



안드로이드 프로그래밍

강사 : 정지영



안드로이드 이해하기

스마트폰 대중화



- 안드로이드폰의 운영체제: 안드로이드
- 아이폰의 운영체제: iOS

안드로이드의 흐름 살펴보기

대규모 시장 점유



- 안드로이드 점유율: 74%
 - iOS 점유율: 23%
- (2019년 1월 기준 전세계 점유율)



안드로이드의 주요 특징

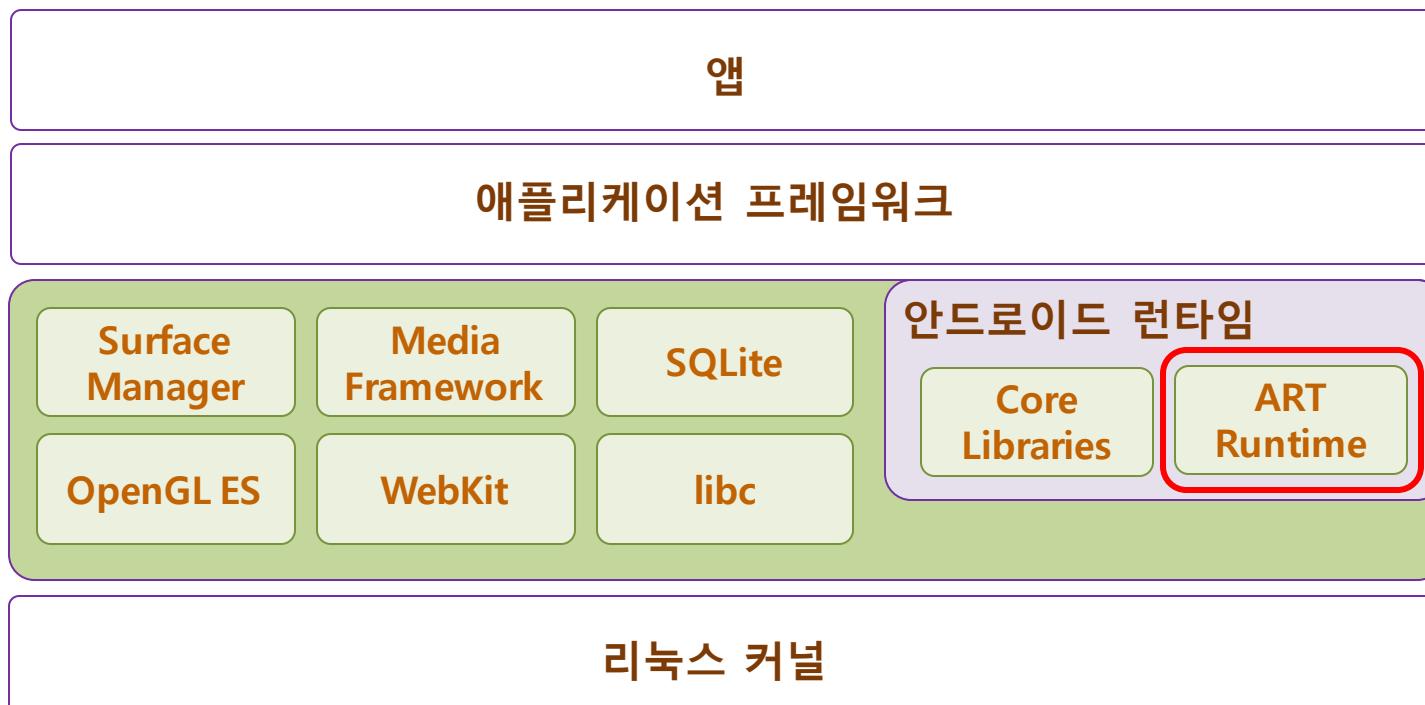




안드로이드의 주요 특징

플랫폼 아키텍쳐

- 리눅스 커널 위에서 동작
- 빌드 시에는 DEX 포맷으로 만들어지고 실행 시에는 ART 런타임에서 실행됨
- 앱은 애플리케이션 프레임워크 위에서 실행됨





안드로이드 버전의 변화

- 스마트폰, 태블릿, 그리고 빠른 업그레이드 ...





안드로이드 에코 시스템

- 에코 시스템 안에서 개발자로서의 역할





개발도구 설치하기

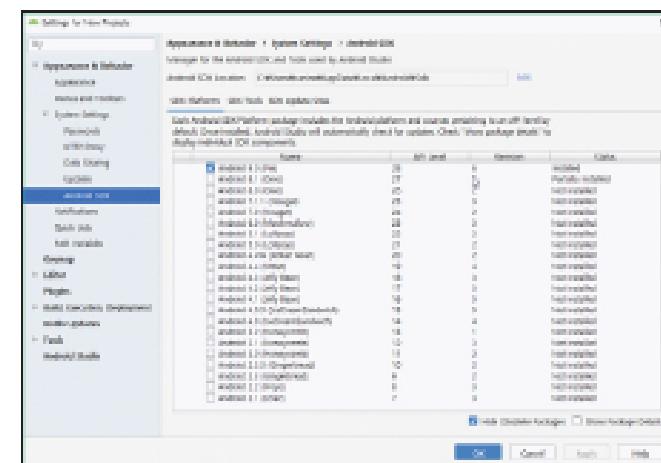
안드로이드 스튜디오 설치하기

android studio

안드로이드 스튜디오



SDK 매니저를 이용해 추가 설치하기



SDK 매니저



안드로이드 스튜디오 다운로드

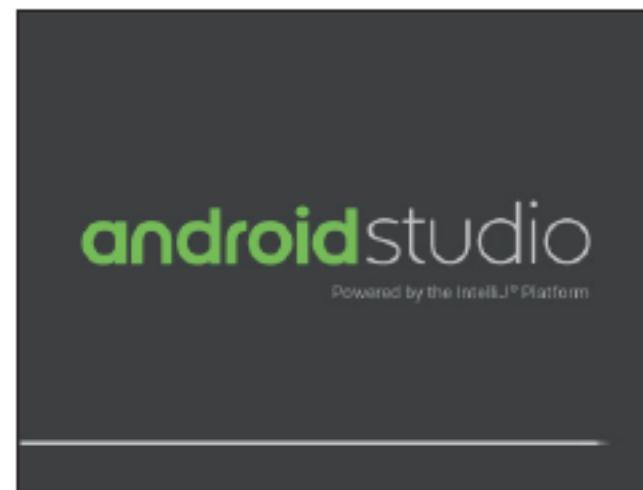
- <https://developer.android.com/studio>

