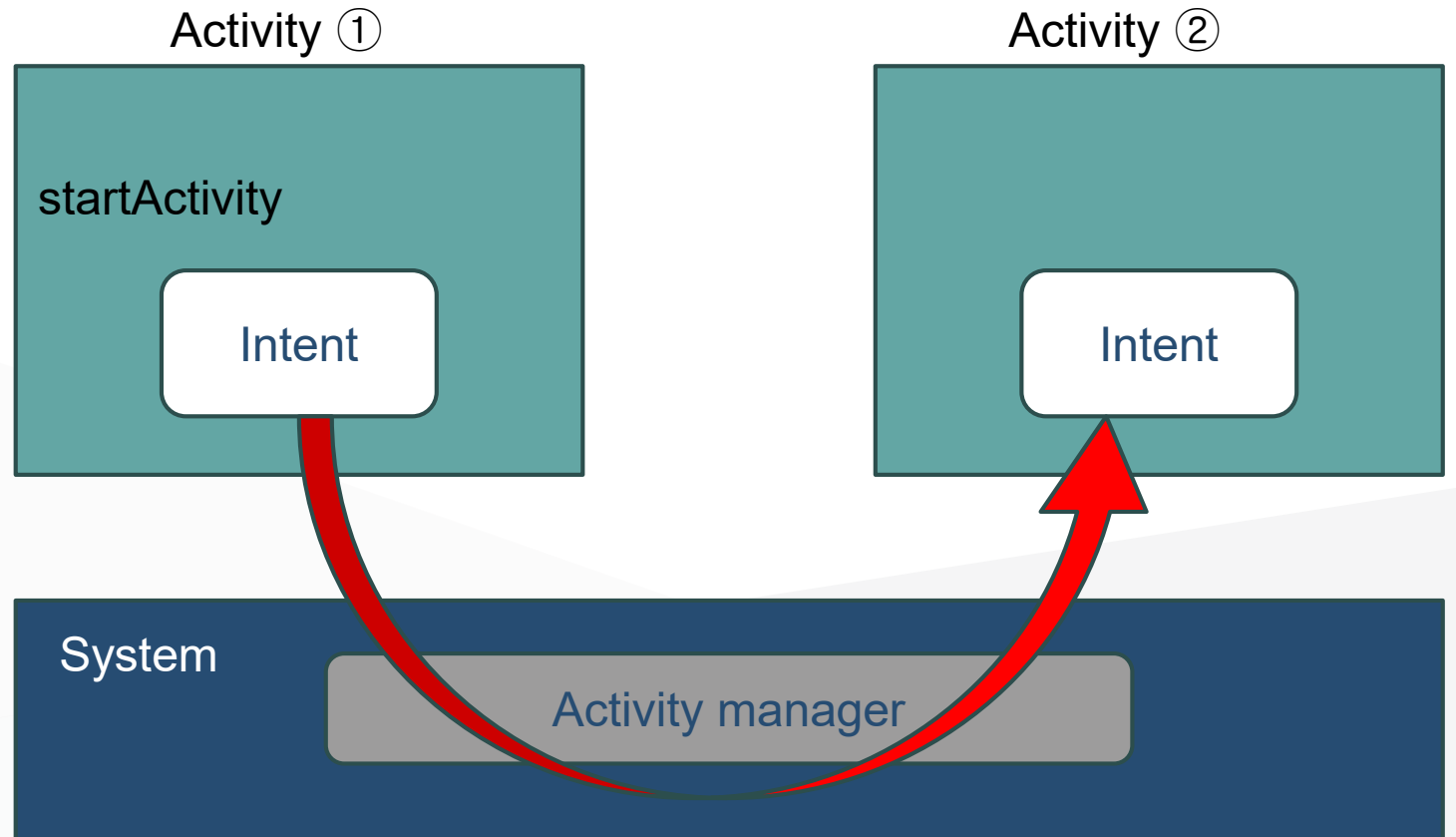




Screen transition User interface

Screen transition

- 하나의 화면은 Activity
- 하나의 화면에서 다른 화면으로 전환이 가능하다.
- startActivity : Activity간에 연결을 해주는 것.
- Intent : 클래스



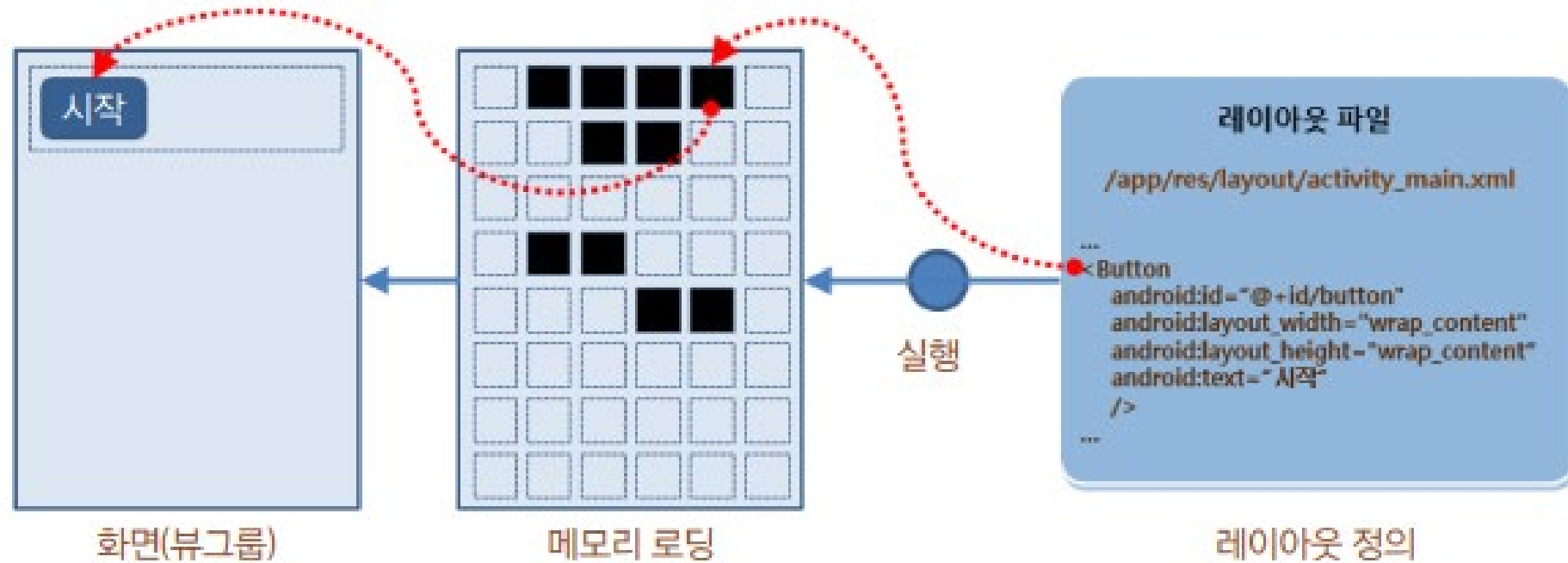
Layout Inflation

- 두 개의 **XML** 레이아웃 파일과 하나의 소스코드만 있다면 어떤 **XML** 레이아웃 파일이 소스 파일과 연결되는 것일까?
- **XML** 레이아웃 파일의 내용을 메모리상에 로드하여 화면에 보여주는 과정을 인플레이션이라고 한다.
- 인플레이션은 크게 **2**가지가 있다. 전체 인플레이션과 부분 인플레이션이다.

전체 인플레이션

- 전체 인플레이션은 자바파일의 기본메소드인 `onCreate` 에 기본 정의된 `setContentView` 를 통해 구현됩니다.
- 전체 인플레이션 사용방법
- `setContentView(R.layout.[XML 레이아웃 파일명]);`

전체 인플레이션



- xml 레이아웃은 앱이 실행되는 시점에 메모리에 객체화
- xml 레이아웃에 `<button>`태그를 정의해도 앱은 자신이 실행되기 전까지 버튼이 있는지 모름
- `setContentView()` 메서드가 화면에 표시할 xml 레이아웃을 지정하거나 화면에 표시할 뷰 객체를 지정하는 역할
- `setContentView(R.layout.activity_main)=>activity_main.xml` 파일을 파라미터로 전달하여 그 xml 레이아웃에 들어 있는 뷰들이 객체화 되도록 함.

전체 인플레이션

XML과 JAVA 연결



```
ty_main.xml × MainActivity.java ×  
package com.example.myapplication;
```

```
import ...
```

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

```
@Override
```

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

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

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

```
}
```

앱이 실행될 때 xml 레이아웃의 내용이 메모리에 객체화. 객체화된 xml 레이아웃을 소스 파일에서 사용
=> Inflation

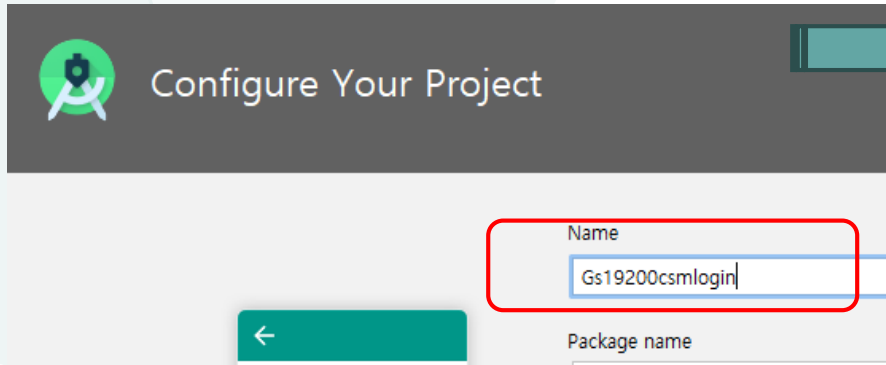


res/layout 폴더의 activity_main.xml 파일과 java와 연결 확장자 없이 파라미터로 전달.
앱을 실행하면 xml 레이아웃(화면배치정보)을 소스 코드에서 사용

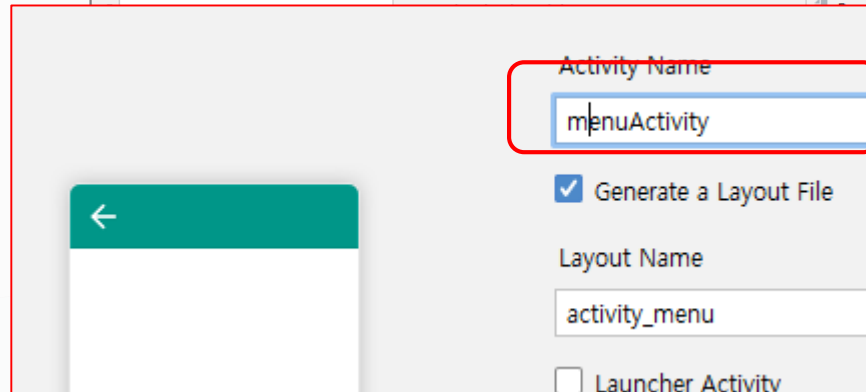
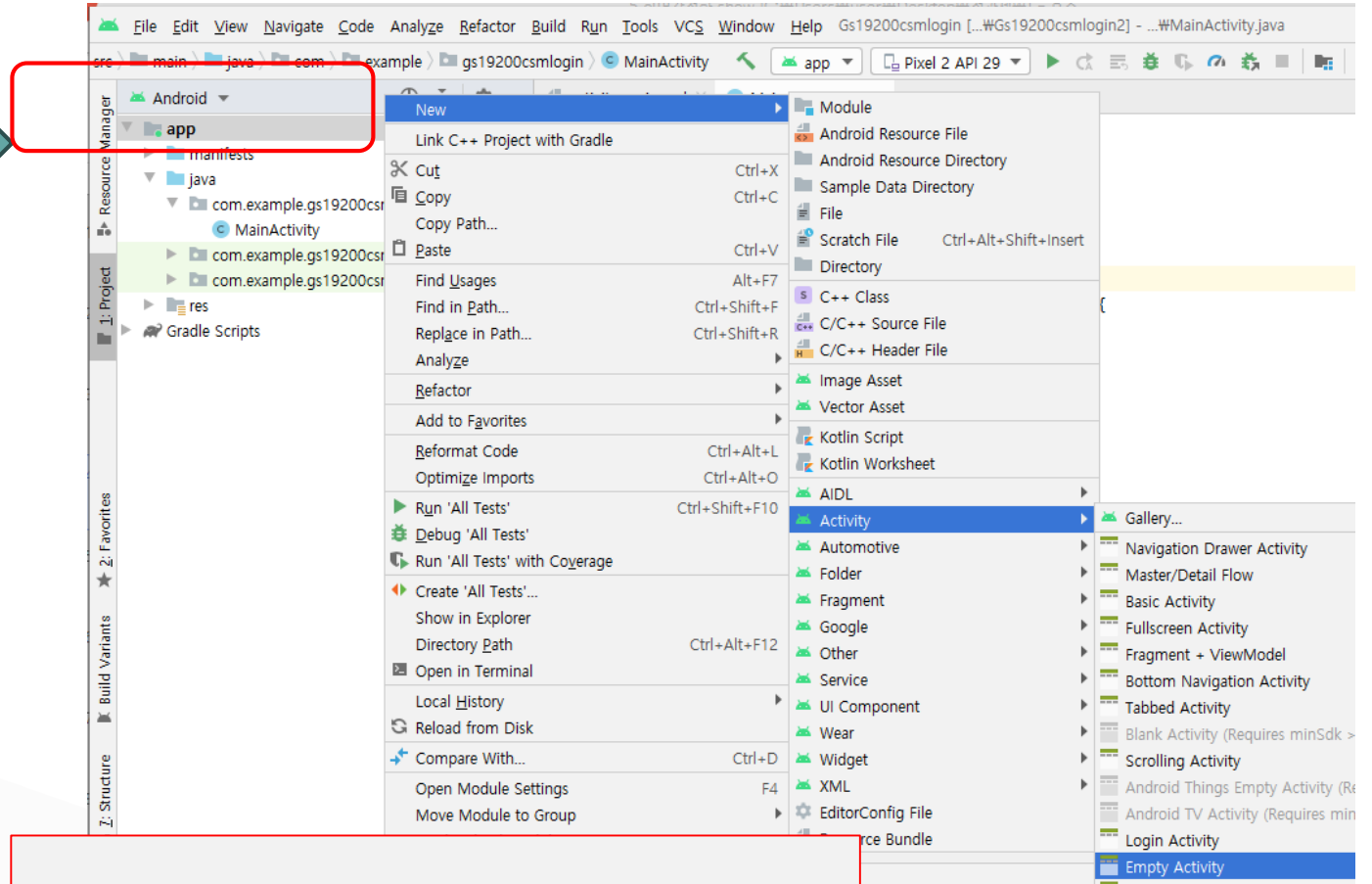
부분 인플레이션

