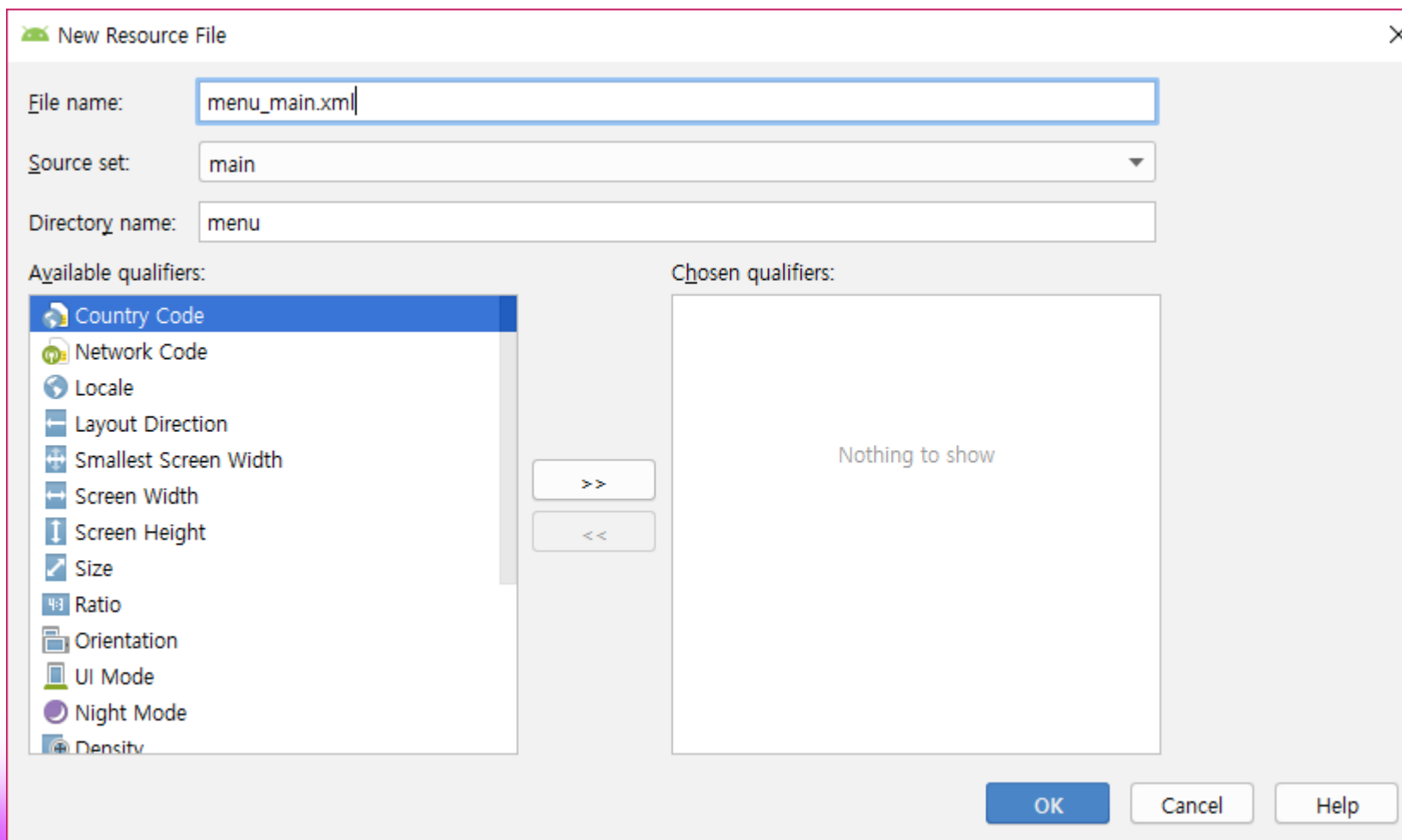
@Override
public boolean onContextItemSelected(MenuItem item) {
    return (applyMenuChoice(item) || super.onContextItemSelected(item));
}
```

[Reference]

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




메뉴를 XML에 정의





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

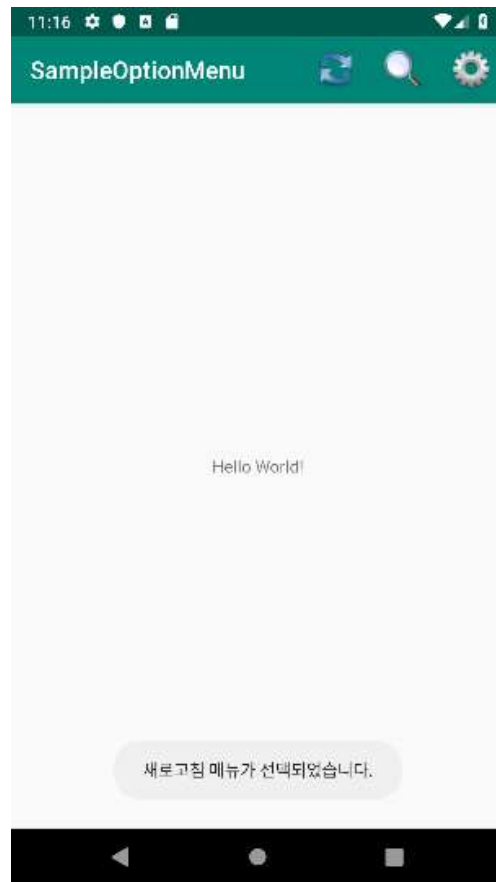
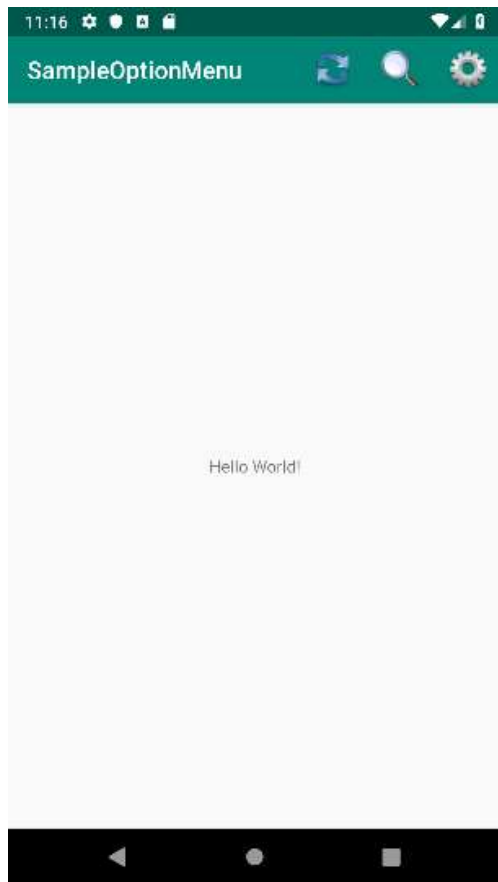
```
public boolean onCreateOptionsMenu(Menu menu) {  
    getMenuInflater().inflate(R.menu.menu_main, menu);  
    return true;  
}
```

@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





menu_main.xml

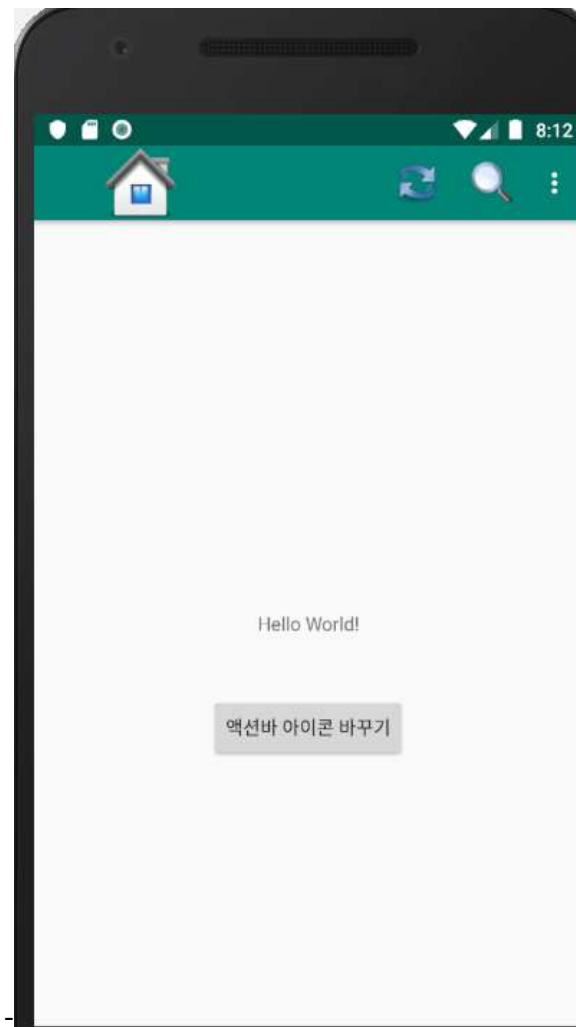
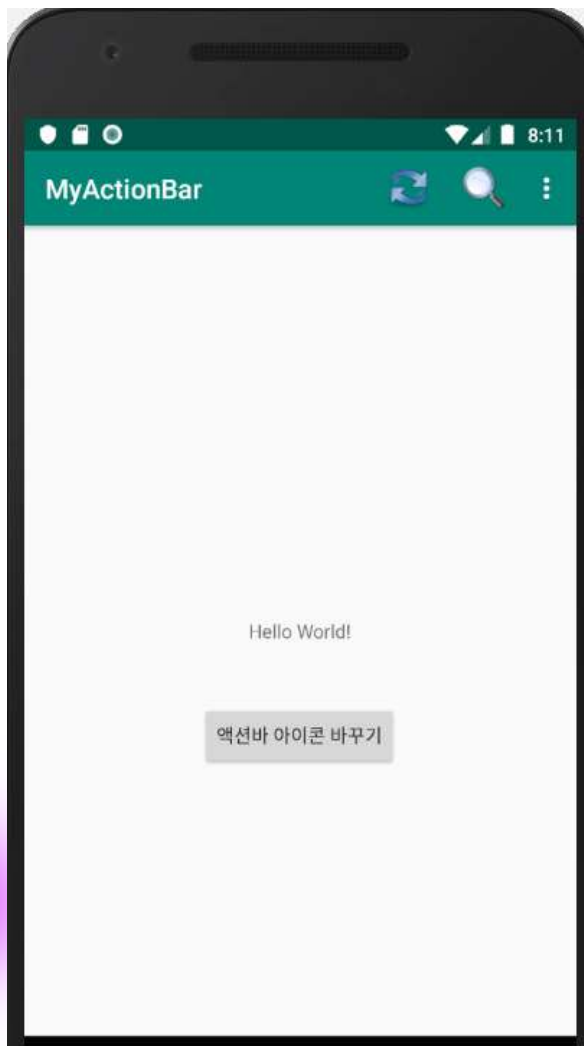
```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">
  <item
    android:id="@+id/menu_refresh"
    android:icon="@drawable/menu_refresh"
    android:title="새로고침"
    app:showAsAction="always" />
  <item
    android:id="@+id/menu_search"
    android:icon="@drawable/menu_search"
    android:title="검색"
    app:showAsAction="always" />
  <item
    android:id="@+id/menu_settings"
    android:icon="@drawable/menu_settings"
    android:title="설정"
    app:showAsAction="always" />
</menu>
```

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    @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;
            case R.id.menu_settings:
                Toast.makeText(this, "설정 메뉴가 선택되었습니다.", Toast.LENGTH_SHORT).show();
                break;
            default:
                break;
        }
        return super.onOptionsItemSelected(item);
    }
}
```