The screenshot shows a web browser displaying the Android Studio download page at <https://developer.android.com/studio>. The page features the Android logo and the text "android studio". It states that "Android Studio provides the fastest tools for building apps on every type of Android device." A prominent green button labeled "DOWNLOAD ANDROID STUDIO" is centered, with a red arrow pointing towards it from the right side of the image. Below the button, it says "3.3.1 for Windows 64-bit (947 MB)". At the bottom of the main content area, there are links for "DOWNLOAD OPTIONS" and "RELEASE NOTES". To the right of the main content, a grey callout box contains the Korean text: "익스플로러에서 [다운로드] 버튼이 작동하지 않을 수 있습니다. 그럴 때는 크롬 등 다른 브라우저에서 다운받으세요." (The [Download] button may not work in Explorer. In that case, download it using Chrome or another browser.)



설치 진행

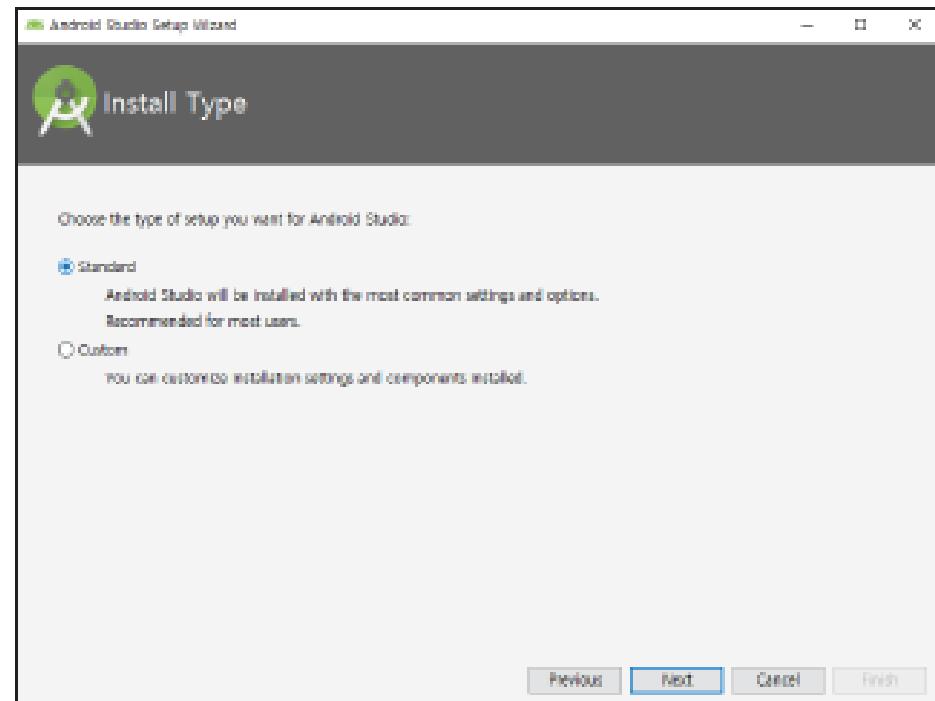
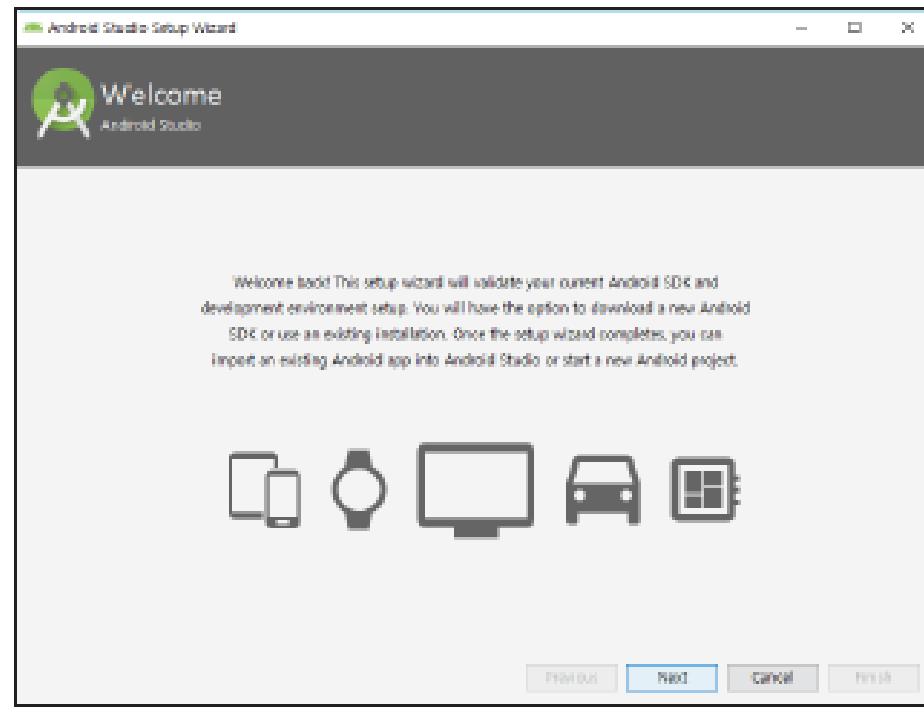
- 설치 파일을 실행하여 설치 진행
- 설치 후에 안드로이드 스튜디오 실행됨