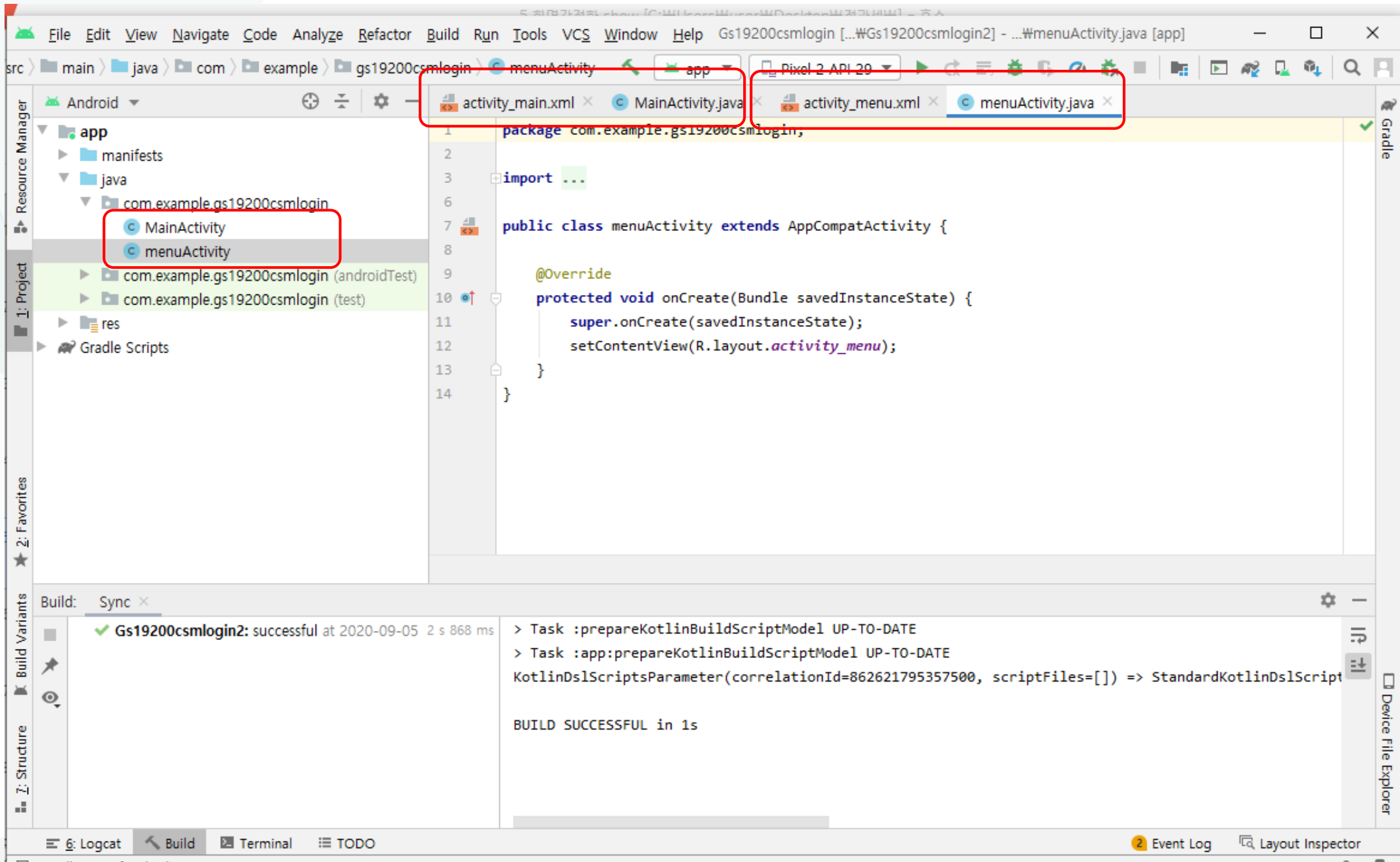
- 부분 인플레이션을 이용하려면 **LayoutInflater** 라는 클래스를 이용하여 사용할 수 있습니다.
- - 부분 인플레이션 사용방법
-
- `LayoutInflater inflater = (LayoutInflater)
getSystemService(Context.LAYOUT_INFLATER_SERVICE);`
- `inflater.inflate(R.layout.[XML 레이아웃 파일명], [레이아웃 객체명], true);`

부분 인플레이션



새 프로젝트-
app 오른쪽 버튼
activity 추가





```

1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:orientation="vertical">
6
7      <TextView
8          android:layout_width="match_parent"
9          android:layout_height="wrap_content"
10         android:id="@+id/textView"
11         android:text="버튼을 눌러 부분 화면을 추가하세요"
12         android:textSize="20sp"/>
13
14     <Button
15         android:layout_width="match_parent"
16         android:layout_height="wrap_content"
17         android:id="@+id/button"
18         android:text="추가하기"/>
19
20     <LinearLayout
21         android:layout_width="match_parent"
22         android:layout_height="match_parent"
23         android:id="@+id/container"
24         android:orientation="vertical">
25
26 </LinearLayout>
27

```

Common

Ab TextView

Text

Buttons

Widgets

Layouts

Containers

Google

Legacy

Button

ImageView

RecyclerView

<> <fragment>

ScrollView

Switch

Component Tree

LinearLayout (vertical)

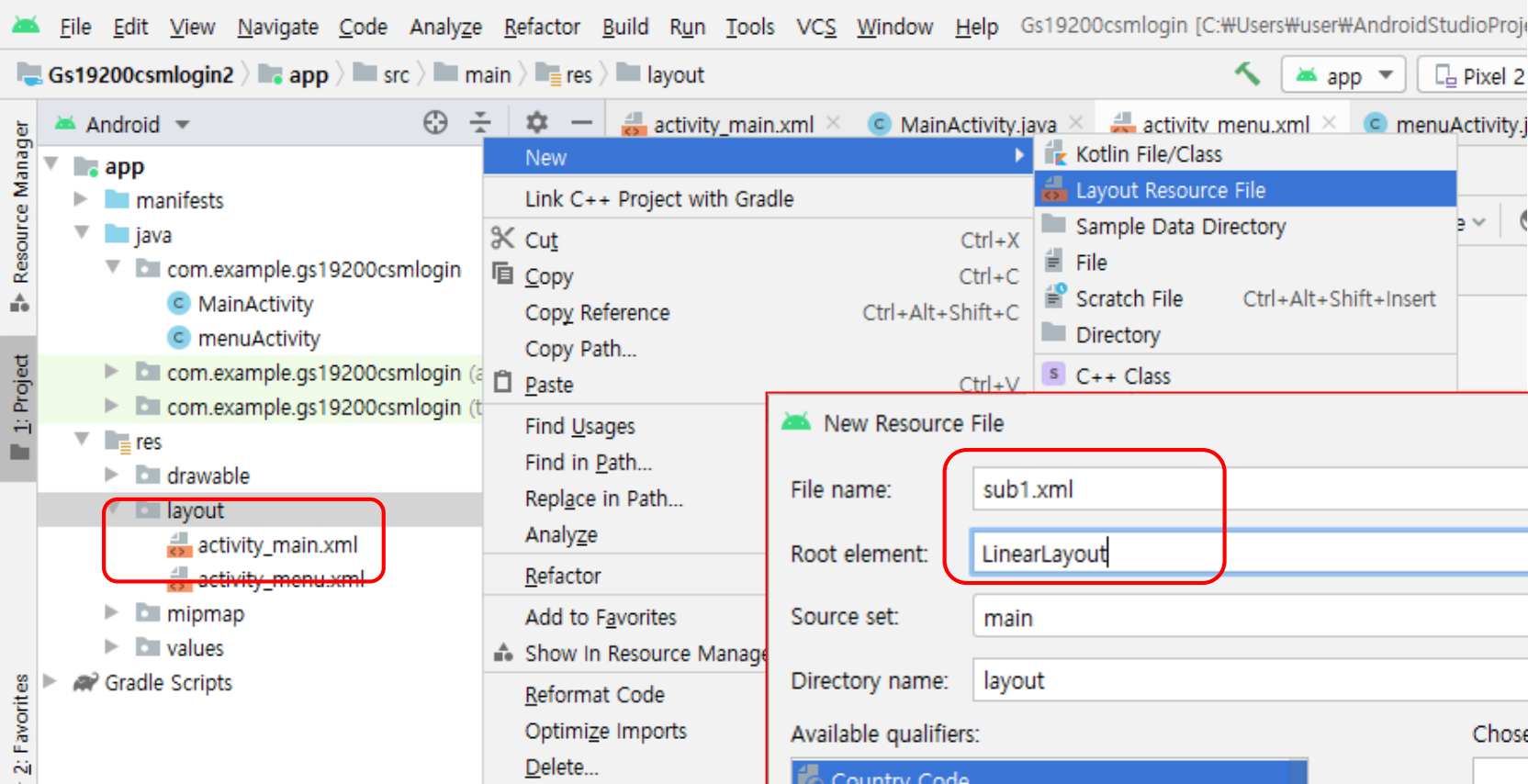
Ab textView "버튼을 눌러 부분 ..."

button "추가하기"

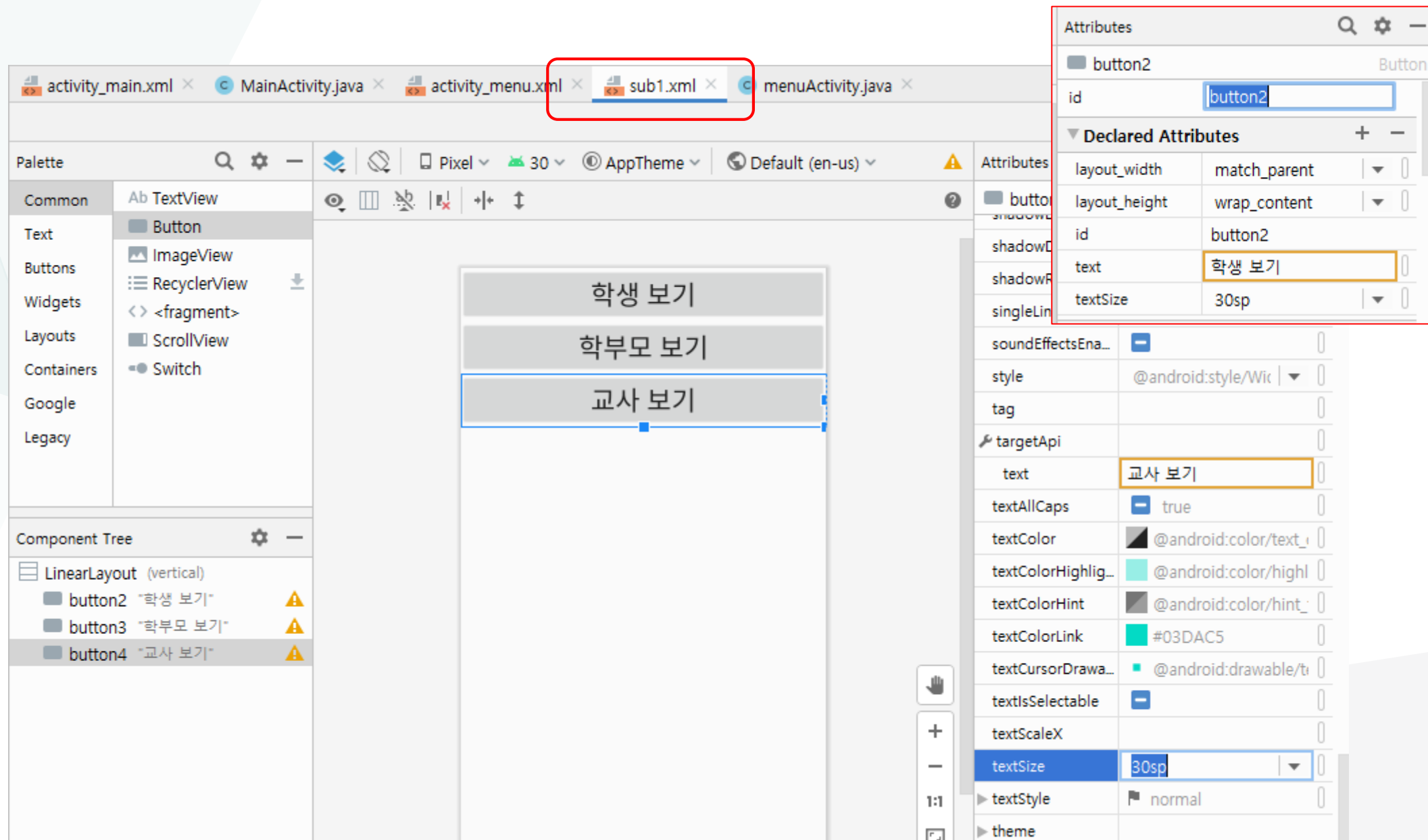
container (vertical)

버튼을 눌러 부분 화면을 추가하세요

추가하기



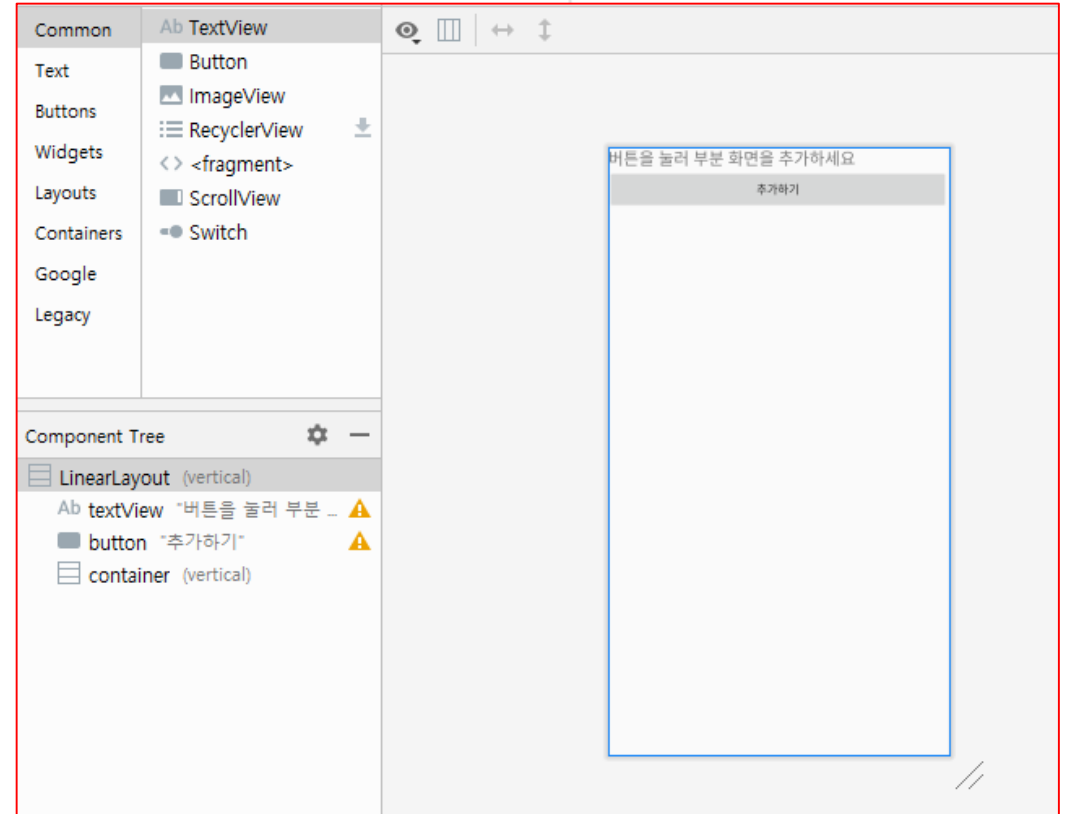
res/layout 클릭
마우스 오른쪽 버튼



```

1 package com.example.gs19200csmlogin;
2
3 import ...
4
11
12 public class menuActivity extends AppCompatActivity {
13     LinearLayout container;
14
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.activity_menu);
19
20         container = findViewById(R.id.container);
21
22         Button button = findViewById(R.id.button);
23         button.setOnClickListener(new View.OnClickListener() {
24             @Override
25             public void onClick(View view) {
26                 LayoutInflater inflater = (LayoutInflater) getSystemService(Context.LAYOUT_INFLATER_SERVICE);
27                 inflater.inflate(R.layout.sub1, container, attachToRoot: true);
28             }
29         });
30     }
31 }
32

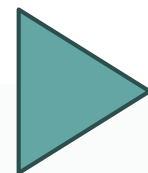
```



setContentView()메서드로는 부분 화면(부분 레이아웃)을 메모리에 객체화할 수 없음
 부분화면을 메모리에 객체화 하려면 인플레이터를 사용
 시스템 서비스로 **LayoutInflater** 라는 클래스 제공 -> 시스템 클래스는
getSystemService() 메서드 이용하여 **LayoutInflater** 객체를 참조한 후 사용.