액션바

- 액티비티의 위쪽에 보이는 타이틀 부분과 옵션 메뉴는 액션바로 합쳐져 보이게 됨
- 액션바는 기본적으로 제목을 보여주는 타이틀의 기능을 하므로 앱의 제목을 보여줄 수 있으며 화면에 보이거나 보이지 않도록 만들 수도 있다.





activity_main.xml

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="44dp"
    android:text="액션바 아이콘 바꾸기"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView" />
```




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:icon="@drawable/menu_refresh"
        android:title="새로고침"
        app:showAsAction="always" />
    <item
        android:id="@+id/menu_search"
        android:icon="@drawable/menu_search"
        android:title="검색"
        app:showAsAction="always|withText" />
    <item
        android:id="@+id/menu_settings"
        android:icon="@drawable/menu_settings"
        android:title="설정"
        app:showAsAction="never" />
</menu>
```

```
public class MainActivity extends AppCompatActivity {
```

```
    ActionBar abar;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        abar = getSupportActionBar();
```

getSupportActionBar() - XML 레이아웃에 들어 있는

ActionBar객체를 참조함

ActionBar객체는 직접 XML레이아웃에 추가할 수도 있고 액티비티에 적용한 테마에 따라 자동으로 부여될 수도 있다

```
        Button button = findViewById(R.id.button);
```

```
        button.setOnClickListener(new View.OnClickListener() {
```

```
            @Override
```

```
            public void onClick(View v) {
```

```
                abar.setLogo(R.drawable.home);
```

```
                abar.setDisplayOptions(ActionBar.DISPLAY_SHOW_HOME|ActionBar.DISPLAY_USE_LOGO);
```

```
            }
```

```
        });
```

```
    }
```

버튼을 클릭했을 때 액션바가 보이는 모양을 바꾸도록

abar.setDisplayOptions() 메서드를 사용

ActionBar.DISPLAY_SHOW_HOME - 홈 아이콘을 표시

ActionBar.DISPLAY_USE_LOGO - 홈 아이콘 부분에 로고 아이콘을 사용

```
    @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;  
        case R.id.menu_settings:  
            Toast.makeText(this, "설정 메뉴가 선택되었습니다.", Toast.LENGTH_SHORT).show();  
            break;  
        default:  
            break;  
    }  
  
    return super.onOptionsItemSelected(item);  
}  
  
}
```



액션바에 입력상자 넣기

입력상자에 검색어를 입력한 후 키패드에서 '완료'키를 누르면 검색기능을 수행할 수 있도록 한다.

이 XML 레이아웃을 액션바에 넣어서 보여주려면 액션바에 추가된 메뉴아이템 중 하나가 화면에 보일 때 이 레이아웃이 보이게 설정해야 함

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;
                }
            });
        }
    }
}

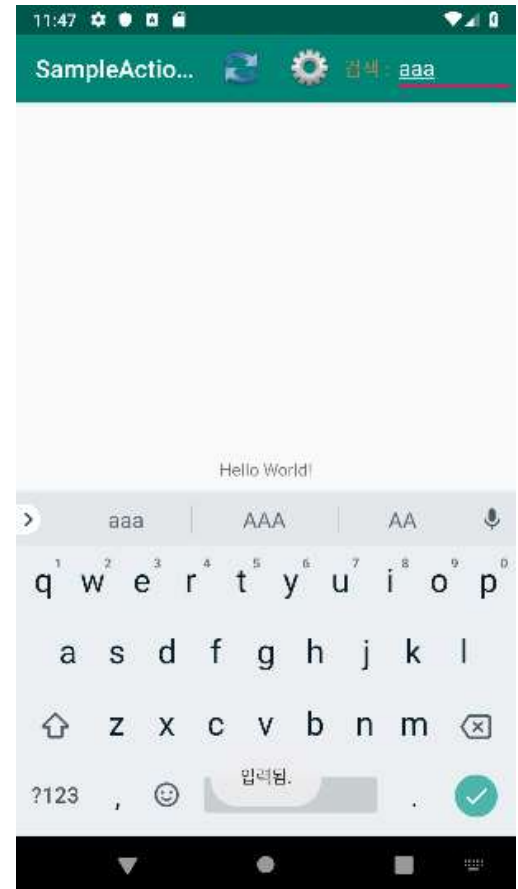
```

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





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" />

    <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"
        />
</LinearLayout>
```



menu_main.xml

```
<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:icon="@drawable/menu_refresh"
    android:title="새 로 고 침"
    app:showAsAction="always" />
```

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

orderInCategory - 메뉴가 보이는 순서를 결정
작은 숫자부터 순서대로 지정

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

```
</menu>
```



```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    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;
                    }
                });
            }
        }
        return true;
    }
}
```


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.google.android.material:material:1.0.0'