추가 설치

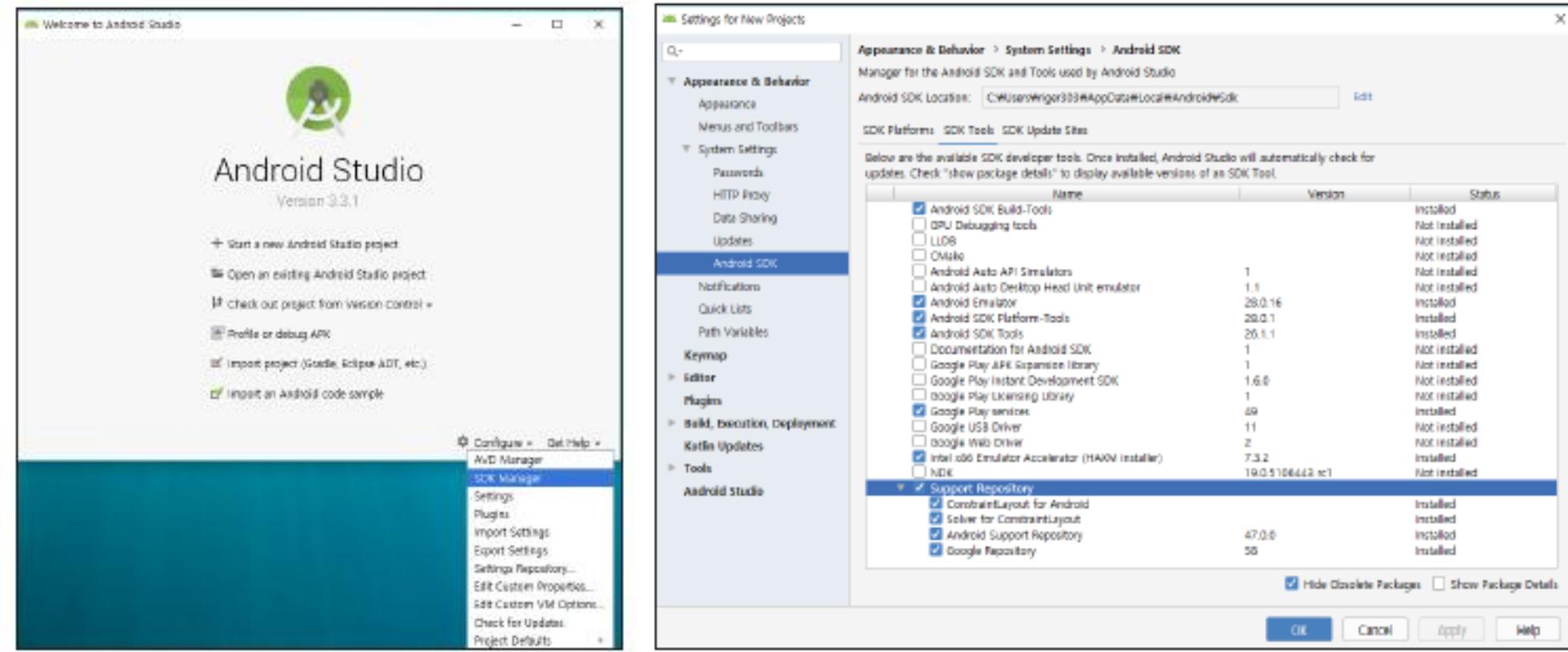
- 최초 설치 시에는 필요한 것들을 추가 설치해야 함
- 추가 설치 진행





구성요소 설치

- 추가 설치가 완료되어 안드로이드 스튜디오 시작화면이 보이면 SDK Manager 실행
- SDK Tools 탭에서 필요한 구성요소를 추가로 선택하여 설치