AndroidManifest.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.gs19200csmlogin">
4
5     <application
6         android:allowBackup="true"
7         android:icon="@mipmap/ic_launcher"
8         android:label="Gs19200csmlogin"
9         android:roundIcon="@mipmap/ic_launcher_round"
10        android:supportsRtl="true"
11        android:theme="@style/AppTheme">
12        <activity android:name=".MainActivity"></activity>
13        <activity android:name=".menuActivity">
14            <intent-filter>
15                <action android:name="android.intent.action.MAIN" />
16
17                <category android:name="android.intent.category.LAUNCHER" />
18            </intent-filter>
19        </activity>
20    </application>
21
22 </manifest>
```



에뮬레이터 실행-동작 확인

여러 화면 간에 전환

- ▶ 앱은 여러 화면-> 화면을 전환하며 실행 -> 화면은 **Activity**로 구현
즉, 화면을 필요에 따라 띄우거나 닫는 과정은 액티비티를 전환하는 것
- ◆ 안드로이드의 앱을 구성하는 4가지 기본 구성 요소
- ◆ 앱을 단말에 설치 했을 때 안드로이드 시스템이 이 요소에 대한 정보를 요구
- ◆ 이 정보들은 어디에? **Manifest.xml**
- ◆ 새로운 **Activity**를 만들어 추가하면 **Manifest**에 정보 추가



Activity

Activity는 화면에 표시되는 **UI** 구성을 위해 가장 기본이 되는 요소.

안드로이드 앱은 화면에 **UI**를 표시하기 위해 최소 하나의 **Activity**를 가져야 하고, 앱 실행 시 지정된 **Activity**를 실행하여 사용자에게 **UI**를 표시.

예제에서 작성한 "**MainActivity**"가 바로 앱 실행 시 최초로 보여지는 **Activity**.

안드로이드 스튜디오에서 자동으로 생성되는 템플릿 코드에서 **MainActivity**가 최초 실행되도록 지정되어 있음.

(반드시 **MainActivity**라는 이름이어야 하는 것은 아닙니다. 프로젝트 생성할 때 입력하는 설정에 의한 것으로 원하는 이름으로 지정할 수 있습니다.)

Activity 클래스의 계층 구조

<https://developer.android.com/reference/android/app/Activity>

Activity

public class Activity

extends ContextThemeWrapper implements LayoutInflater.Factory2, Window.Callback, KeyEvent.Callback, View.OnCreateContextMenuListener, ComponentCallbacks2

java.lang.Object

↳ android.content.Context

↳ android.content.ContextWrapper

↳ android.view.ContextThemeWrapper

↳ android.app.Activity

✓ Known direct subclasses

AccountAuthenticatorActivity, ActivityGroup, AliasActivity, ExpandableListActivity, ListActivity, NativeActivity

✓ Known indirect subclasses

LauncherActivity, PreferenceActivity, TabActivity

Context

- 이름 그대로 해석한다면 애플리케이션(객체)의 현재 상태의 맥락(context)를 의미합니다. 컨텍스트는 새로 **생성된 객체가 지금 어떤 일이 일어나고 있는지 알 수 있도록 합니다.** 따라서 액티비티와 애플리케이션에 대한 정보를 얻기 위해서는 컨텍스트를 사용하면 됩니다.
- 또한, 컨텍스트(Context)는 시스템의 핸들과도 같습니다. 리소스, 데이터베이스, preferences 등에 대한 접근을 제공합니다. 안드로이드 앱에는 ‘액티비티’라는 것이 있습니다. 액티비티는 애플리케이션이 현재 실행중인 환경에 대한 핸들과도 같습니다. 액티비티 객체는 컨텍스트 객체를 상속받습니다. 액티비티는 애플리케이션의 특정 리소스와 클래스, 그리고 애플리케이션 환경에 대한 정보에 대해 접근할 수 있게 해줍니다.
- 안드로이드 개발에서 컨텍스트(Context)는 어디에나 있고, 가장 중요한 것입니다. 따라서 컨텍스트를 이해하고 올바르게 써야만합니다.

액티비티 컨텍스트(Activity Context)

- ▶ 액티비티 컨텍스트는 액티비티에서 사용 가능하며 이 컨텍스트는 액티비티의 라이프사이클과 연결되어 있습니다. 액티비티의 범위 내에서 컨텍스트를 전달하거나, 라이프사이클이 현재의 컨텍스트에 붙은 컨텍스트가 필요할 때(**need the context whose lifecycle is attached to the current context**) 액티비티 컨텍스트를 사용합니다.
- ▶ 예시: 라이프사이클이 액티비티에 붙은 객체를 생성해야 할 때 액티비티 컨텍스트를 사용할 수 있습니다.
- ▶ **ContentProvider**에서의 **getContext()**
- ▶ 이 컨텍스트는 액티비티 컨텍스트이며 애플리케이션 컨텍스트와 비슷하게 쓰일 수 있습니다. 이는 **getContext()** 메소드로 접근할 수 있습니다.

애플리케이션 컨텍스트(Application Context)

애플리케이션 컨텍스트는 싱글톤 인스턴스이며 액티비티에서 `getApplicationContext()`를 통해 접근할 수 있습니다. 이 컨텍스트는 애플리케이션의 라이프사이클과 연결되어 있습니다.

애플리케이션 컨텍스트는 현재의 컨텍스트와 분리된 라이프사이클을 가진 컨텍스트가 필요할 때나 액티비티의 범위를 넘어서 컨텍스트를 전달할 때에 사용합니다.

예시: 만약 당신의 애플리케이션에서 싱글톤 객체를 생성하였는데 그 객체에 컨텍스트가 필요하다면, 애플리케이션 컨텍스트를 사용하면 됩니다.

만약 이러한 상황에서 액티비티 컨텍스트를 전달한다면 메모리 누수가 발생할 것입니다. 액티비티는 가비지 콜렉터에 의해 수집되지 않는데 액티비티 컨텍스트는 액티비티에 대한 참조를 계속 유지하기 때문입니다.

액티비티에서 라이브러리를 초기화해야 하는 경우, 액티비티 컨텍스트가 아닌 애플리케이션 컨텍스트를 전달해야 합니다.

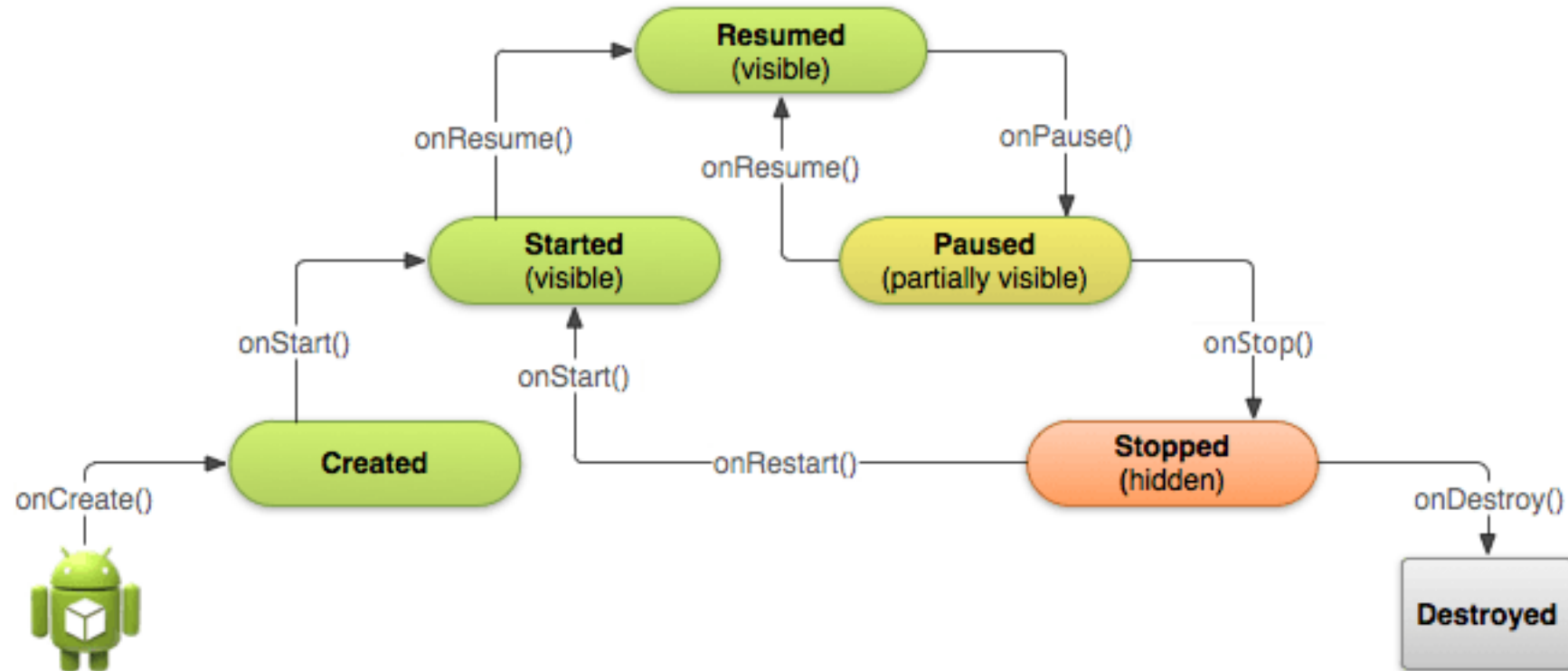
그 어떤 컨텍스트(Context)보다 오래 유지되는 컨텍스트(Context)가 필요할때에만 `getApplicationContext()`를 사용하십시오.

언제 `getApplicationContext()`를 쓰지 말아야 할까?

- 액티비티(**Activity**)가 하는 일 모두를 어플리케이션 컨텍스트(**Context**)가 완전히 지원하는 것은 아닙니다. 컨텍스트(**Context**)를 사용하여 작업하려고 하는 많은 것들이 안될 것이며 특히 **GUI**와 관련된 것은 실패할 것입니다.
- `getApplicationContext()`의 컨텍스트(**Context**)가 정리되지 않은 호출로 생성된 무언가를 유지하고 있으면 메모리 누수가 발생할 수 있습니다. 액티비티(**Activity**) 객체는 가비지 콜렉터에 의해 관리되지만, 어플리케이션(**Application**) 객체는 프로세스 수명 동안 유지됩니다.

Activity Lifecycle and state

- Activity는 생성, 시작, 실행, 일시정지, resume, 중지, 소멸의 생명주기를 가진다.



Activity

- ▶ 소스에서 띄울 때 **startActivity()** 사용. 단순히 Activity를 띄워 화면에 보이게 하는 것
- ▶ 하지만. 띄워야 할 화면이 많아지면 띄웠던 화면을 닫고 원래의 메인 화면으로 돌아 올

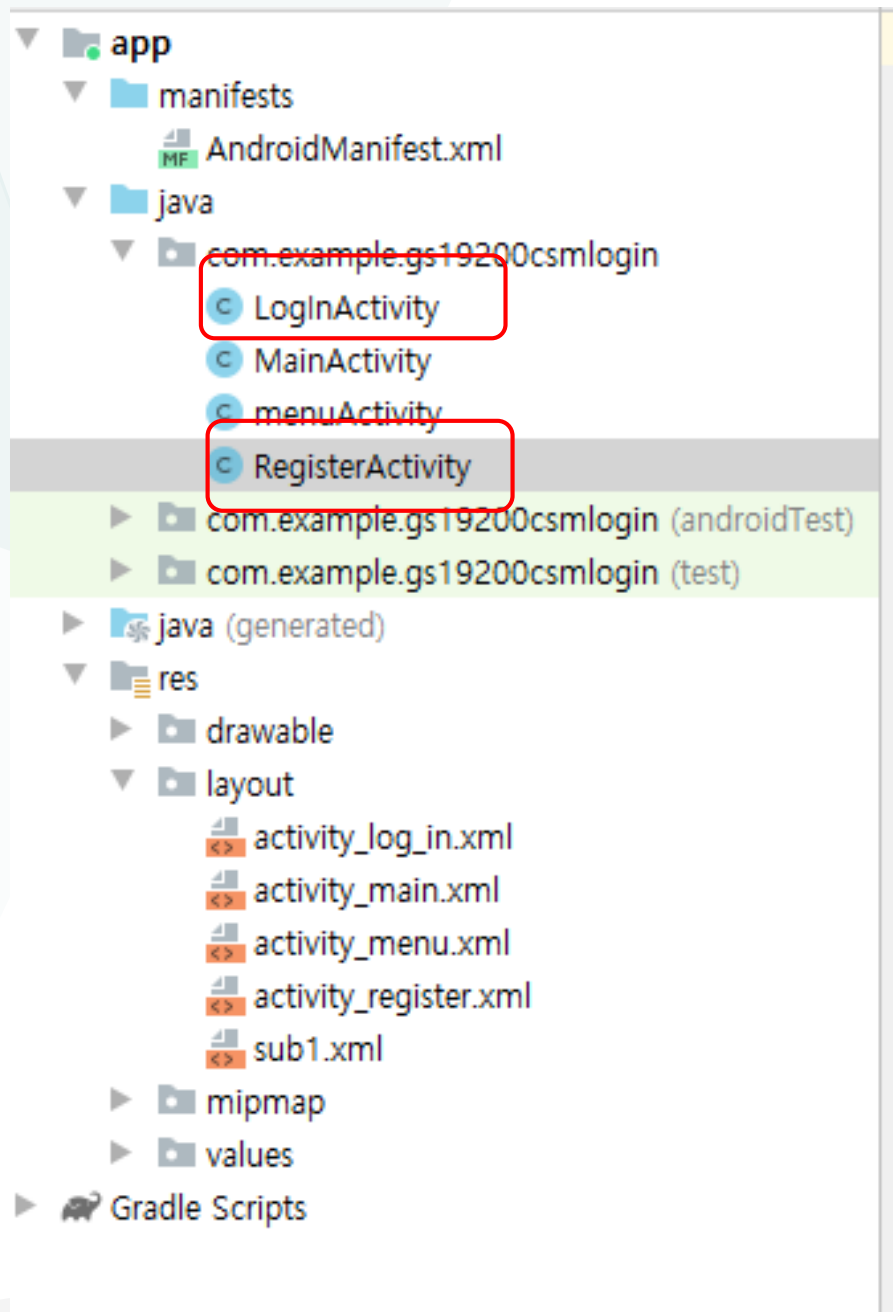
경우가 생기고 데이터를 넘겨 받아 새롭게 적용해야 하는 경우 생김.

=> 어떤 Activity①을 띄우고 부른 Activity②에서 다시 원래의 Activity①로 돌아올 때

응답을 받아 처리하는 코드가 필요

startActivityForResult() (인텐트 or 정수 코드값) 사용했으나

=> Activity Result API 사용권장 : **registerForActivityResult**



Activity 2개 추가

activity_menu.xml x sub1.xml x menuActivity.java x AndroidManifest.xml x activity_log_in.xml x LoginActivity.java x activity_register.xml x RegisterActivity.java x

Code Split Design

Palette

Common
Text
Buttons
Widgets
Layouts
Containers
Google
Legacy

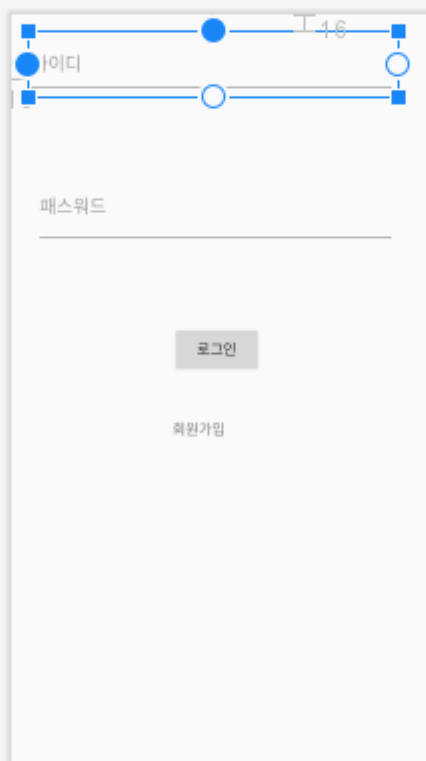
Ab TextView
Ab Plain Text
Ab Password
Ab Password (Num...
Ab E-mail
Ab Phone
Ab Postal Address
Ab Multiline Text
Ab Time
Ab Date
Ab Number

Component Tree

ConstraintLayout
Ab idText (Plain Text)
Ab passwordText (Password)
loginButton "로그인"
Ab registerButton "회원가입"

Plain Text
Password
Button
TextView

각각의 아이디



plain Text, Password는
text는 지우고
hint 속성에 글자 넣기
나머지는 그림과 동일

Attributes

Ab idText EditText

id idText

Declared Attributes + -

layout_width 362dp

layout_height 64dp

layout_constraint... parent

layout_constraint... parent

layout_marginLeft 16dp

layout_marginTo... 16dp

ems 10

hint 아이디

id idText

inputType textPersonName

Layout

Common Attributes

All Attributes

alpha

autoLink

autoText

background

activity_menu.xml x sub1.xml x menuActivity.java x AndroidManifest.xml x activity_log_in.xml x LoginActivity.java x **activity_register.xml** x Regist x

Code Split Design

Palette

- Common
 - Button
- Text
 - ImageButton
- Buttons
 - ChipGroup
 - Chip
- Widgets
 - CheckBox
- Layouts
 - RadioGroup
- Containers
 - RadioButton
- Google
 - ToggleButton
- Legacy
 - Switch
 - FloatingActionB...

Component Tree

- ConstraintLayout
 - idText (Plain Text)**
 - passwordText (Password)
 - nameText (Plain Text)
 - ageText (Plain Text)
 - registerButton "회원가입"

Attributes

idText EditText

id idText

Declared Attributes

layout_width	wrap_content
layout_height	wrap_content
layout_constraint...	parent
layout_constraint...	parent
layout_marginLeft	16dp
layout_marginTo...	72dp
ems	10
hint	아이디
id	idText
inputType	textPersonName

Layout

Common Attributes

All Attributes

alpha	
autoLink	
autoText	
background	@android:drawable/e

plain Text, Password는 text는 지우고 hint 속성에 글자 넣기 나머지는 그림과 동일

아이디

패스워드

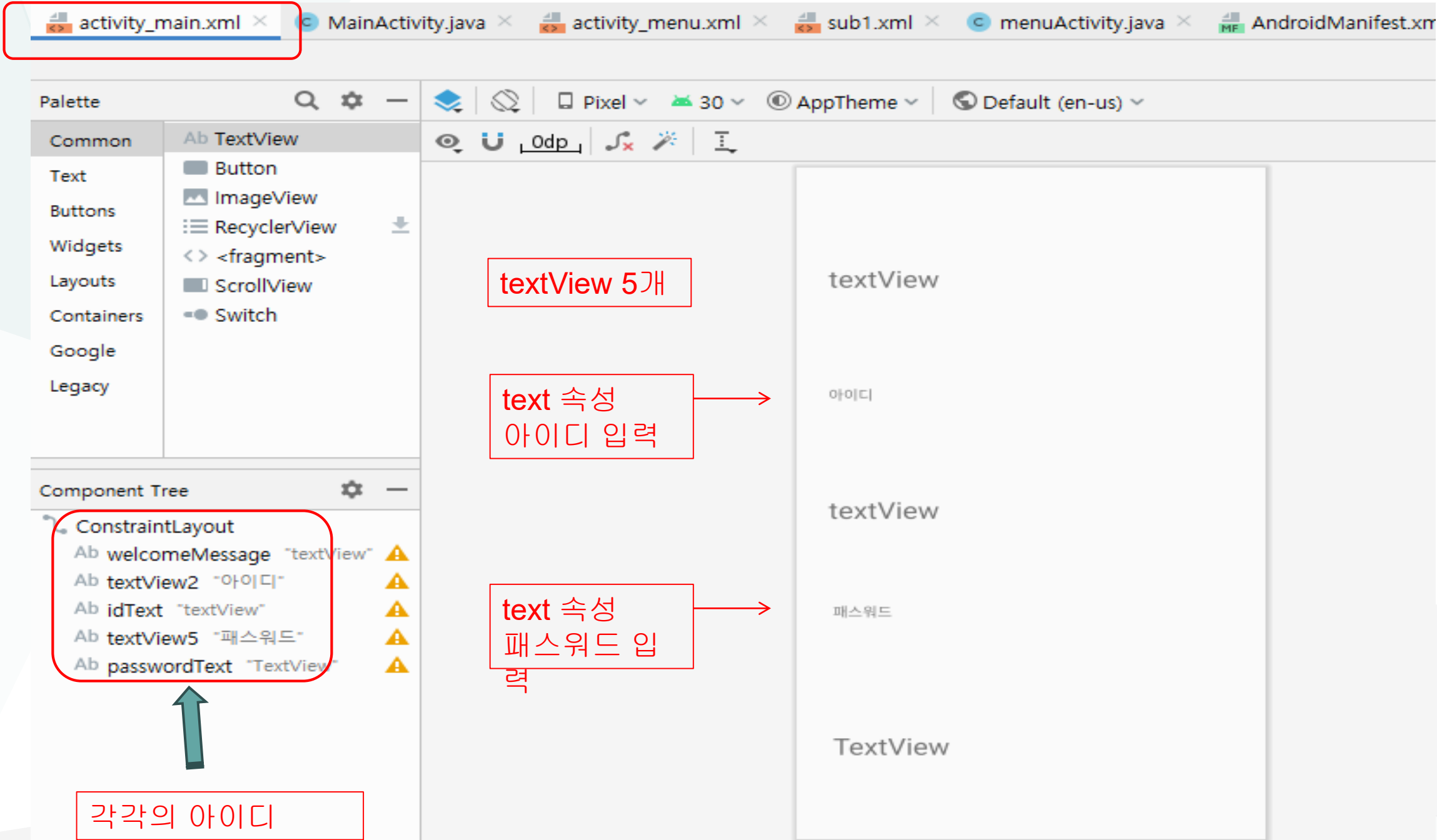
이름

나이

회원가입

Button

각각의 아이디