}
```

하나의 뷰에서 여러 개의 정보를 볼 때 일반적으로 사용하는 뷰로 탭(Tab)을 들 수 있다
상단에 있는 탭을 누를 때마다 내용이 보이는 화면 영역이 전환되어 나타남
탭은 내비게이션 위젯이라고 불리기도 하며 상단 탭과 하단 탭으로 구분할 수 있다
최근에는 하단탭을 더 많이 사용



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">

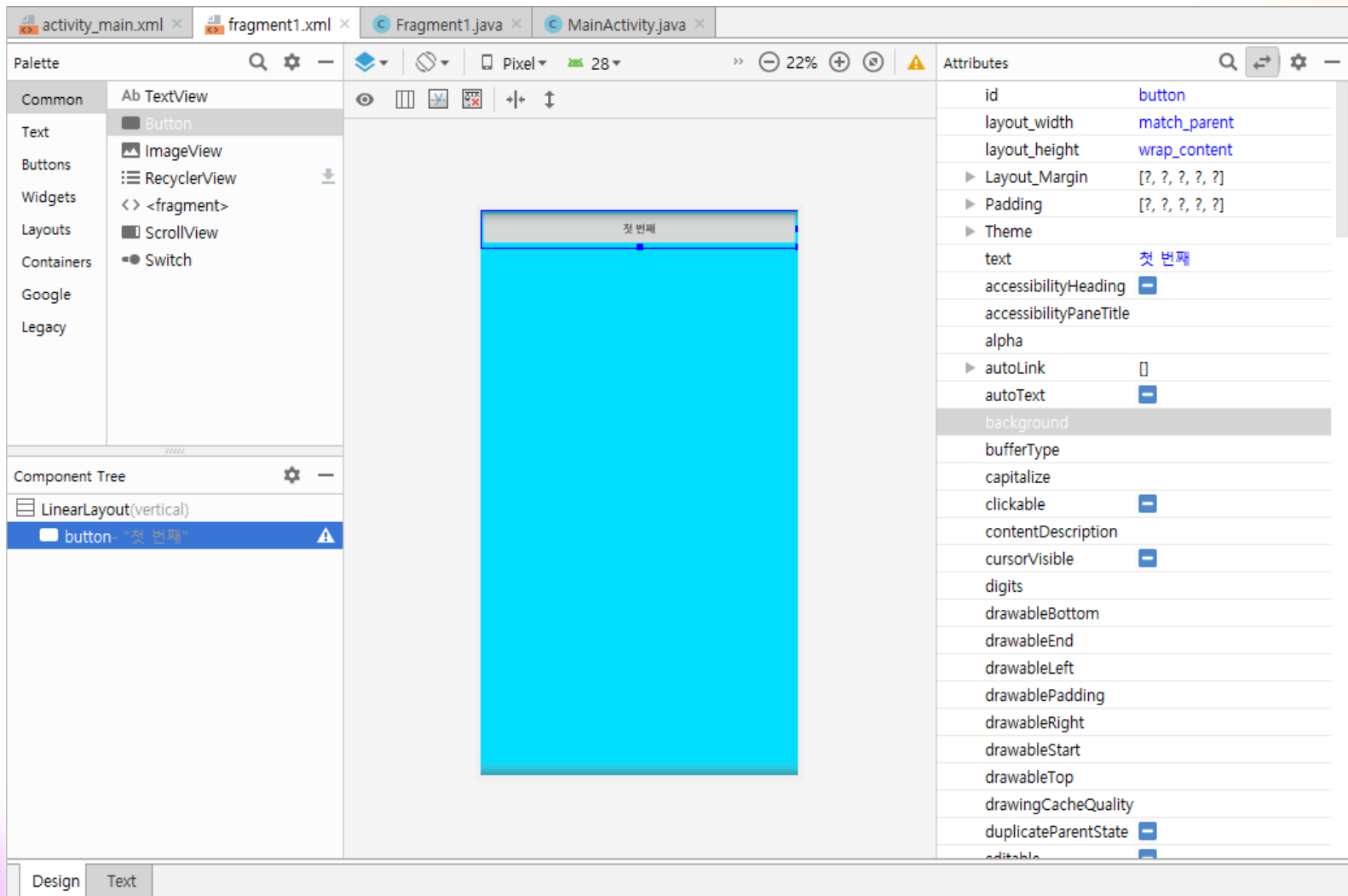
    </FrameLayout>
</android.support.design.widget.CoordinatorLayout>
</RelativeLayout>
```

tabMode=fixed
tabGravity=fill
⇒[탭]버튼들이 동일한 크기를 갖게 한다

FrameLayout 의 id값은 container로 설정하여
자바 소스코드에서 이 안에 프래그먼트를 넣을
수 있도록 한다



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();
```

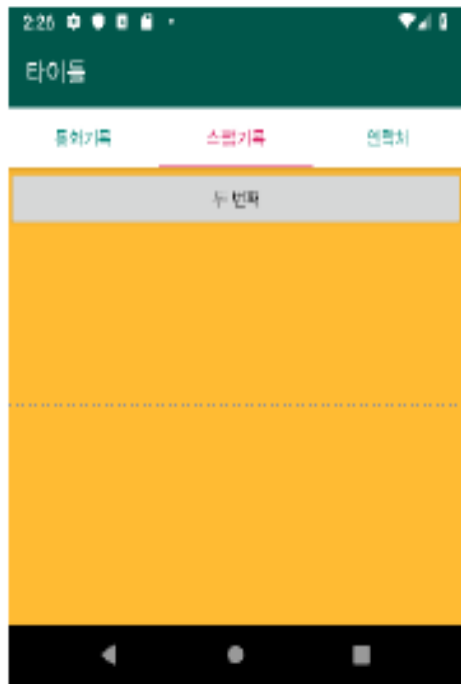
```
getSupportFragmentManager().beginTransaction().replace(R.id.container, fragment1).commit();
```

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



앱 실행 결과





activity_main.xml