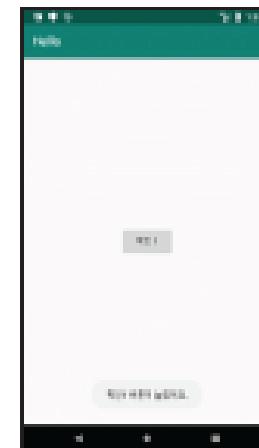


첫번째 앱 만들기

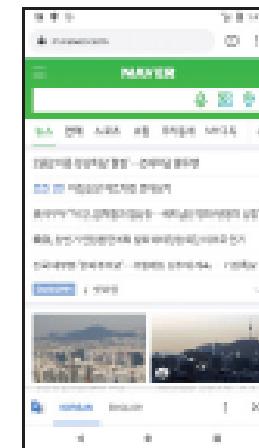
프로젝트 만들기



하나씩 바꾸어보기



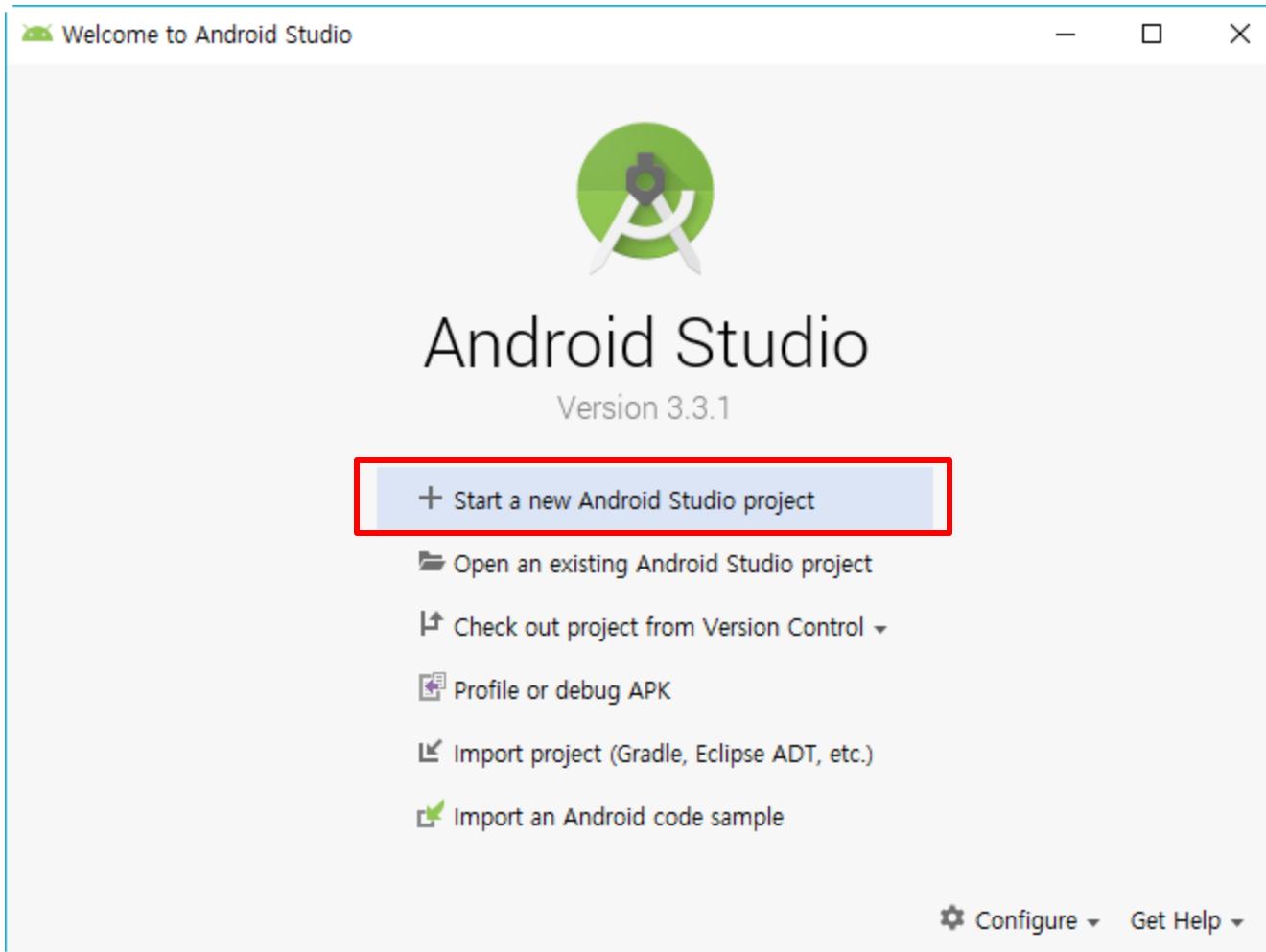
여러 개의 버튼에 기능 추가하기





시작 화면에서 새로운 프로젝트 만들기

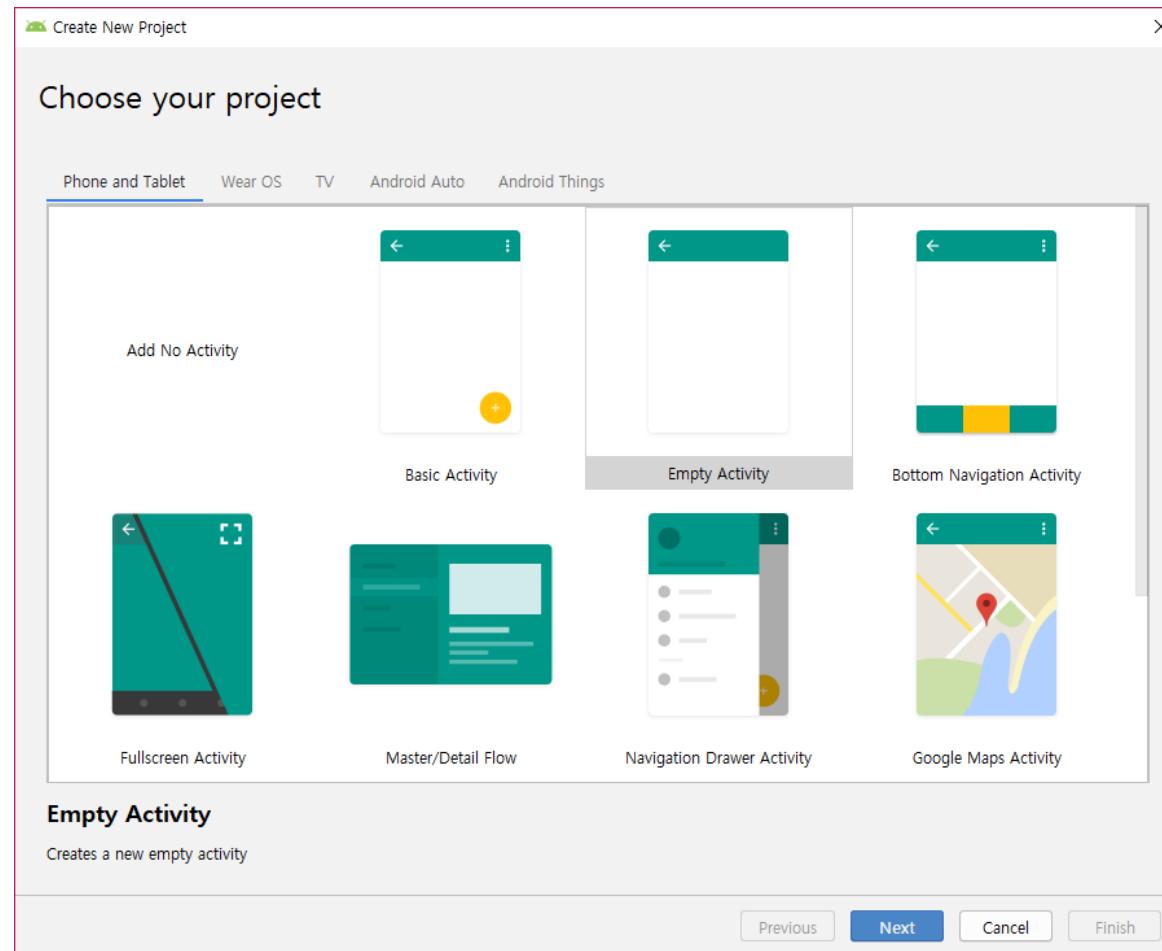
(1) 'Start a new Android Studio project' 메뉴 선택