```

1  package com.example.gs19200csmlogin;
2
3  import ...
4
5
6
7
8
9  public class RegisterActivity extends AppCompatActivity {
10
11      @Override
12      protected void onCreate(Bundle savedInstanceState) {
13          super.onCreate(savedInstanceState);
14          setContentView(R.layout.activity_register);
15
16          EditText idText = (EditText)findViewById(R.id.idText);
17          EditText passwordText = (EditText)findViewById(R.id.passwordText);
18          EditText nameText = (EditText)findViewById(R.id.nameText);
19          EditText ageText = (EditText)findViewById(R.id.ageText);
20
21          Button registerButton =(Button)findViewById(R.id.registerButton);
22
23      }
24  }

```

RegisterActivity.java 코드 작성

LogInActivity.java 코드 작성

```
2
3 import ...
11
12 public class LogInActivity extends AppCompatActivity {
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_log_in);
18
19         EditText idText = (EditText)findViewById(R.id.idText);
20         EditText passwordText = (EditText)findViewById(R.id.passwordText);
21
22         Button loginButton = (Button)findViewById(R.id.LoginButton);
23         TextView registerButton = (TextView) findViewById(R.id.registerButton);
24
25         registerButton.setOnClickListener(new View.OnClickListener() {
26             @Override
27             public void onClick(View view) {
28                 Intent registerIntent = new Intent( packageContext: LogInActivity.this, RegisterActivity.class);
29                 LogInActivity.this.startActivity(registerIntent);
30             }
31         });
32     }
33 }
```

각각의 아이디



```

1 package com.example.gs19200csmlogin;
2
3 import ...
4
5
6
7
8
9 public class MainActivity extends AppCompatActivity {
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_main);
15
16
17         EditText idText = (EditText)findViewById(R.id.idText);
18         EditText passwordText = (EditText)findViewById(R.id.passwordText);
19         TextView welcomeMessage = (TextView)findViewById(R.id.welcomeMessage);
20     }
21 }

```

MainActivity.java 코드 작성

textView

아이디

textView

패스워드

TextView

```

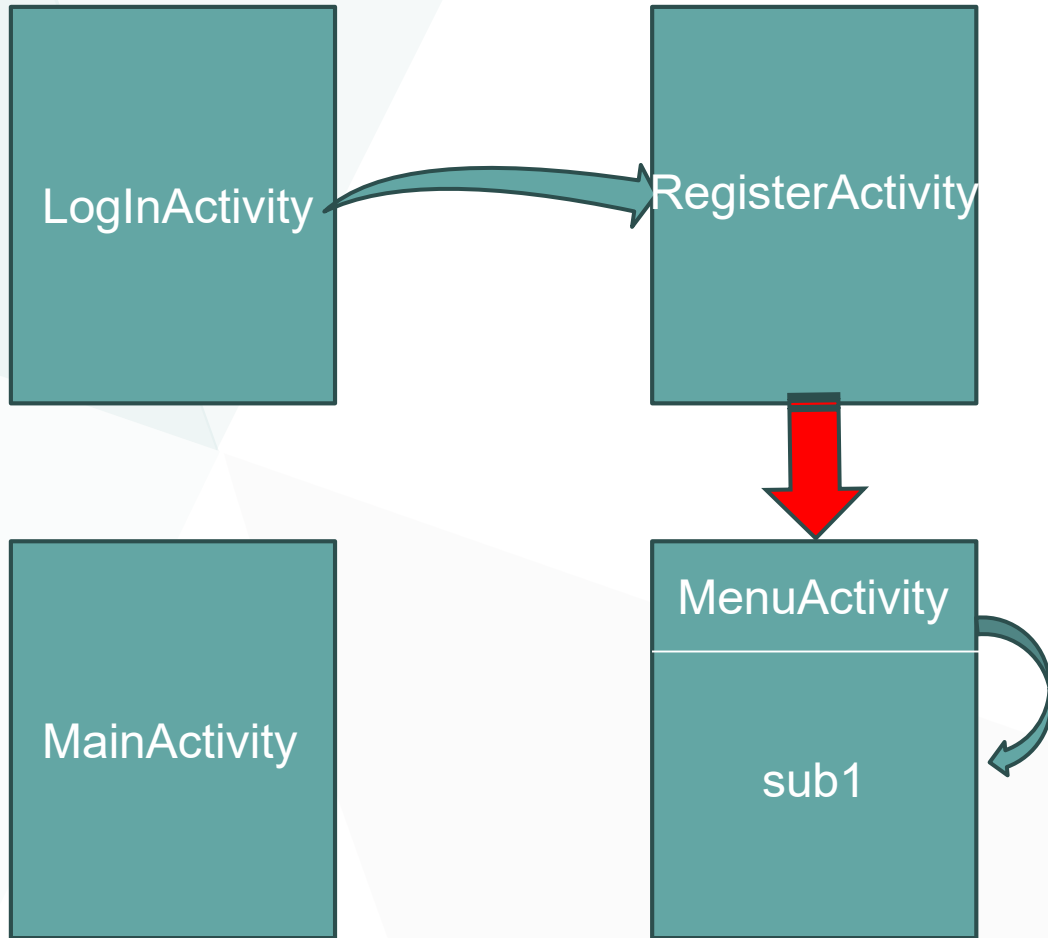
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="com.example.gs19200csmlogin">
4
5      <application
6          android:allowBackup="true"
7          android:icon="@mipmap/ic_launcher"
8          android:label="@string/app_name"
9          android:roundIcon="@mipmap/ic_launcher_round"
10         android:supportsRtl="true"
11         android:theme="@style/AppTheme">
12         <activity android:name=".RegisterActivity"></activity>
13         <activity android:name=".menuActivity" />
14         <activity android:name=".MainActivity" />
15         <activity android:name=".LogInActivity">
16             <intent-filter>
17                 <action android:name="android.intent.action.MAIN" />
18
19                 <category android:name="android.intent.category.LAUNCHER" />
20             </intent-filter>
21         </activity>
22     </application>
23
24 </manifest>

```

AndroidManifest.xml 코드 작성

에뮬레이터 실행-동작확인

에뮬레이터 실행 동영상 녹화 파일저장은 20XXX아무개실습



RegisterActivity 에 버튼을 추가하여
클릭하면 MenuActivity로 화면이 넘어가도록
기능 추가하기
세부 조건은 재량

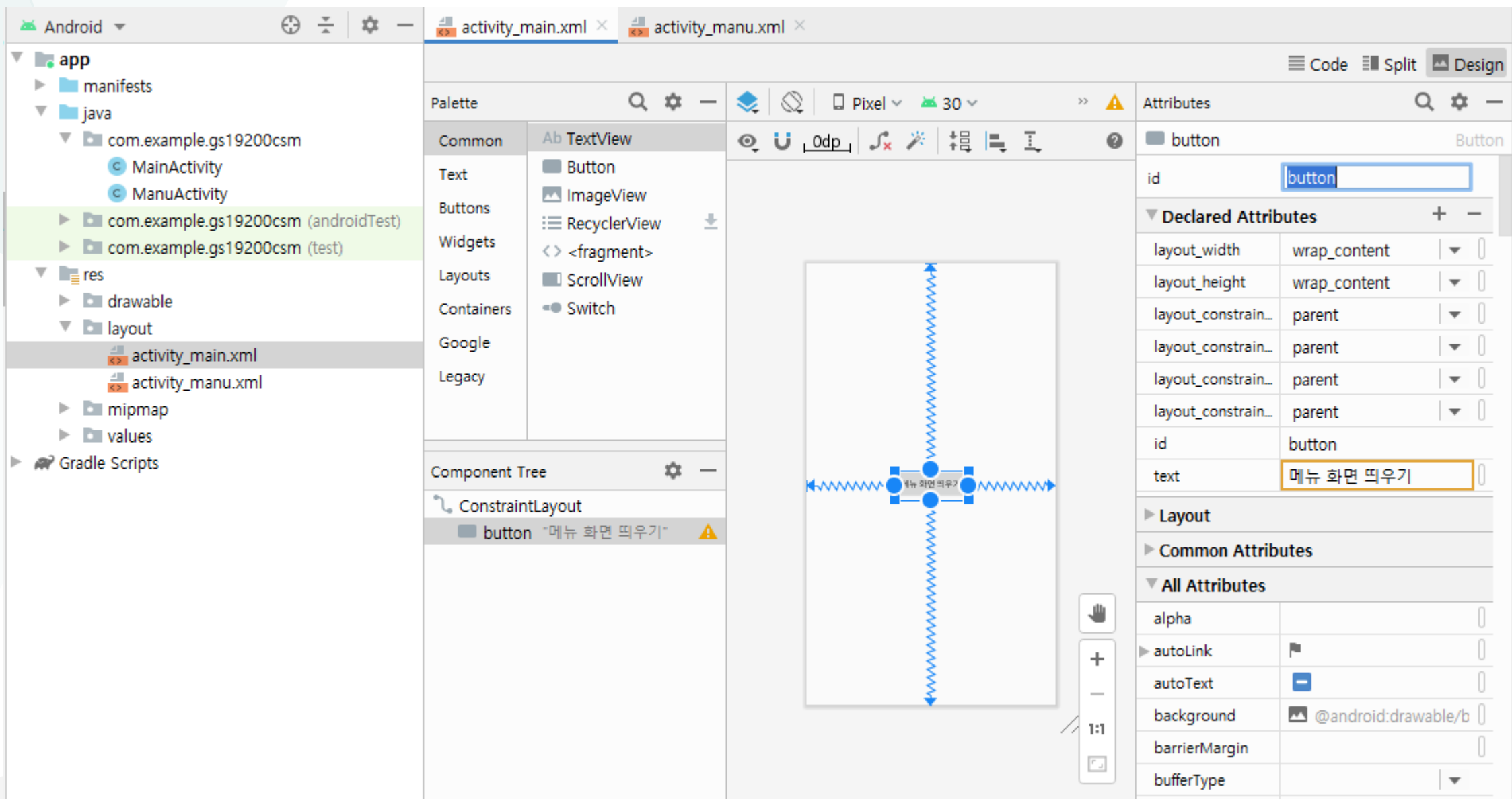
LoginActivity->RegisterActivity
->MenuActivity->sub1 의 순서로
화면이 전환되는 과정을 녹화

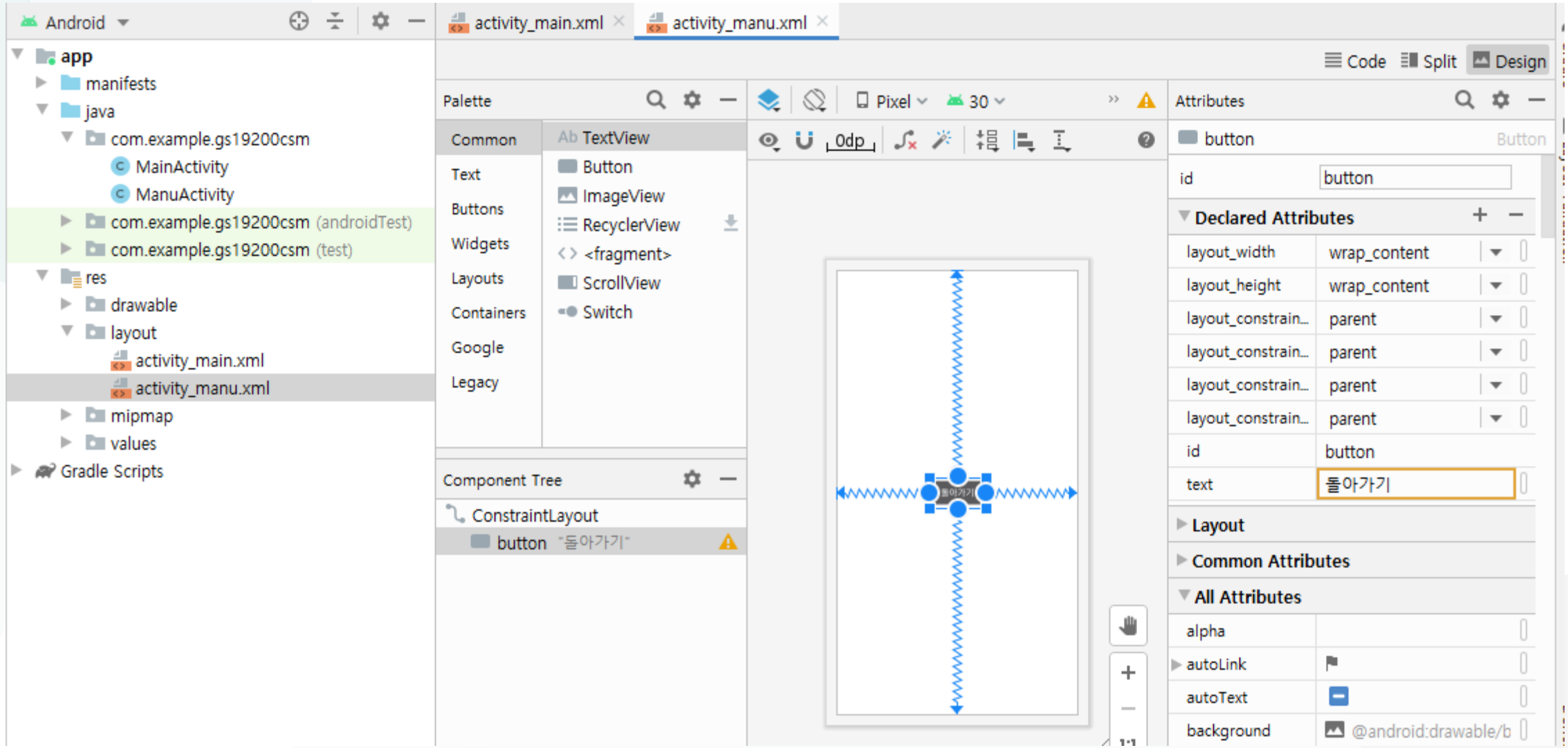
Intent

- Intent는 안드로이드 시스템을 통해 다른 앱의 구성요소나 액티비티의 작업 (action)을 요청하는데 사용되는 객체
- Intent로 할 수 있는 일
 1. start activity
 2. start service
 3. deliver Broadcast

Intent - data 전달

새 프로젝트- Gs20200csm-Activity추가-화면과 같이 구성