```
<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">
    <androidx.coordinatorlayout.widget.CoordinatorLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <com.google.android.material.appbar.AppBarLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">
            <androidx.appcompat.widget.Toolbar
                android:id="@+id/toolbar"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:background="@color/colorPrimaryDark"
                android:elevation="1dp"
                android:theme="@style/ThemeOverlay.AppCompat.Dark">
```

```
<TextView
    android:id="@+id/titleText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="타이틀"
    android:textAppearance="@style/Base.TextAppearance.Widget.AppCompat.Toolbar.Title" />
</androidx.appcompat.widget.Toolbar>
<com.google.android.material.tabs.TabLayout
    android:id="@+id/tabs"
    android:layout_width="match_parent"    android:layout_height="wrap_content"
    android:background="@android:color/background_light"
    android:elevation="1dp"
    app:tabGravity="fill"
    app:tabMode="fixed"
    app:tabSelectedTextColor="@color/colorAccent"
    app:tabTextColor="@color/colorPrimary" >
    </com.google.android.material.tabs.TabLayout>
</com.google.android.material.appbar.AppBarLayout>
<FrameLayout
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior">
    </FrameLayout>
</androidx.coordinatorlayout.widget.CoordinatorLayout>
</RelativeLayout>
```



fragment1.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@android:color/holo_blue_bright"
    android:orientation="vertical"
    tools:context=".Fragment1">

    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="첫 번째" />

</LinearLayout>
```



fragment2.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@android:color/holo_orange_light"
    android:orientation="vertical"
    tools:context=".Fragment1">

    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="두 번째" />

</LinearLayout>
```



fragment3.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@android:color/holo_purple"
    android:orientation="vertical"
    tools:context=".Fragment1">

    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="세 번째" />

</LinearLayout>
```



Fragment1

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



Fragment2

```
public class Fragment2 extends Fragment {  
  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle  
        savedInstanceState) {  
        return inflater.inflate(R.layout.fragment2, container, false);  
    }  
}
```



Fragment3

```
public class Fragment3 extends Fragment {  
  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle  
        savedInstanceState) {  
        return inflater.inflate(R.layout.fragment3, container, false);  
    }  
}
```



```

public class MainActivity extends AppCompatActivity {
    Toolbar toolbar;

    Fragment1 fragment1;
    Fragment2 fragment2;
    Fragment3 fragment3;

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

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

        ActionBar actionBar = getSupportActionBar();
        actionBar.setDisplayShowTitleEnabled(false);

        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"));
    }
}

```

XML 레이아웃에서 정의한 Toolbar 객체는 코드에서 `setSupportActionBar()` 메서드를 사용해 액션바로 설정해야 함
`setSupportActionBar()` 메서드는 액티비티에 디폴트로 만들어진 액션바가 없을 경우에만 동작함
프로젝트가 만들어질 때 메인 액티비티에는 자동으로 액션바가 만들어짐
이것은 테마를 액션바가 들어 있는 테마로 설정했기 때문
액티비티에 설정된 테마를 변경하기 위해 `style.xml`의 `AppTheme` 라는 `name` 속성값을 가진 `<style>` 태그에서 `parent` 속성의 값을 `액션바가 없는 스타일(NoActionBar)`로 변경
`NoActionBar` 스타일로 바꾸면 이 스타일을 적용한 액티비티에는 액션바가 만들어지지 않는다
코드에서 `setSupportActionBar()` 메서드를 호출하여 직접 액션바를 설정해야 함

```

tabs.addTab(tabs.newTab().setText("스팸기록"));
tabs.addTab(tabs.newTab().setText("연락처"));
tabs.addTabOnTabSelectedListener(new TabLayout.OnTabSelectedListener() {
    @Override
    public void onTabSelected(TabLayout.Tab tab) {
        int position = tab.getPosition();
        Log.d("MainActivity", "선택된 탭 : " + position);

        Fragment selected = null;
        if (position == 0) {
            selected = fragment1;
        } else if (position == 1) {
            selected = fragment2;
        } else if (position == 2) {
            selected = fragment3;
        }
        getSupportFragmentManager().beginTransaction()
            .replace(R.id.container, selected).commit();
    }

    @Override
    public void onTabUnselected(TabLayout.Tab tab) {
    }

    @Override
    public void onTabReselected(TabLayout.Tab tab) {
    }
});
}
}

```

TabLayout에는 addTab()메서드가 있어서 [탭]버튼을 추가할 수 있게 함

세개의 [탭]버튼 추가

각각의 [탭]버튼을 눌렀을 때 container 라는 id를 가진
FrameLayout 안에 각각의 [탭]버튼에 해당하는 프래그먼트
화면이 보이도록 함

TabLayout에는 OnTabSelectedListener를 설정할 수 있는데 이
리스너는 [탭]버튼이 선택될 때 마다 그 리스너 안에 있는
onTabSelected() 메서드가 호출되도록 함
이 메서드로는 현재 선택된 탭 객체가 전달되므로 탭의 position
정보를 확인한 후 그 값이 0 일때는 첫번째 프래그먼트, 1일때는
두번째 프래그먼트를 FrameLayout안에 추가함



style.xml

```
<resources>
  <!-- Base application theme. -->
  <style name="AppTheme" parent="Theme.AppCompat.NoActionBar">
    <!-- Customize your theme here. -->
    <item name="colorPrimary">@color/colorPrimary</item>
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
    <item name="colorAccent">@color/colorAccent</item>
  </style>
</resources>
```