첫 화면의 모양 선택하기

(2) 'Empty Activity'가 선택된 상태 그대로 둠

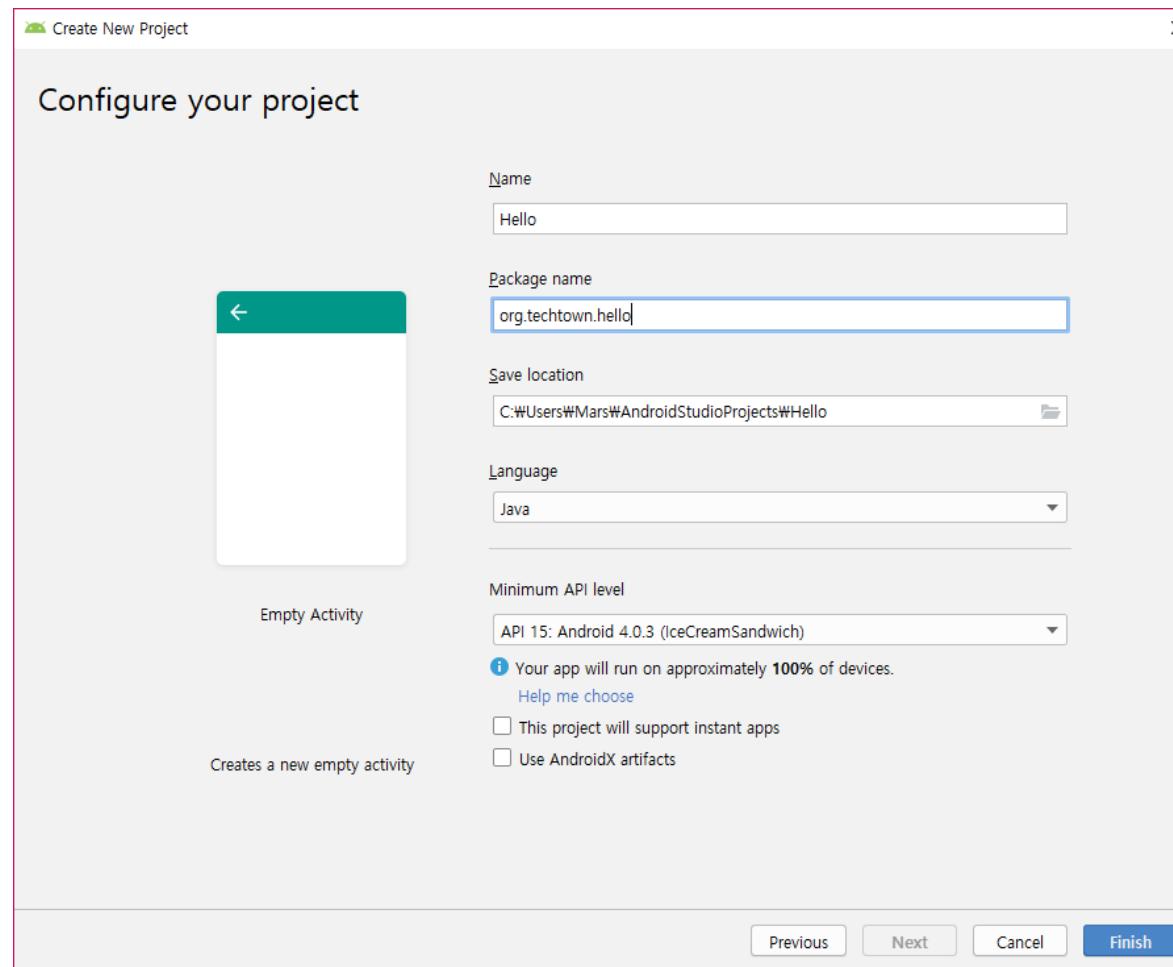




프로젝트 이름 입력하기

(3) Name: 항목에 'Hello' 입력

Package name: 항목을 'org.techtown.hello'로 설정





안내글 표시 영역 닫기

(4) 오른쪽의 안내글 표시 영역 닫기

The screenshot shows the Android Studio interface with the 'What's New' panel closed. A red box highlights the close button in the top right corner of the panel header.

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Hello > app > src > main > java > org > techtown > hello > MainActivity.java [app] - Android Studio

activity_main.xml MainActivity.java

```
1 package org.techtown.hello;
2
3 import ...
4
5 public class MainActivity extends AppCompatActivity {
6
7     @Override
8     protected void onCreate(Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10        setContentView(R.layout.activity_main);
11    }
12}
```

Project

- app
 - manifests
 - java
 - org.techtown.hello
 - MainActivity
 - generatedJava
 - res
- Gradle Scripts

Assistant What's New

What's New in 3.3

This panel describes some of the new features and behavior changes included in this update.

To open this panel again later, select Help > What's New in Android Studio from the main menu.