```

1 package com.example.gs19200csm;
2
3 import ...
4
11
12 public class MainActivity extends AppCompatActivity {
13
14     public static final int REQUEST_CODE_MENU = 101;
15
16
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_main);
21
22         Button button = findViewById(R.id.button);
23         button.setOnClickListener(new View.OnClickListener(){
24             @Override
25             public void onClick(View v){
26                 Intent intent = new Intent(getApplicationContext(), ManuActivity.class);
27                 startActivityForResult(intent, REQUEST_CODE_MENU);
28             }
29         });
30     }
31
32     @Override
33     protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
34         super.onActivityResult(requestCode, resultCode, data);
35

```

```

35
36     if(requestCode == REQUEST_CODE_MENU) {
37         Toast.makeText(getApplicationContext(),
38             text: "onActivityResult 메서드 호출됨. 요청 코드 : " + requestCode +
39             ", 과 코드 : " + resultCode, Toast.LENGTH_LONG).show();
40
41         if(resultCode == RESULT_OK) {
42             String name = data.getStringExtra(name: "name");
43             Toast.makeText(getApplicationContext(), text: "응답으로 전달된 name : " + name,
44                 Toast.LENGTH_LONG).show();
45         }
46     }
47 }
48
49

```

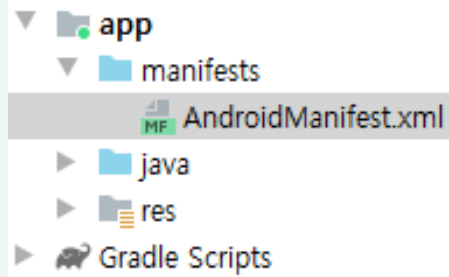
Activity Result API 로 구현 - 참고

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Button button = findViewById(R.id.menubutton);

    ActivityResultLauncher<Intent> mainActivityResultLauncher = registerForActivityResult(
        new ActivityResultContracts.StartActivityForResult(),
        new ActivityResultCallback<ActivityResult>() {
            public void onActivityResult(ActivityResult result) {
                if(result.getResultCode() == Activity.RESULT_OK) {
                    Intent data = result.getData();
                    String name = data.getStringExtra("name");
                    Toast.makeText(getApplicationContext(), "response name: " + name,
                        Toast.LENGTH_LONG).show();
                }
            }
        });

    button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(getApplicationContext(), MenuActivity.class);
            mainActivityResultLauncher.launch(intent);
        }
    });
}
```

```
activity_main.xml x activity_manu.xml x MainActivity.java x ManuActivity.java x
1 package com.example.gs19200csm;
2
3 import ...
9
10 public class ManuActivity extends AppCompatActivity {
11
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_manu);
16
17         Button button = findViewById(R.id.button);
18         button.setOnClickListener(new View.OnClickListener() {
19             public void onClick(View v) {
20                 Intent intent = new Intent();
21                 intent.putExtra(name: "name", value: "mike");
22                 setResult(RESULT_OK, intent);
23                 finish();
24             }
25         });
26     }
27 }
```



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.gs19200csm">
4
5     <application
6         android:allowBackup="true"
7         android:icon="@mipmap/ic_launcher"
8         android:label="Gs19200csm"
9         android:roundIcon="@mipmap/ic_launcher_round"
10        android:supportsRtl="true"
11        android:theme="@style/AppTheme">
12        <activity android:name=".MenuActivity"
13            android:label="메뉴 액티비티"
14            android:theme="@style/Theme.AppCompat.Dialog">
15        </activity>
16        <activity android:name=".MainActivity">
17            <intent-filter>
18                <action android:name="android.intent.action.MAIN" />
19
20                <category android:name="android.intent.category.LAUNCHER" />
21            </intent-filter>
22        </activity>
23    </application>
24
25 </manifest>
```

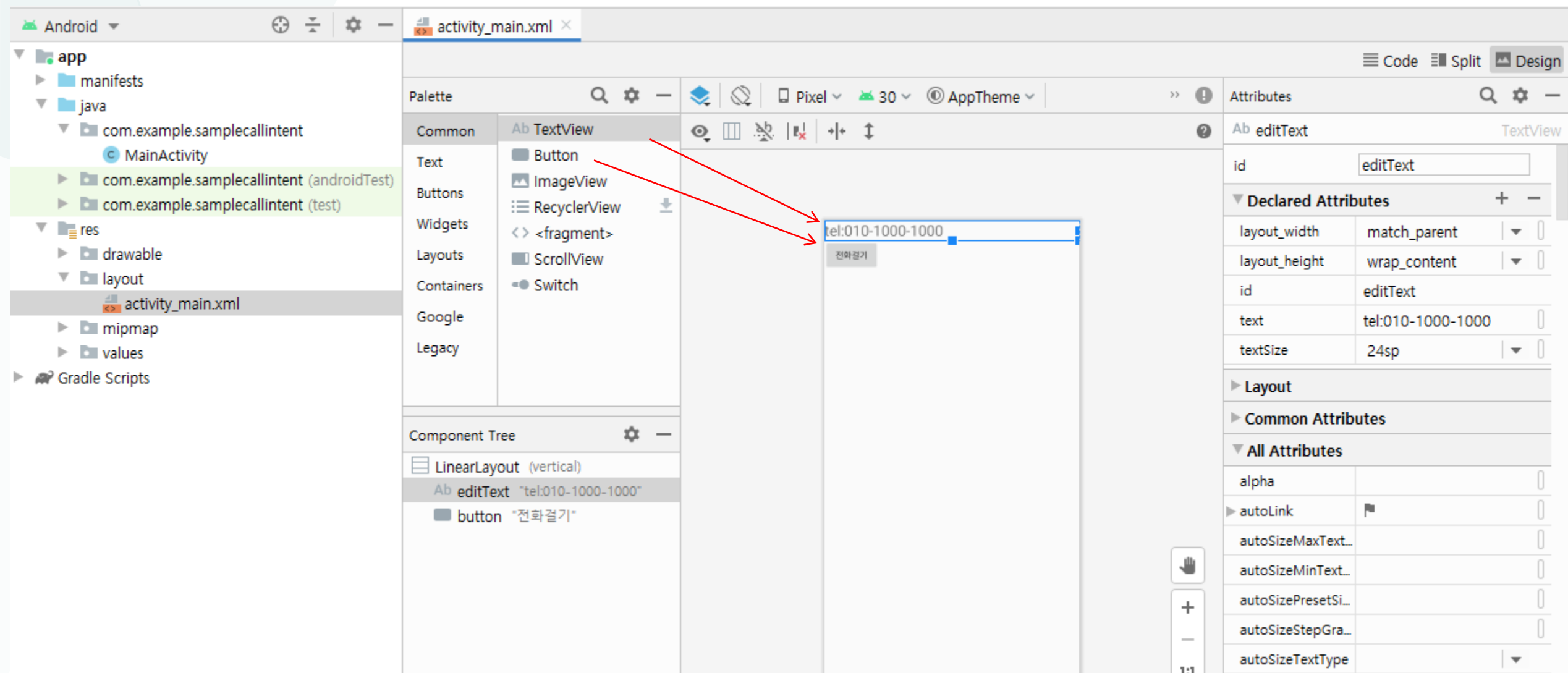
에뮬레이터 실행 동영상 녹화 파일저장은 20XXX아무개실습7



MainActivity->MenuActivity->MainActivity 의 순서로
화면이 전환되는 과정을 녹화

새 프로젝트- Gs20200csmCallIntent

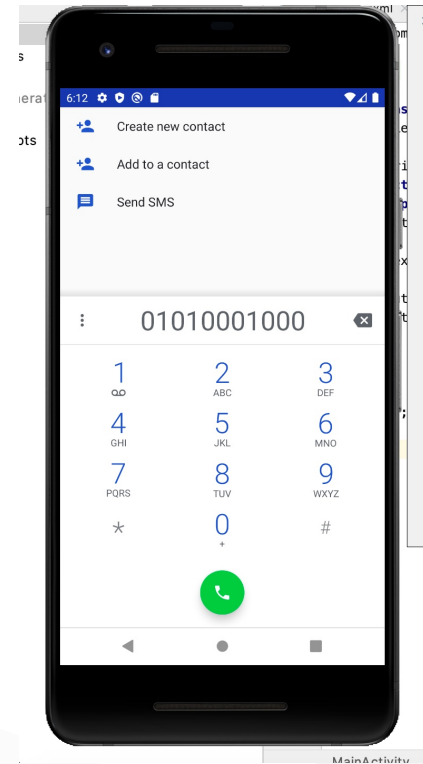
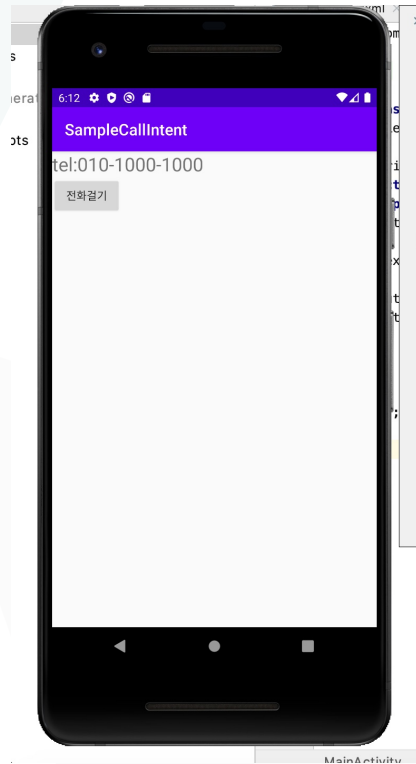
-Activity추가-화면과 같이 구성



activity_main.xml x MainActivity.java x