액티비티에 설정된 테마를 변경하기 위해 style.xml의 AppTheme 라는 name 속성값을 가진 <style>태그에서 parent 속성의 값을 액션바가 없는 스타일(NoActionBar)로 변경
NoActionBar 스타일로 바꾸면 이 스타일을 적용한 액티비티에는 액션바가 만들어지지 않는다
코드에서 `setSupportActionBar()` 메서드를 호출하여 직접 액션바를 설정해야 함



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 탭은 BottomNavigationView 위젯으로 만들 수 있다

이 위젯도 design 라이브러리를 사용
하단 탭에 보이는 각각의 탭에는 이미지나 글자가
들어갈 수 있는데 이 버튼들은 메뉴 XML 파일로
만든다

@android:drawable 은 기본 API에 포함된 이미지를
참조할 수 있도록 함



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>
```

itemBackground - 각 탭의 배경색
itemIconTint - 아이콘 색상
itemTextColor - 텍스트 색상
menu="@menu/menu_bottom"
=> menu_bottom.xml 파일의 내용이 탭으로 보인다



하단 탭 보여주기

각 탭을 눌렀을 때 가운데 보이는 프레임 레이아웃 안에 프래그먼트를 바꿔가며 보여줘야 함

탭이 선택되었을 때 `onNavigationItemSelectedListener()` 메서드가 호출됨

• 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





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="이 메 일" />
  <item
    android:id="@+id/tab2"
    app:showAsAction="ifRoom"
    android:icon="@android:drawable/ic_dialog_info"
    android:title="정 보" />
  <item
    android:id="@+id/tab3"
    app:showAsAction="ifRoom"
    android:enabled="true"
    android:icon="@android:drawable/ic_dialog_map"
    android:title="위 치" />
</menu>
```

Fragment1, Fragment2, Fragment3는 동일



activity_main.xml

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

<com.google.android.material.bottomnavigation.BottomNavigationView
    android:id="@+id/bottom_navigation"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintStart_toStartOf="parent"
    android:layout_marginEnd="0dp"
    android:layout_marginStart="0dp"
    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" />
```



item_color.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<selector xmlns:android="http://schemas.android.com/apk/res/android">  
    <item android:state_checked="true" android:color="#51032d" />  
    <item android:color="#CFD8DC" />  
</selector>
```

```
public class MainActivity extends AppCompatActivity {  
    Fragment1 fragment1;  
    Fragment2 fragment2;  
    Fragment3 fragment3;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        fragment1 = new Fragment1();  
        fragment2 = new Fragment2();  
        fragment3 = new Fragment3();  
  
        getSupportFragmentManager().beginTransaction().replace(R.id.container, fragment1).commit();  
  
        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();

        return true;
    case R.id.tab3:
        Toast.makeText(getApplicationContext(), "세 번째 탭 선택됨", Toast.LENGTH_LONG).show();
        getSupportFragmentManager().beginTransaction()
            .replace(R.id.container, fragment3).commit();

        return true;
}
return false;
}
});
}
```

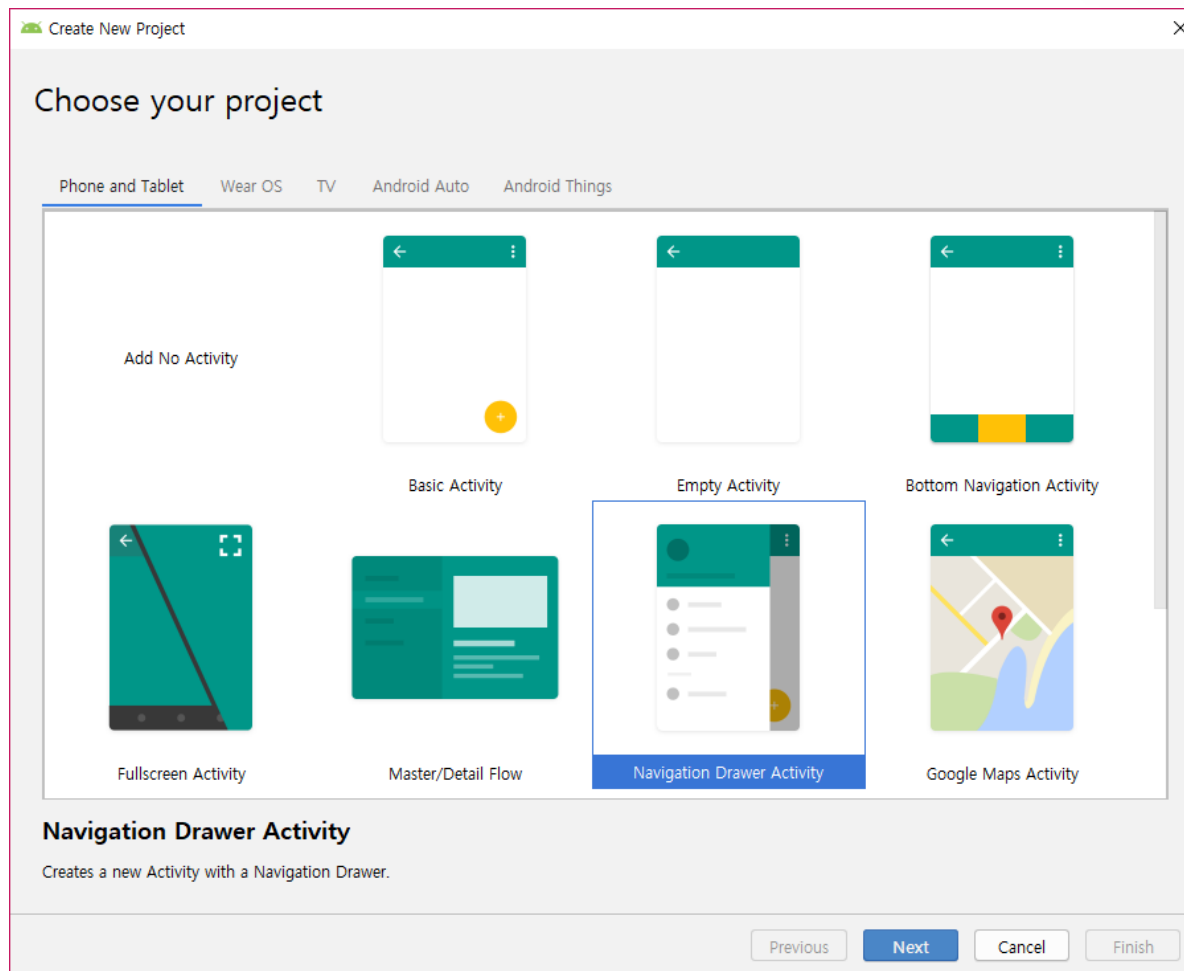
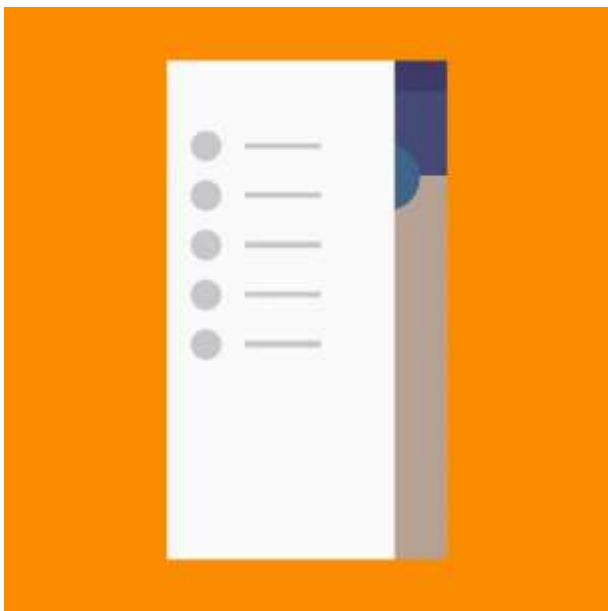
5