[Read in a browser](#)

Navigation Editor

Build Sync

- Build: completed successfully at 2018-11-22 오전 7:05
- Run build C:\Users\Mars\AndroidStudioProjects\Hello
 - Load build
 - Configure build
 - Calculate task graph
 - Run tasks

Layout Captures

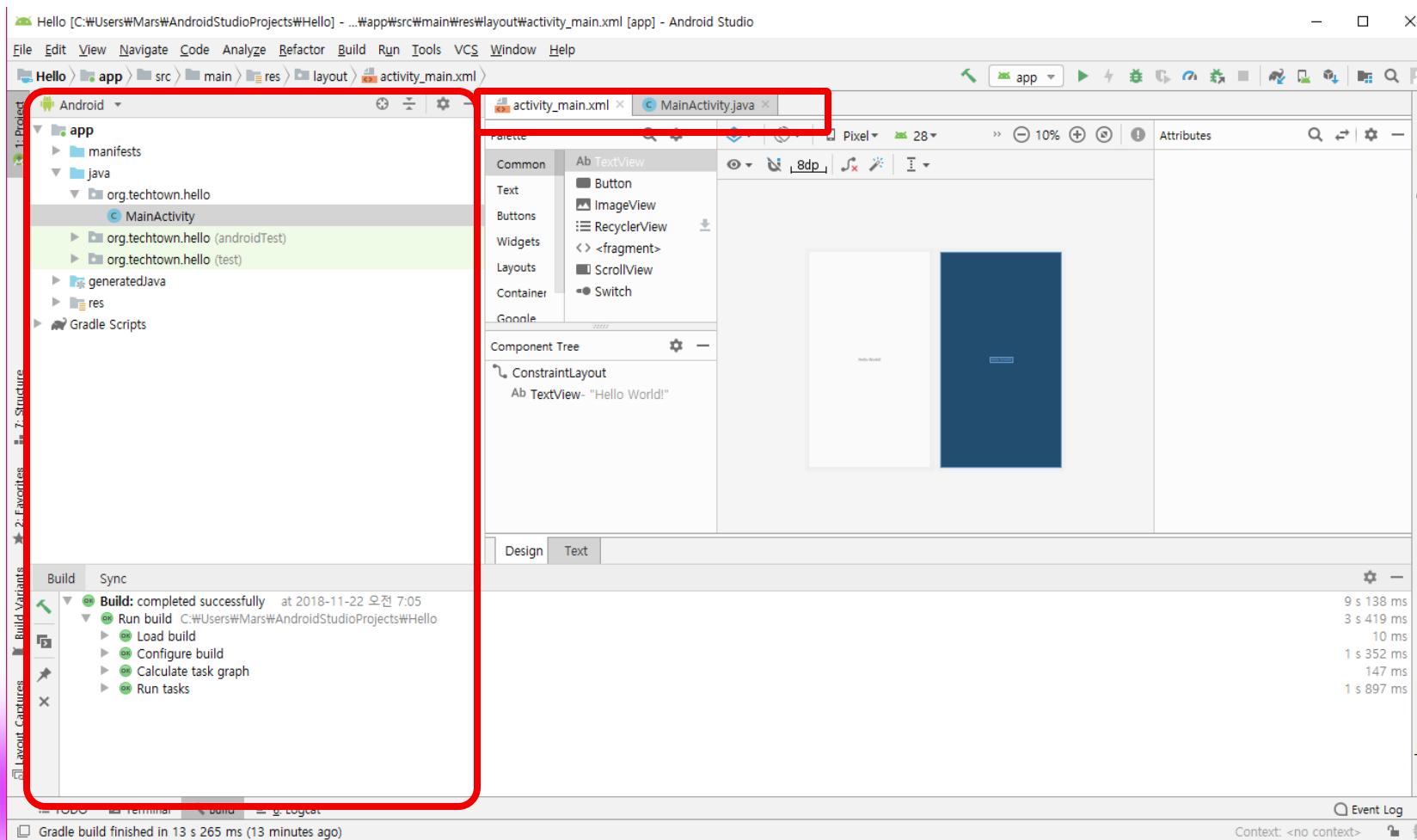
Event Log

Gradle build finished in 13 s 265 ms (8 minutes ago)



생성된 프로젝트 확인

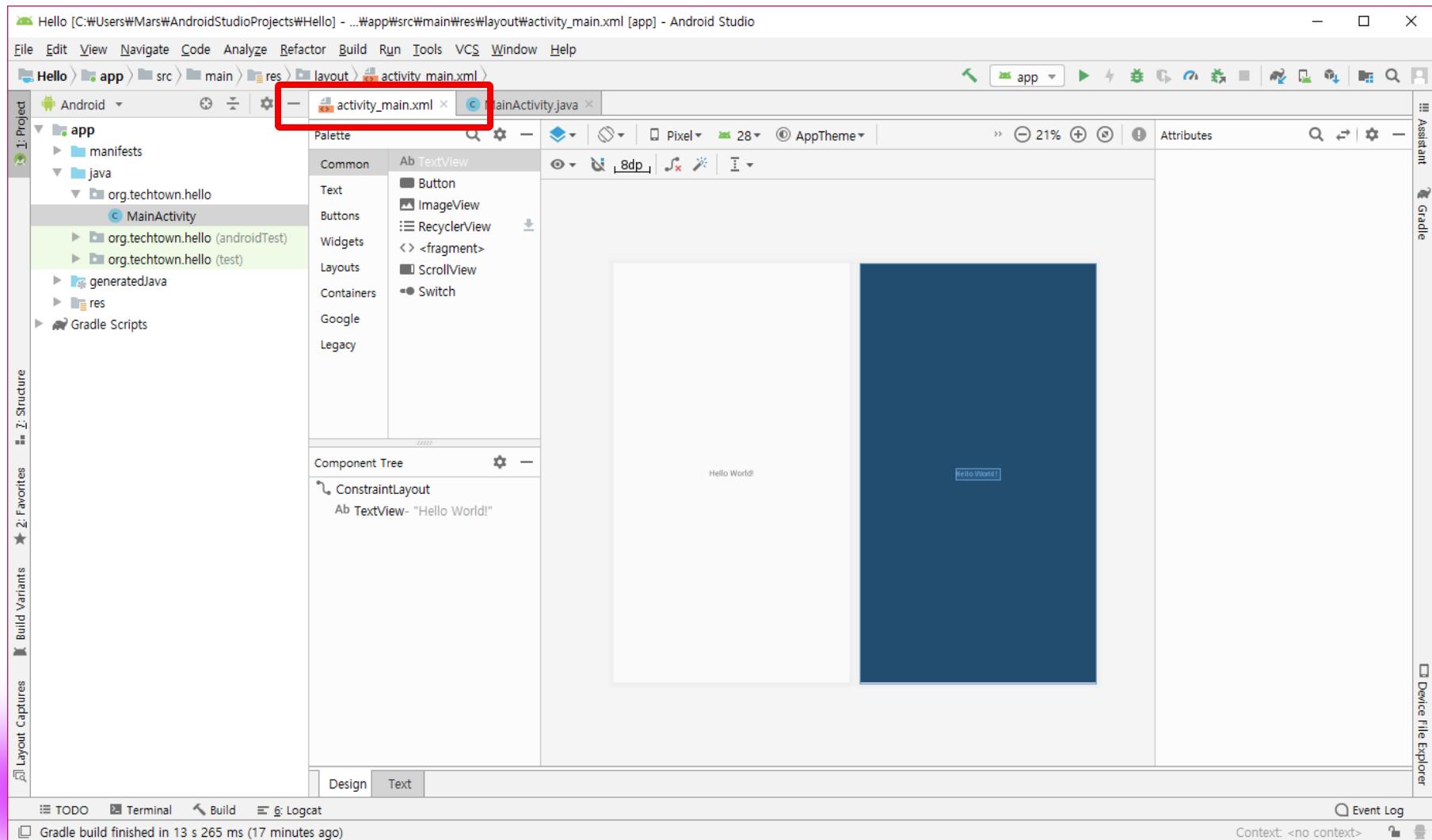
- 왼쪽의 프로젝트 화면에 새로 만들어진 프로젝트의 내용 표시
- 오른쪽에 자동으로 만들어진 두 개 파일이 탭으로 표시