```
1 package com.example.samplecallintent;
2
3 import ...
4
12
13 public class MainActivity extends AppCompatActivity {
14     TextView textView;
15
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.activity_main);
20
21         textView = findViewById(R.id.editText);
22
23         Button button = findViewById(R.id.button);
24         button.setOnClickListener(new View.OnClickListener() {
25             @Override
26             public void onClick(View view) {
27                 String data = textView.getText().toString();
28
29                 Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(data));
30                 startActivity(intent);
31             }
32         });
33     }
34 }
```

에뮬레이터 실행 동영상 녹화 파일저장은 20XXX아무개실습8



화면이 전환되는 과정을 녹화



User Interface (Fragment, Tab, Viewpager)

Fragment / 상, 하 탭 / 뷰페이지

- ▶ 목표 : 필요한 기능을 합쳐라~!!
- ▶ 카카오를 실행시켜 아래 버튼을 눌러보세요.
- ▶ 화면을 옆으로 밀어 보세요. (BAND 앱)
- ▶ 앱을 실행 중에 메시지가 왔을 때는?

각각의 기능에 대해 실습할 예정입니다. 실제론 기능이 하나의 앱에서 작동할 것입니다.

우선 따라 하면서 코드를 이해하고 하나로 합쳐 보세요.

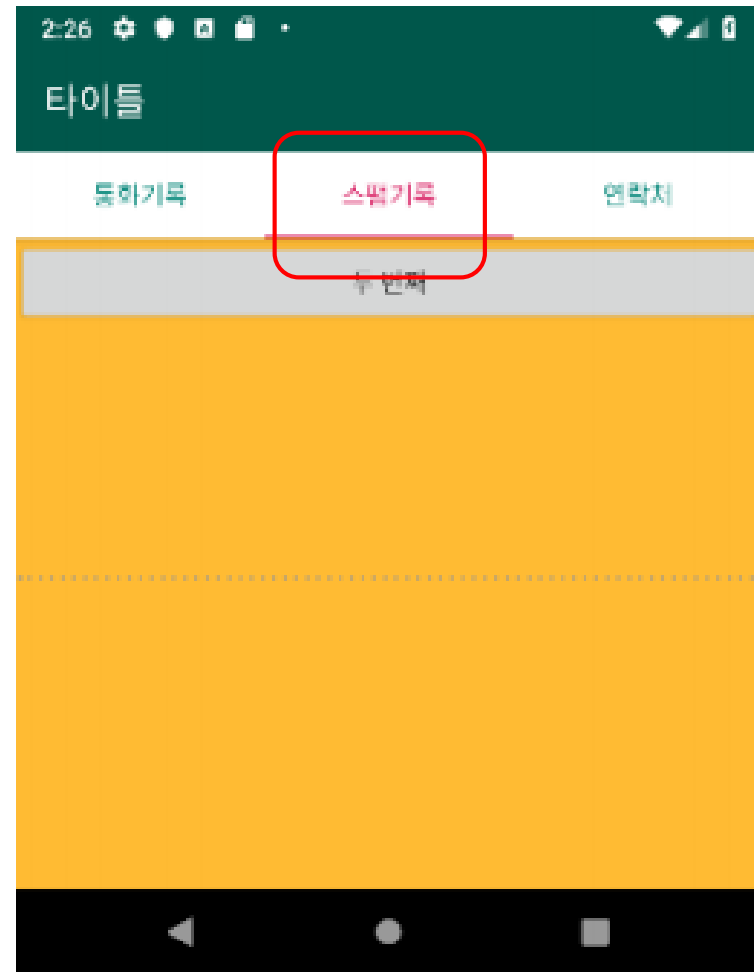
이내용은 실제로 스마트폰으로 실행시켜 볼 것을 추천합니다.

상위에 있는 탭보다 하위에 있는 탭이 요즘 많이 사용됩니다. 왜그럴까?

앱을 만들 때 직관적인 UI를 잘~구성하는 것이 50% 이상입니다.

어떤 모양을 기본으로 앱을 만들지 고민해 보세요.

실습 - ①



activity_main.xml
MainActivity.java

새 프로젝트 이름 : Gs20200csmTab
패키지명 : org.techtown.tab

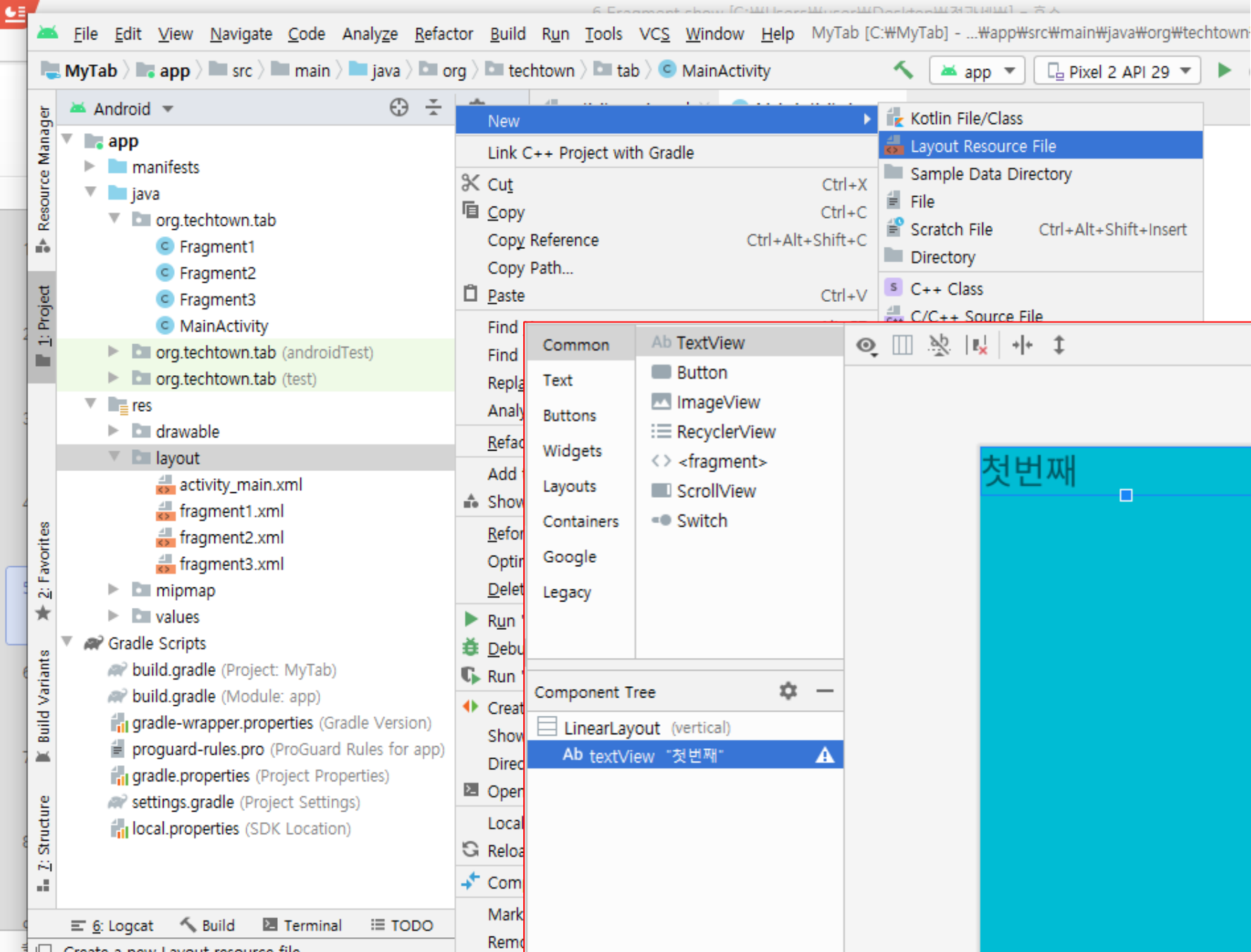
fragment1.xml
Fragment1.java

fragment2.xml
Fragment2.java

fragment3.xml
Fragment3.java

실습 순서

- ▶ fragment1.xml , fragment2.xml , fragment3.xml 추가
- ▶ Fragnnet1.java , Fragnnet2.java , Fragnnet3.java 추가
- ▶ Activity_main.xml 작업
- ▶ MainActivity.java 작업
- ▶ 에뮬레이터 결과 확인



fragment1.xml '첫번째' 배경색 자유
fragment2.xml '두번째' 배경색 자유
fragment3.xml '세번째' 배경색 자유

추가,

activity_main.xml x MainActivity.java x fragment1.xml x Fragment1.java x

세 번 반복

```
1 package org.techtown.tab;
2
3 import android.os.Bundle;
4 import android.view.LayoutInflater;
5 import android.view.View;
6 import android.view.ViewGroup;
7
8 import androidx.annotation.NonNull;
9 import androidx.annotation.Nullable;
10 import androidx.fragment.app.Fragment;
11
12 public class Fragment1 extends Fragment {
13
14     @Nullable
15     @Override
16     public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
17         return inflater.inflate(R.layout.fragment1, container, attachToRoot: false);
18     }
19 }
20
```

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">

    <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/TextAppearance.AppCompat.Widget.ActionBar.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.appbar.AppBarLayout>

        <FrameLayout
            android:id="@+id/container"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            app:layout_behavior="com.google.android.material.appbar.AppBarLayout$Scrolli...">

        </FrameLayout>

    </androidx.coordinatorlayout.widget.CoordinatorLayout>
```

타이틀

```
activity_main.xml x MainActivity.java x fragment1.xml x Fragment1.java x
1 package org.techtown.tab;
2
3 import android.os.Bundle;
4 import android.util.Log;
5 import androidx.appcompat.app.ActionBar;
6 import androidx.appcompat.app.AppCompatActivity;
7 import androidx.appcompat.widget.Toolbar;
8 import androidx.fragment.app.Fragment;
9 import com.google.android.material.tabs.TabLayout;
10 import static com.google.android.material.tabs.TabLayout.*;
11
12 public class MainActivity extends AppCompatActivity {
13     Fragment1 fragment1;
14     Fragment2 fragment2;
15     Fragment3 fragment3;
16
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_main);
21
22         fragment1 = new Fragment1();
23         fragment2 = new Fragment2();
24         fragment3 = new Fragment3();
25
26         getSupportFragmentManager().beginTransaction().replace(R.id.container, fragment1).commit();
27     }
}
```

```

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();

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

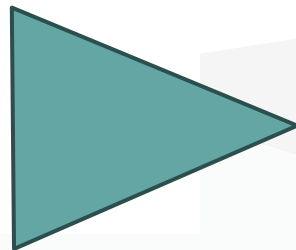
    @Override
    public void onTabUnselected(TabLayout.Tab tab){}

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

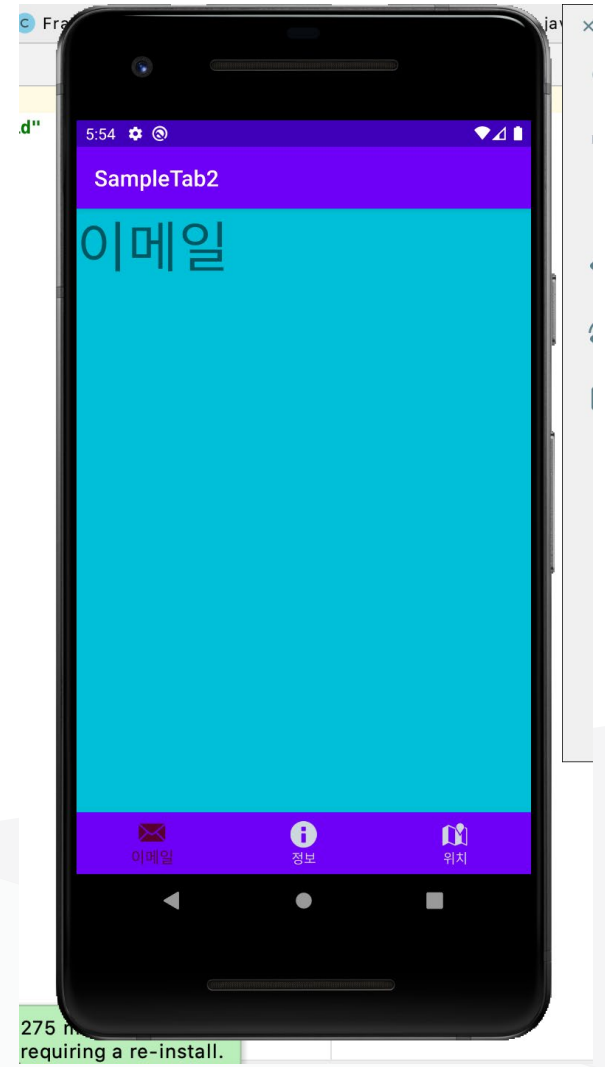
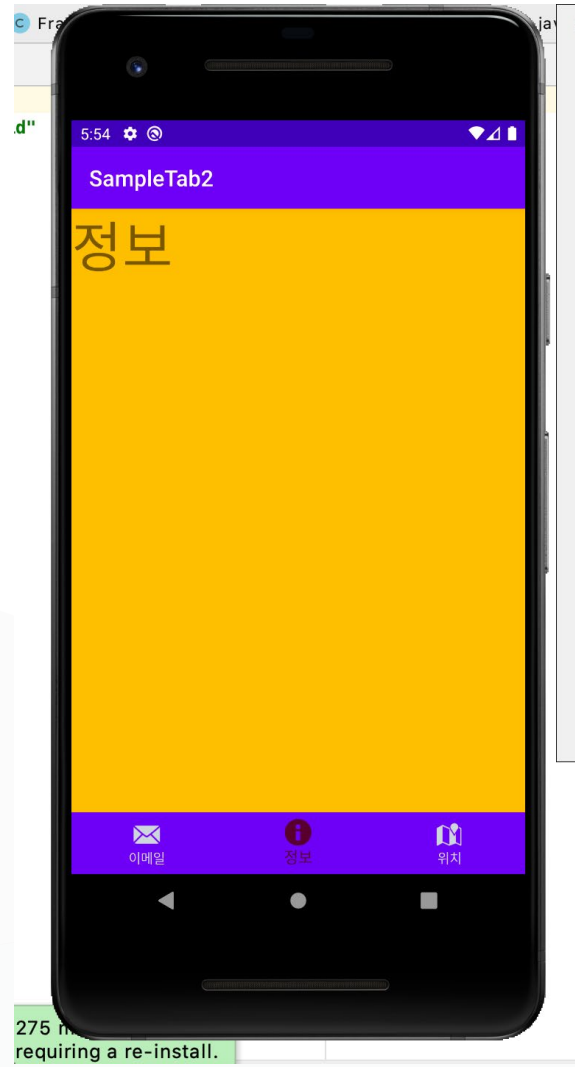
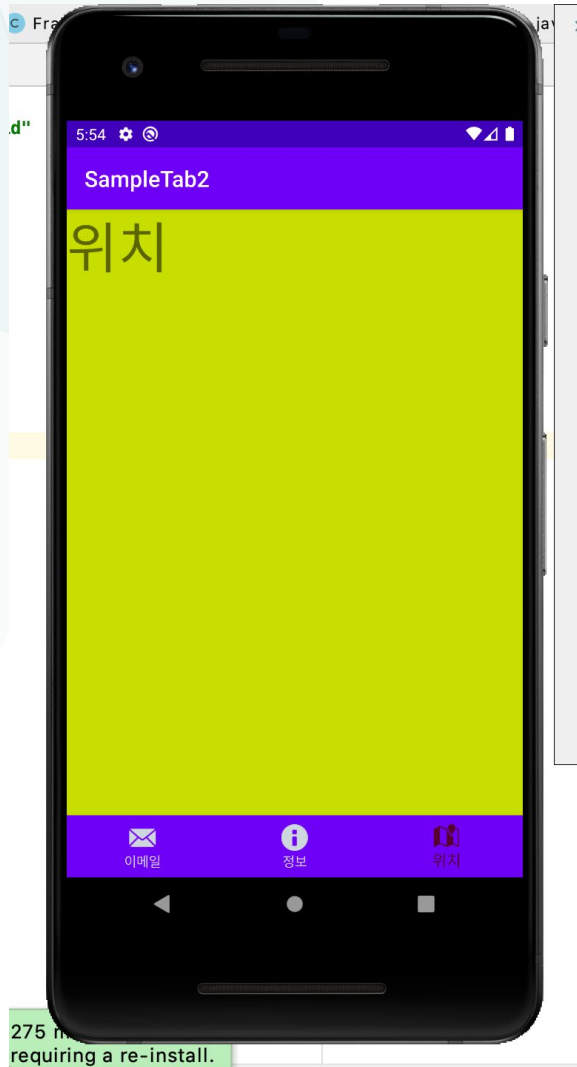
});
}