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>
```

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

기본테마가 아니라 상단의 액션바가
없는 스타일로 테마가 설정됨
액션바가 없어졌으니
activity_main.xml에서 직접 액션바를
만들어 준다

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>
```

액션바가 없어졌으니
activity_main.xml에서 직접 액션바를
만들어 준다
AppBarLayout 태그로 화면 상단에
액션바를 보여줄 수 있다



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>
```

NavigationView가 바로가기 메뉴를 위한 뷰이다
이 뷰는 평소에는 보이지 않다가 햄버거 모양 아이콘을
눌렀을 때만 보이게 됨
이런 동작을 위해 최상위 레이아웃은 DrawLayout으로 설정

FrameLayout의 속성으로
app:layout_behavior 속성이 부여되었는데 이 속성을
부여함으로써 CoordinatorLayout 안에서 해당 레이아웃이
스크롤등의 작업이 진행될 때 차지할 면적을 자동으로
계산하도록 만든다

NavigationView 의 headerLayout 속성 - 바로가기 메뉴의
상단에 표시되면서 사용자 프로필 등을 보여줄 수 있도록
한다

menu 속성 - 그 아래에 메뉴를 보여줄 수 있도록 한다



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

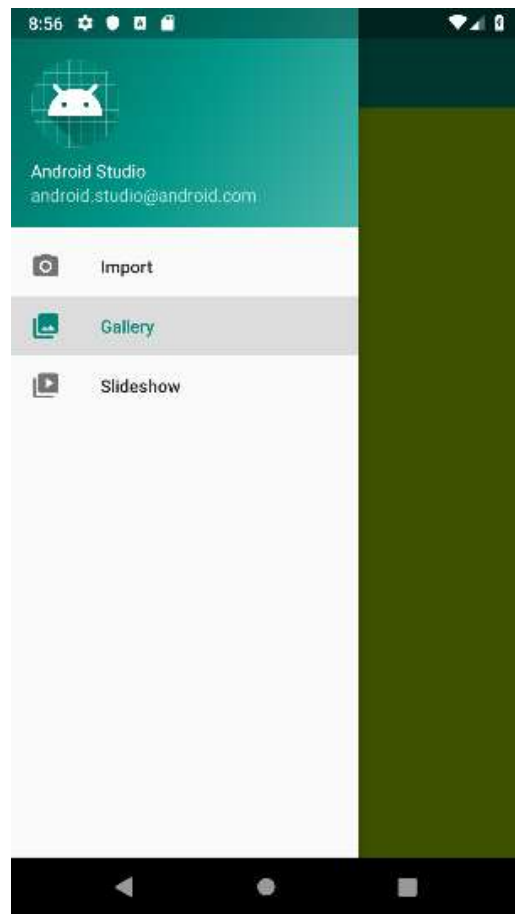
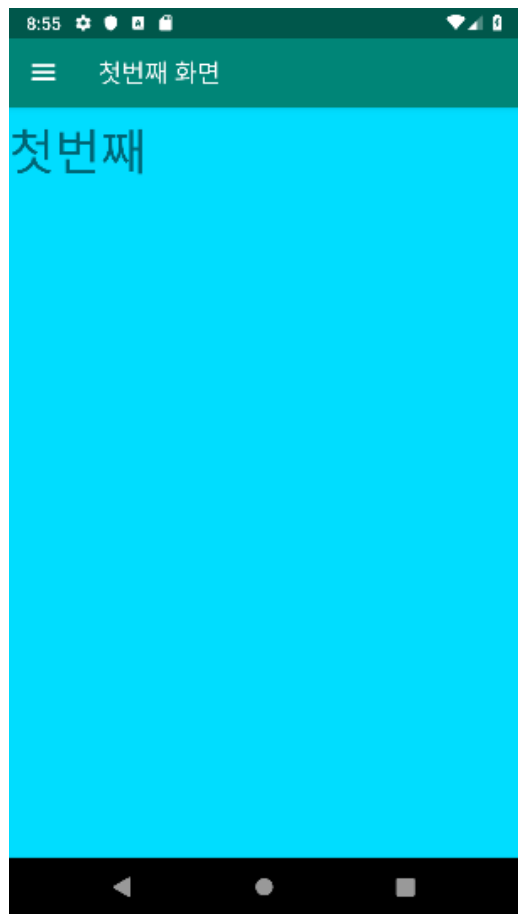
```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:tools="http://schemas.android.com/tools"
      tools:showIn="navigation_view">