Xml 텁 눌러보기

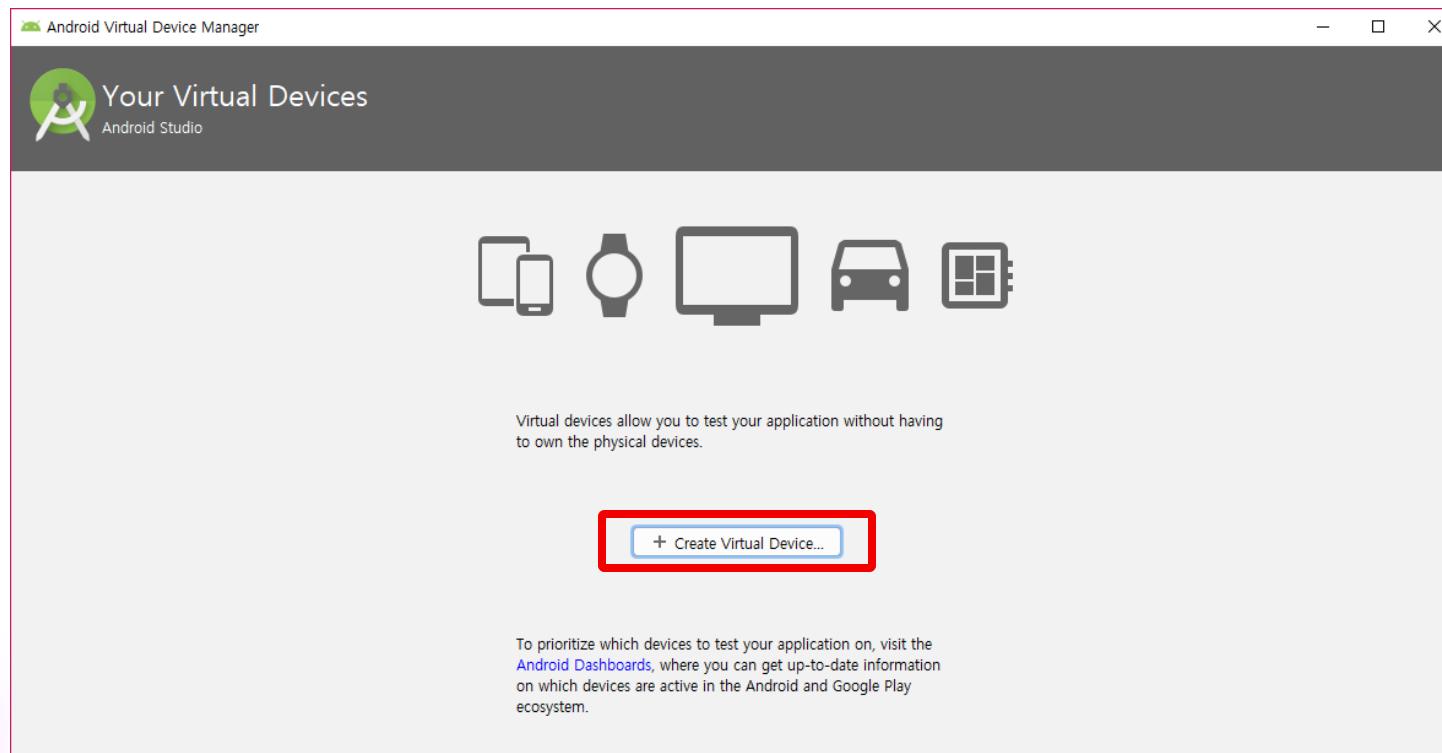
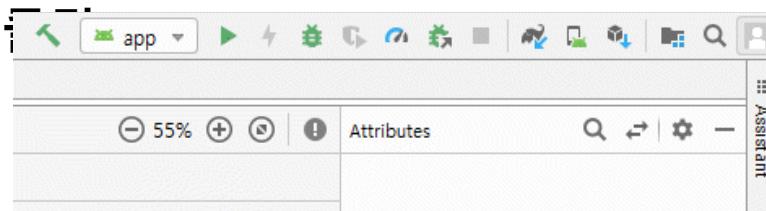
- 가운데 영역의 두 개 텁 중에 activity_main.xml 텁 선택