```

에뮬레이터 실행



실습 - ②



activity_main.xml
MainActivity.java

새 프로젝트 이름 : Gs20200csmTab2
패키지명 : org.techtown.tab2

fragment1.xml
Fragment1.java

fragment2.xml
Fragment2.java

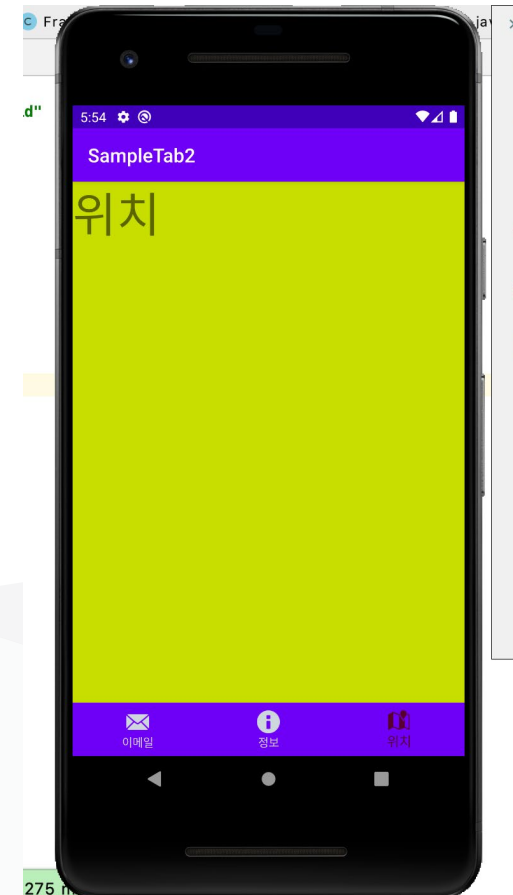
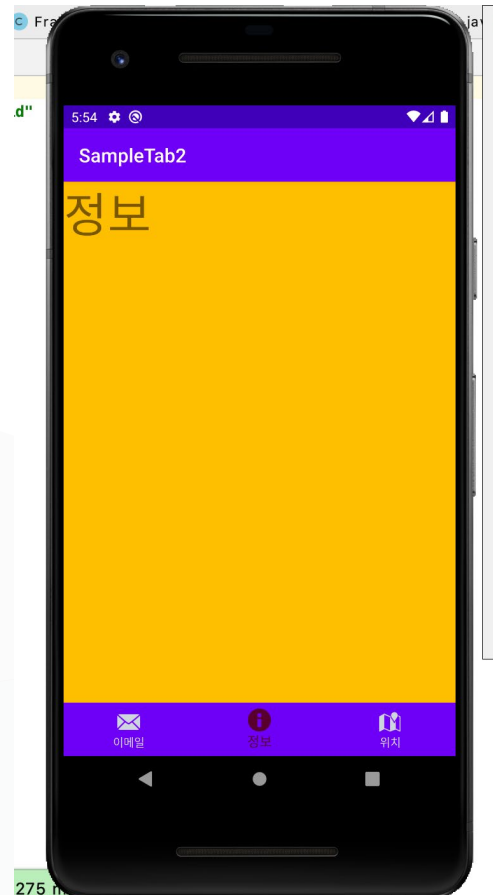
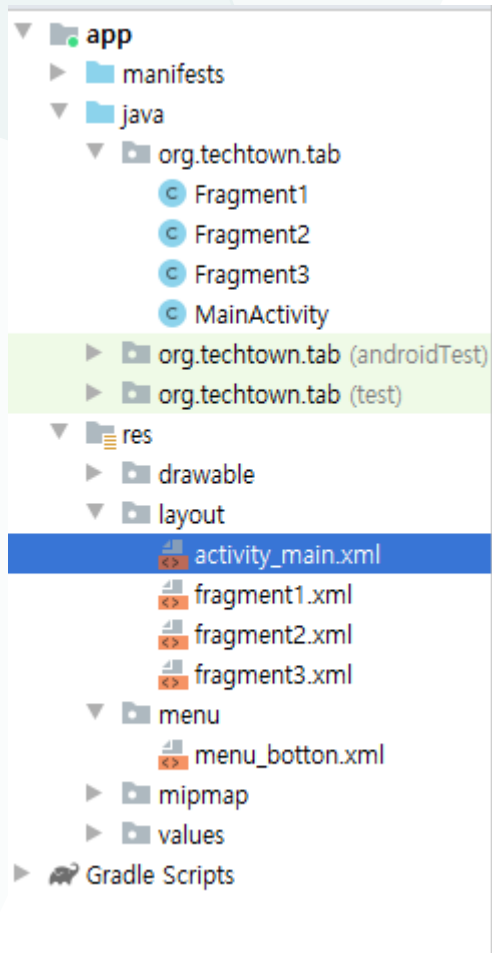
fragment3.xml
Fragment3.java

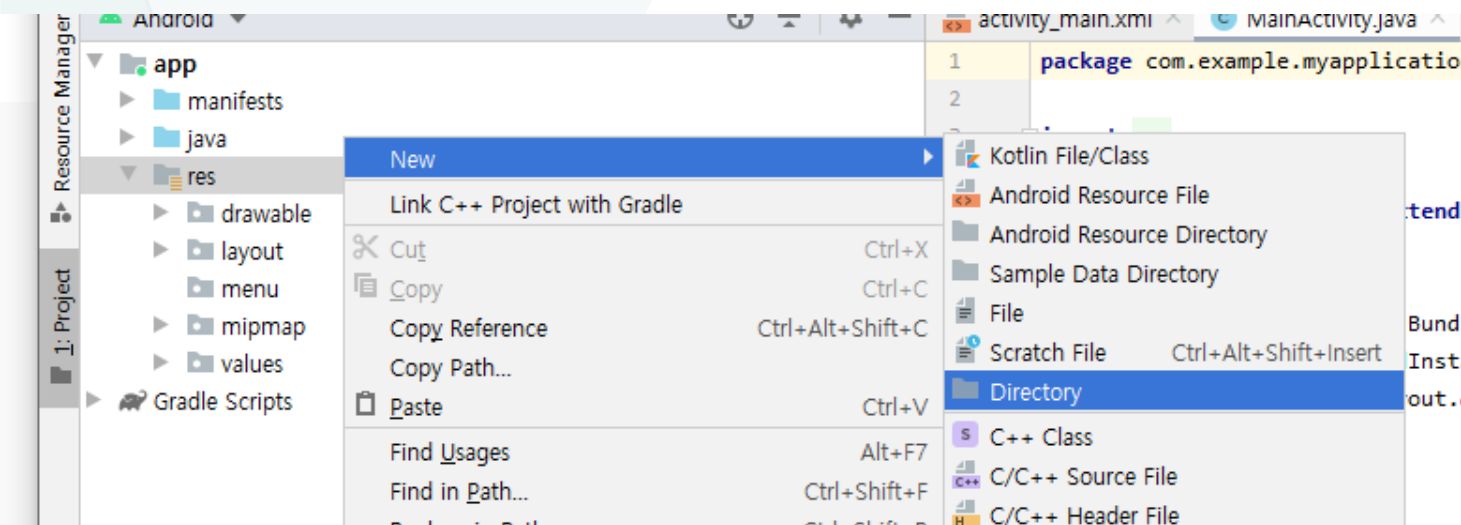
실습 순서

- ▶ fragment1.xml , fragment2.xml , fragment3.xml 추가
- ▶ Fragnnet1.java , Fragnnet2.java , Fragnnet3.java 추가
- ▶ app/res/menu 디렉토리 추가 - menu_bottom.xml 생성
- ▶ munu_botton.xml 작업
- ▶ Activity_main.xml 작업
- ▶ MainActivity.java 작업
- ▶ 에뮬레이터 결과 확인

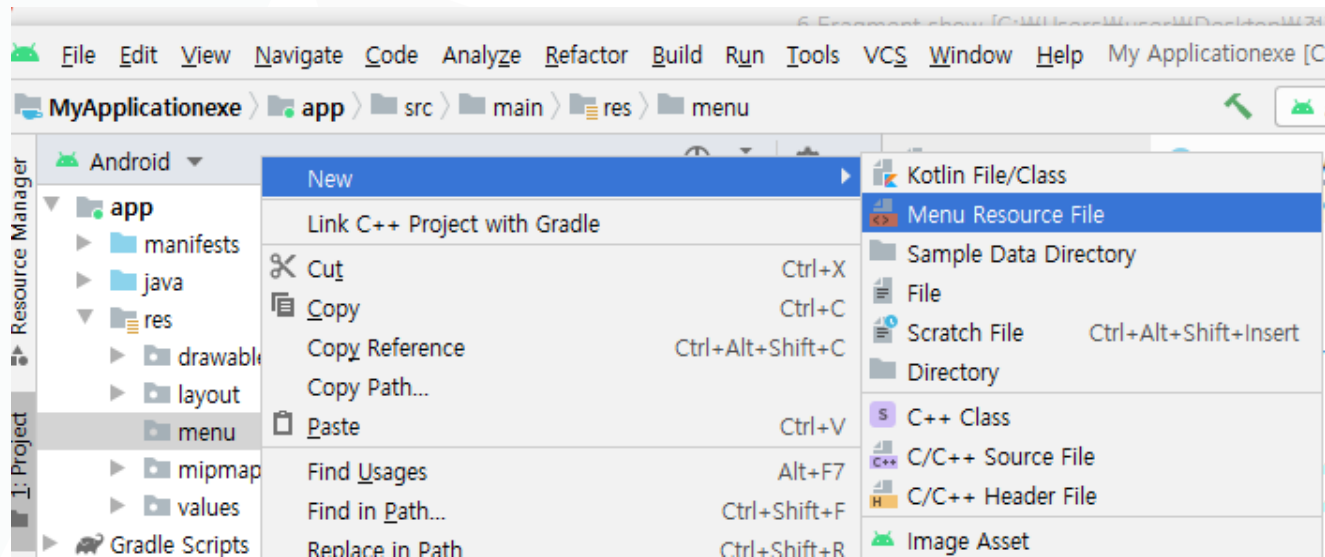
이전 프로젝트에서 fragment1,2,3.xml

Fragment1,2,3.java 복사 - 붙여넣기
fragment 1,2,3 의 textView
이메일, 정보, 위치로 수정

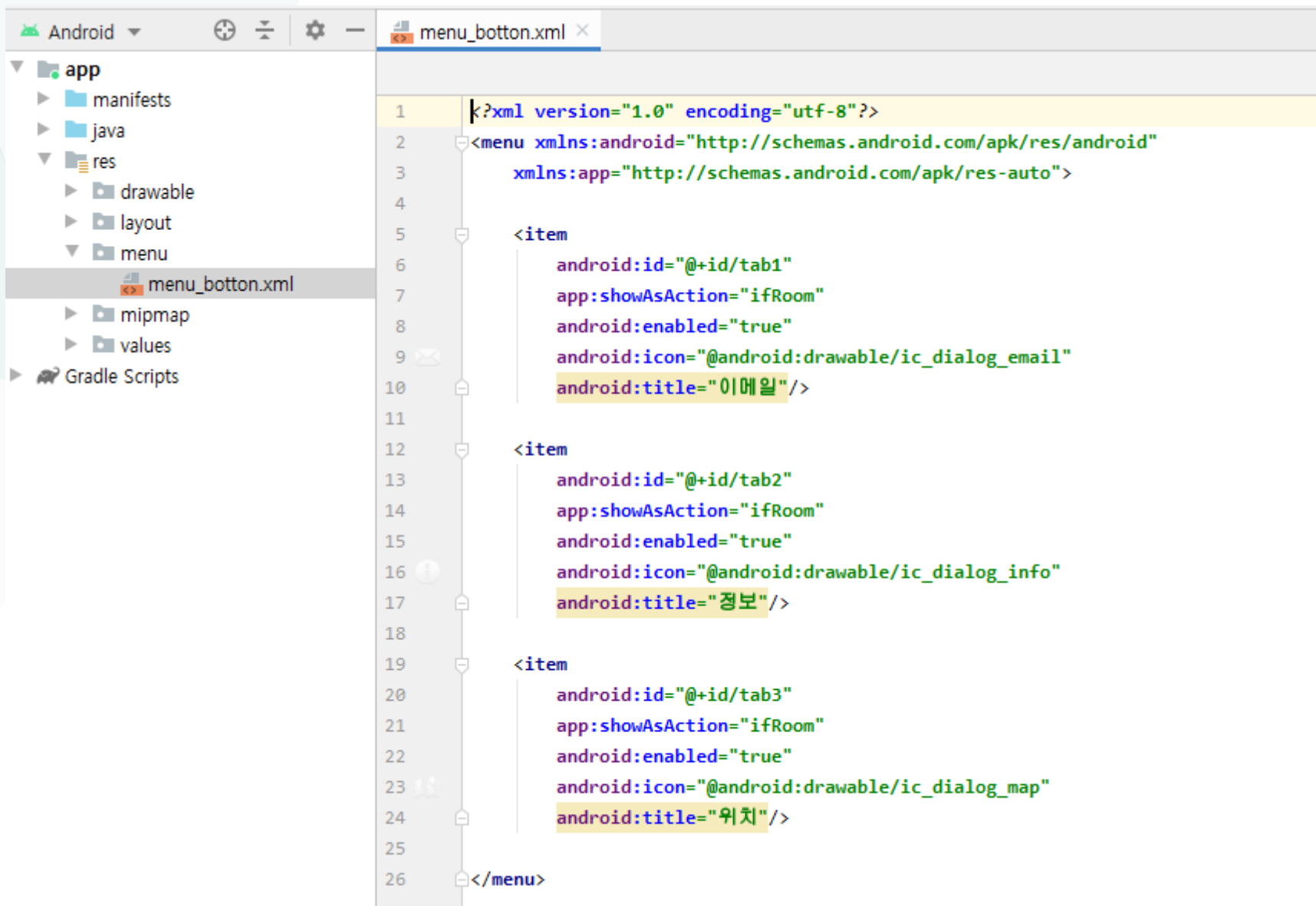




/app/res/아래
menu 디렉토리 생성



menu_bottom.xml 파일 생성



Android

app

manifests

java

org.techtown.tab

Fragment1

Fragment2

Fragment3

MainActivity

org.techtown.tab (androidTest)

org.techtown.tab (test)

res

drawable

layout

activity_main.xml

fragment1.xml

fragment2.xml

fragment3.xml

menu

menu_bottom.xml

mipmap

values

Gradle Scripts

menu_bottom.xml

activity_main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".MainActivity">
8     <FrameLayout
9         android:id="@+id/container"
10        android:layout_width="match_parent"
11        android:layout_height="match_parent"
12        app:layout_behavior="com.google.android.material.appbar.AppBarLayout$Scrolli..."/>
13
14    <com.google.android.material.bottomnavigation.BottomNavigationView
15        android:id="@+id/bottom_navigation"
16        android:layout_width="match_parent"
17        android:layout_height="wrap_content"
18        android:layout_marginEnd="0dp"
19        android:layout_marginStart="0dp"
20        app:layout_constraintBottom_toBottomOf="parent"
21        app:layout_constraintLeft_toLeftOf="parent"
22        app:layout_constraintRight_toRightOf="parent"
23        app:itemBackground="@color/colorPrimary"
24        app:itemIconTint="@drawable/item_color"
25        app:itemTextColor="@drawable/item_color"
26        app:menu="@menu/menu_bottom"/>
27 </androidx.constraintlayout.widget.ConstraintLayout>
```

Android

+

−

⚙

—

menu_bottom.xml

activity_main.xml

MainActivity.java

app

manifests

java

org.techtown.tab

Fragment1

Fragment2

Fragment3

MainActivity

org.techtown.tab (androidTest)

org.techtown.tab (test)

res

drawable

layout

activity_main.xml

fragment1.xml

fragment2.xml

fragment3.xml

menu

menu_bottom.xml

mipmap

values

Gradle Scripts

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

```
package org.techtown.tab;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.MenuItem;
import com.google.android.material.bottomnavigation.BottomNavigationView;

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();