    <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





```
dependencies {  
    implementation fileTree(dir: 'libs', include: ['*.jar'])  
    implementation 'androidx.appcompat:appcompat:1.1.0'  
    implementation 'com.google.android.material:material:1.0.0'  
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'  
    implementation 'androidx.navigation:navigation-fragment:2.1.0'  
    implementation 'androidx.navigation:navigation-ui:2.1.0'  
    implementation 'androidx.lifecycle:lifecycle-extensions:2.1.0'  
    testImplementation 'junit:junit:4.12'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.1'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'  
}
```




AndroidManifest.xml

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme.NoActionBar">
    <activity
        android:name=".MainActivity"
        android:label="@string/app_name"
        android:theme="@style/AppTheme.NoActionBar">
```



activity_main.xml

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

    <androidx.coordinatorlayout.widget.CoordinatorLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <com.google.android.material.appbar.AppBarLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:theme="@style/AppTheme.AppBarOverlay">

            <androidx.appcompat.widget.Toolbar
                android:id="@+id/toolbar"
                android:layout_width="match_parent"
```




activity_main.xml

```
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimary"
        app:popupTheme="@style/AppTheme.PopupOverlay" />
</com.google.android.material.appbar.AppBarLayout>
```

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

```
<com.google.android.material.navigation.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" />
</androidx.drawerlayout.widget.DrawerLayout>
```



fragment1.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout 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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@android:color/holo_blue_bright">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:text="첫 번째"
        android:textSize="40dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```



nav_header_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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="@dimen/nav_header_height"
    android:background="@drawable/side_nav_bar"
    android:gravity="bottom"
    android:orientation="vertical"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:theme="@style/ThemeOverlay.AppCompat.Dark">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:contentDescription="@string/nav_header_desc"
        android:paddingTop="@dimen/nav_header_vertical_spacing"
        app:srcCompat="@mipmap/ic_launcher_round" />
```



nav_header_main.xml

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:paddingTop="@dimen/nav_header_vertical_spacing"
    android:text="@string/nav_header_title"
    android:textAppearance="@style/TextAppearance.AppCompat.Body1" />
```

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/nav_header_subtitle" />
```

```
</LinearLayout>
```



activity_main_drawer.xml

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

```
public class MainActivity extends AppCompatActivity
```

```
    implements NavigationView.OnNavigationItemSelectedListener,
```

```
    FragmentCallback {
```

```
    Fragment1 fragment1;
```

```
    Fragment2 fragment2;
```

```
    Fragment3 fragment3;
```

```
    Toolbar toolbar;
```

```
@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();
}

@Override
public void onBackPressed() {
    DrawerLayout drawer = findViewById(R.id.drawer_layout);
    if (drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawer(GravityCompat.START);
    } else {
        super.onBackPressed();
    }
}
}

```

onBackPressed()메서드는 시스템 [BACK]키를 눌렀을 때 호출되는 메서드로 여기에는 바로가기 메뉴가 열려 있을 경우 달는 코드가 추가되어 있다.

```

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

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

```

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

@Override

```
public void onFragmentSelected(int position, Bundle bundle) {
```

```
    Fragment curFragment = null;
```

```
    if (position == 0) {
```

```
        curFragment = fragment1;
```

```
        toolbar.setTitle("첫 번째 화면");
```

```
    } else if (position == 1) {
```

```
        curFragment = fragment2;
```

```
        toolbar.setTitle("두 번째 화면");
```

```
    } else if (position == 2) {
```

```
        curFragment = fragment3;
```

```
        toolbar.setTitle("세 번째 화면");
```

```
    }
```

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

```
}
```

```
}
```




FragmentCallback

```
public interface FragmentCallback {
```

```
    public void onFragmentSelected(int position, Bundle bundle);
```

```
}
```

FragmentCallback 인터페이스는 어떤
프래그먼트를 보여줄지 선택하는 메서드를
포함

```
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
```

```
public class Fragment1 extends Fragment {
```

```
    @Nullable
```

```
    @Override
```

```
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
        ViewGroup rootView = (ViewGroup) inflater.inflate(R.layout.fragment1, container, false);
```

```
        return rootView;
```

```
    }
```

```
}
```



[References]

- 기본 서적
2019, 정재곤, "Do it! 안드로이드 앱 프로그래밍(개정6판)", 이지스퍼블리싱(주)
- Android Website
<http://www.android.com/>
- Google Developer's Conference
<http://code.google.com/events/io/>
- Android SDK Documentation