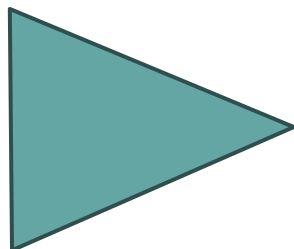
        getSupportFragmentManager().beginTransaction().replace(R.id.container, fragment1).commit();
    }
}
```

```

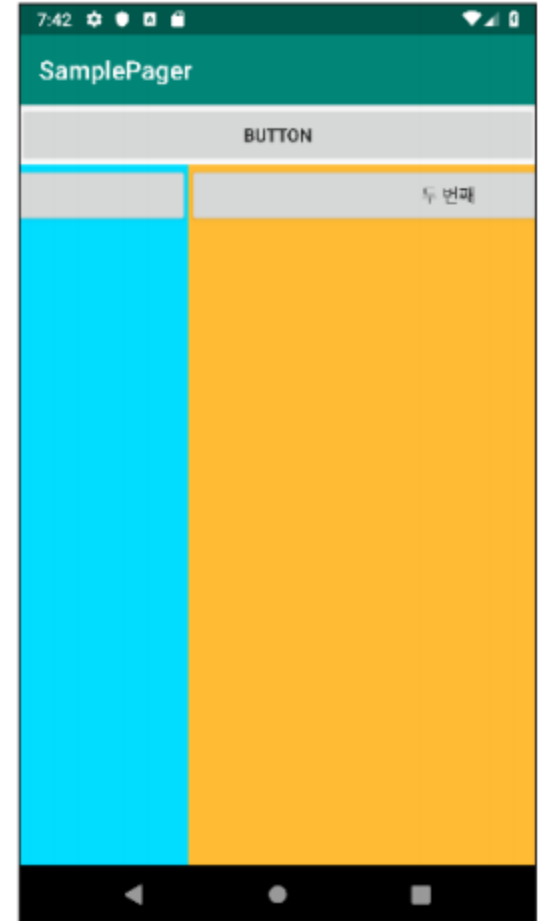
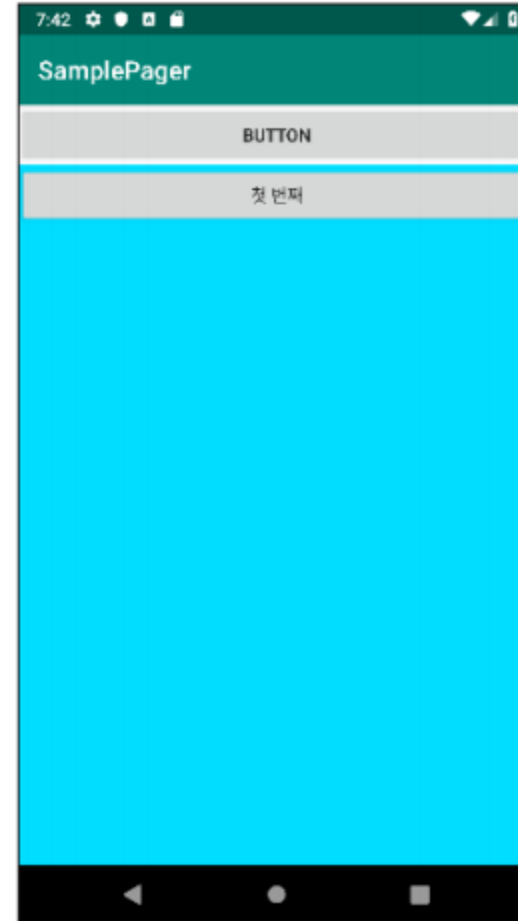
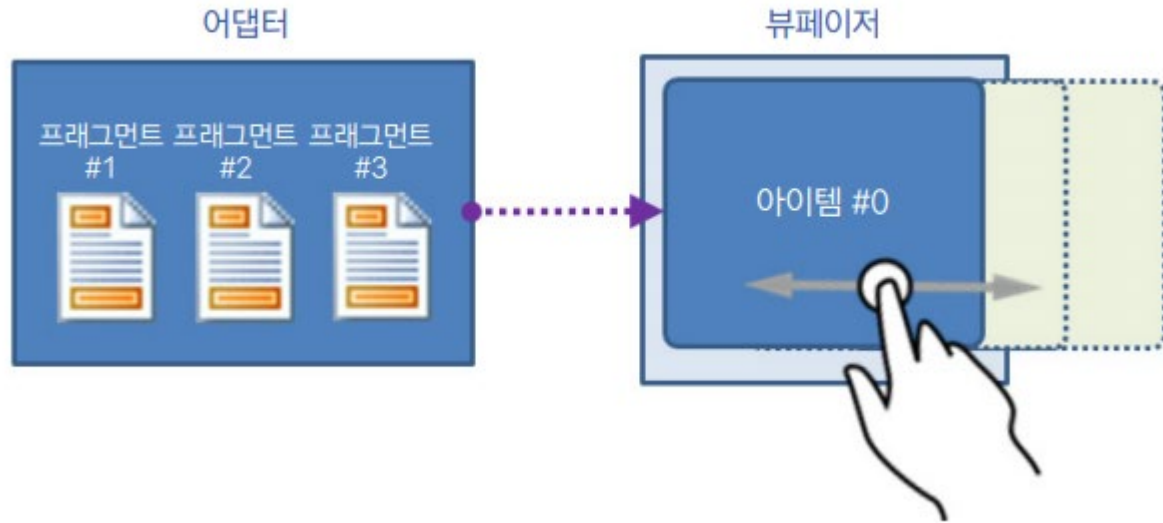
24
25 BottomNavigationView bottomNavigation = findViewById(R.id.bottom_navigation);
26 bottomNavigation.setOnNavigationItemSelectedListener(
27     new BottomNavigationView.OnNavigationItemSelectedListener() {
28         @Override
29         public boolean onNavigationItemSelected(@NonNull MenuItem item) {
30
31             switch (item.getItemId()) {
32                 case R.id.tab1:
33                     getSupportFragmentManager().beginTransaction().replace(R.id.container, fragment1).commit();
34                     return true;
35                 case R.id.tab2:
36                     getSupportFragmentManager().beginTransaction().replace(R.id.container, fragment2).commit();
37                     return true;
38                 case R.id.tab3:
39                     getSupportFragmentManager().beginTransaction().replace(R.id.container, fragment3).commit();
40                     return true;
41             }
42             return false;
43         }
44     }
45 );
46 }
47

```

에뮬레이터 실행



실습 - ③



activity_main.xml
MainActivity.java

버튼

fragment1.xml
Fragment1.java

fragment2.xml
Fragment2.java

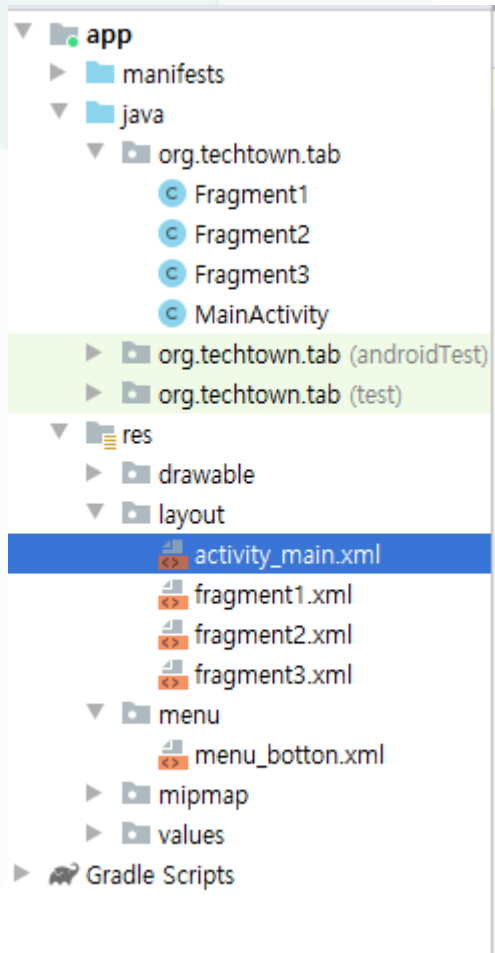
fragment3.xml
Fragment3.java

새 프로젝트 이름 : Gs20200csmpager
패키지명 : org.techtown.pager

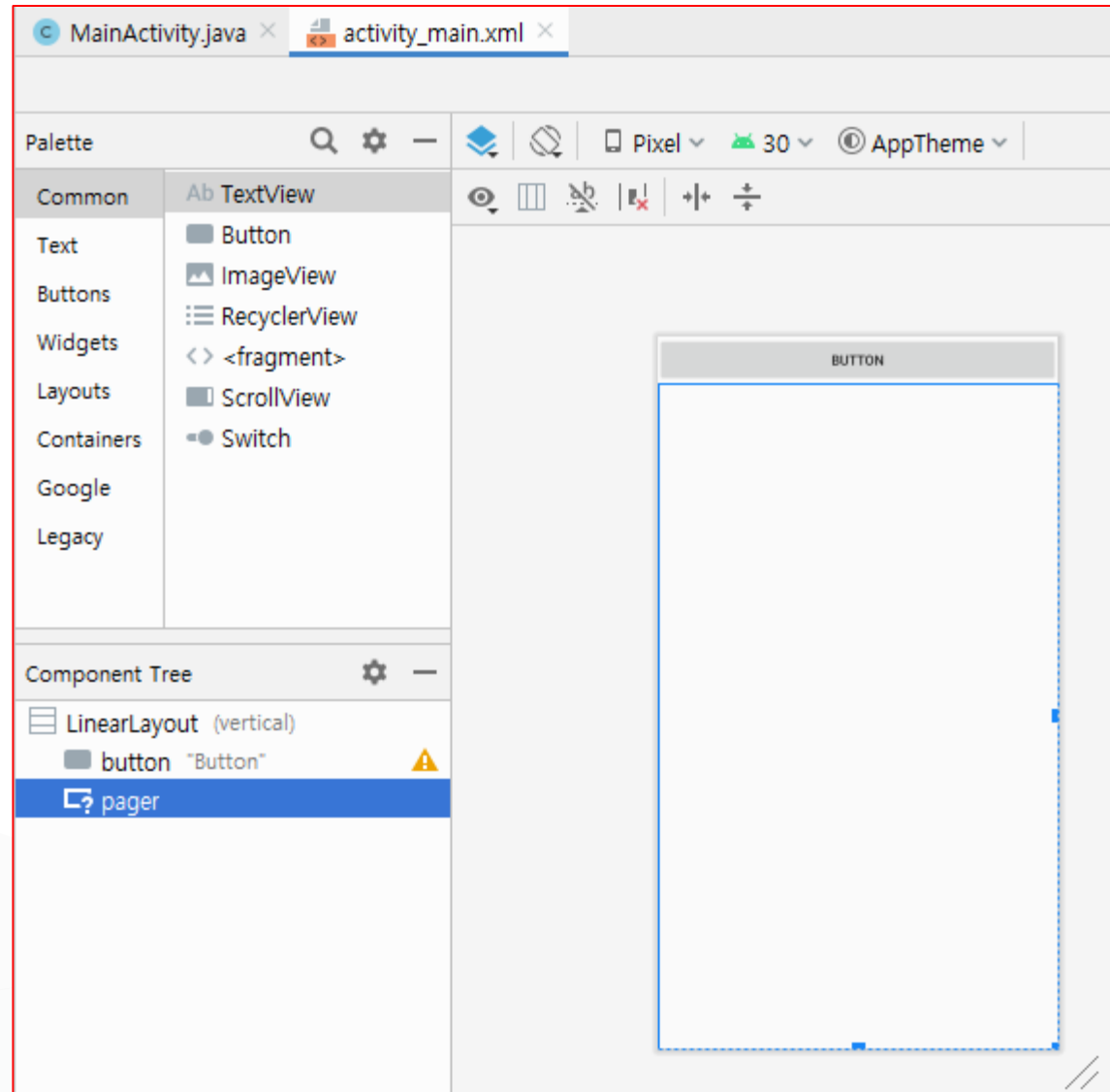
실습 순서

- ▶ fragment1.xml , fragment2.xml , fragment3.xml 추가
- ▶ Fragnnet1.java , Fragnnet2.java , Fragnnet3.java 추가
- ▶ Activity_main.xml 작업
- ▶ MainActivity.java 작업
- ▶ 에뮬레이터 결과 확인

이전 프로젝트에서 fragment1,2,3.xml
Fragmnet1,2,3.java 복사 - 붙여넣기



```
MainActivity.java x activity_main.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   xmlns:tools="http://schemas.android.com/tools"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   android:orientation="vertical"
8   tools:context=".MainActivity" >
9
10   <Button
11     android:id="@+id/button"
12     android:layout_width="match_parent"
13     android:layout_height="wrap_content"
14     android:text="Button" />
15
16   <androidx.viewpager2.widget.ViewPager2
17     android:id="@+id/pager"
18     android:layout_width="match_parent"
19     android:layout_height="match_parent" />
20 </LinearLayout>
```



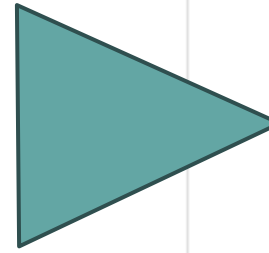
```
1
2     package org.techtown.pager;
3
4     import androidx.annotation.NonNull;
5     import androidx.appcompat.app.AppCompatActivity;
6     import androidx.fragment.app.Fragment;
7     import androidx.fragment.app.FragmentManager;
8     import androidx.fragment.app.FragmentStatePagerAdapter;
9     import androidx.lifecycle.Lifecycle;
10    import androidx.viewpager.widget.PagerAdapter;
11    import androidx.viewpager2.adapter.FragmentStateAdapter;
12    import androidx.viewpager2.widget.ViewPager2;
13
14    import android.os.Bundle;
15    import android.view.View;
16    import android.widget.Button;
17
18    import java.util.ArrayList;
19
20    public class MainActivity extends AppCompatActivity {
21        ViewPager2 pager;
22        @Override
23        protected void onCreate(Bundle savedInstanceState) {
24            super.onCreate(savedInstanceState);
25            setContentView(R.layout.activity_main);
26
27            pager = findViewById(R.id.pager);
28            pager.setOrientation(ViewPager2.ORIENTATION_HORIZONTAL);
```

```

29 pager.setOffscreenPageLimit(3);
30
31 MyPagerAdapter adapter = new MyPagerAdapter(getSupportFragmentManager(), getLifecycle());
32 pager.setAdapter(adapter);
33
34 Button button = findViewById(R.id.button);
35 button.setOnClickListener(new View.OnClickListener() {
36     @Override
37     public void onClick(View view){
38         pager.setCurrentItem(1);
39     }
40 });
41
42 class MyPagerAdapter extends FragmentStateAdapter {
43     int itemCount = 3;
44     public MyPagerAdapter(FragmentManager fm, Lifecycle cycle) { super(fm, cycle); }
45
46     @Override
47     public int getItemCount() { return itemCount; }
48
49     @NonNull
50     @Override
51     public Fragment createFragment(int position) {
52         switch(position) {
53             case 0: {
54                 return new Fragment1();
55             }
56             case 1: {
57                 return new Fragment2();
58             }
59             default: {
60                 return new Fragment3();
61             }
62         }
63     }
64 }

```

에뮬레이터 실행



```

61 case 2: {
62     return new Fragment3();
63 }
64 }
65 return new Fragment1();
66 }
67 }
68 }
69

```

세개의 기능을 합쳐서 하나의 앱에서 작동하도록 하세요.
상위탭 세개 클릭-> 하위탭 세개 클릭->
뷰페이지 세번 실행의 과정을 녹화

모두 합치지 못할 경우 가능한 것만 녹화하여 제출
부분 점수 인정

모두 함께 실행:100

2개 함께 실행: 90

파일저장은 20XXX아무개실습9

각각 실행: 80

매 시간 실습 점수는 누적하여 후에 변환하여 반영됨.