에뮬레이터 만들기

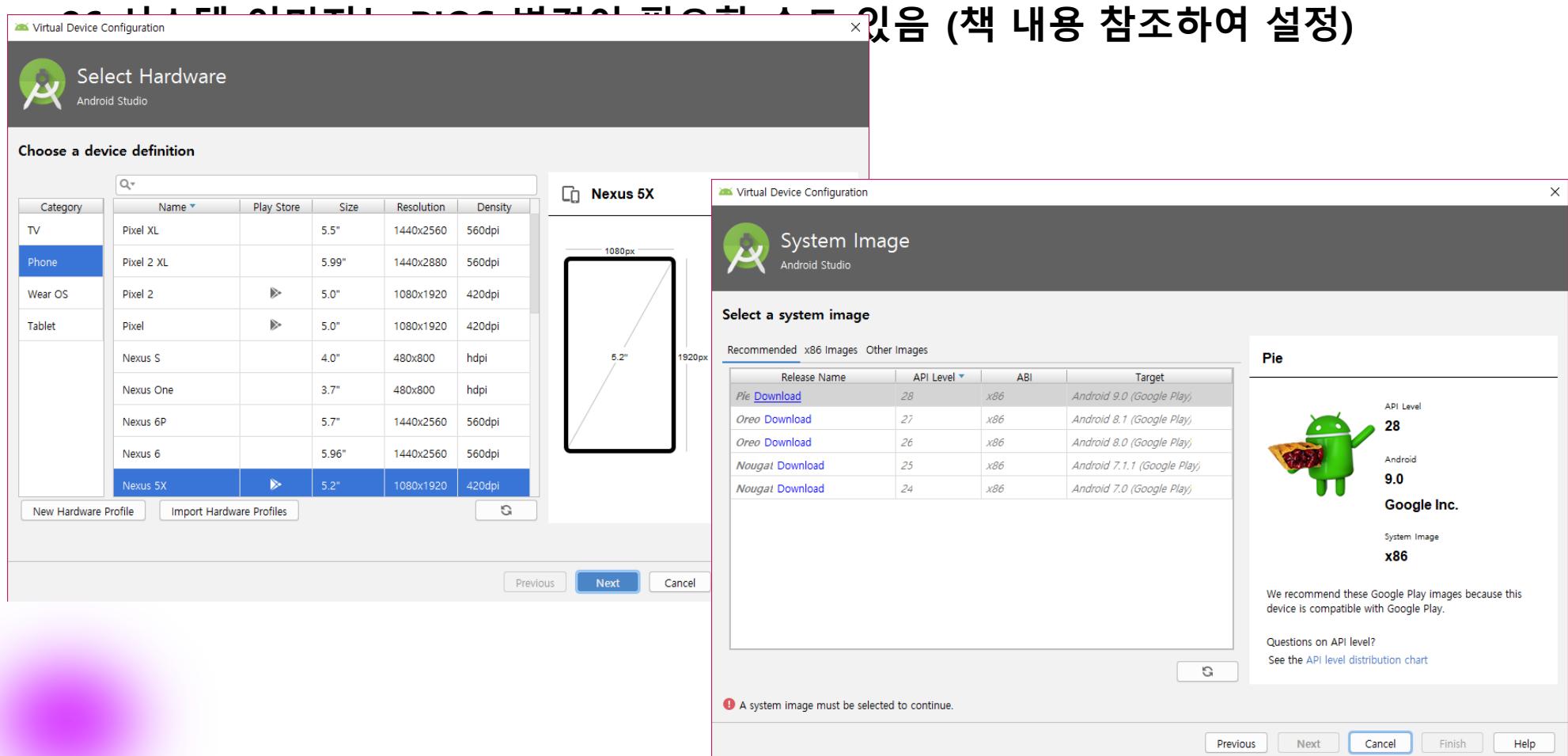
- 상단의 아이콘 중에서 [AVD Manager] 아이콘 클릭하고 [Create Virtual Device] 버튼





에뮬레이터 만들기

- Nexus 5X, Pie 시스템 이미지 선택된 상태로 [Next] 버튼 클릭 (필요 시 Download 클릭)





에뮬레이터 만들기

- 에뮬레이터 정보 확인 화면에서 [Finish] 버튼 클릭하면 에뮬레이터 생성되어 리스트에

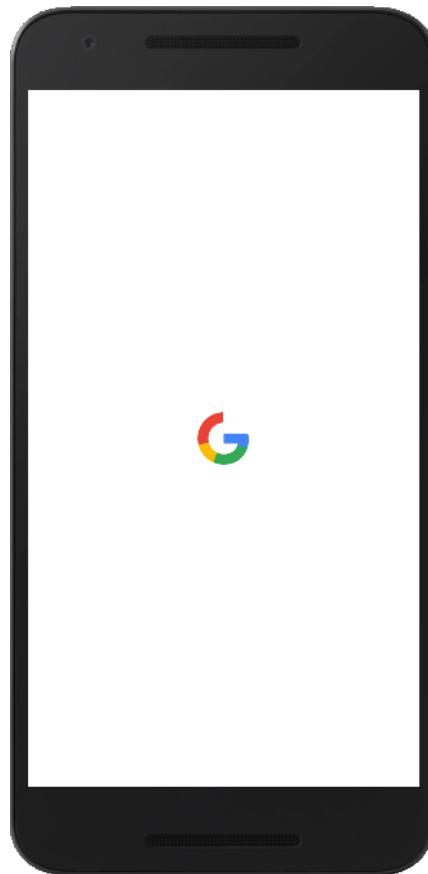
The image shows two windows from the Android Studio interface. On the left is the 'Virtual Device Configuration' dialog, which is a 'Verify Configuration' screen. It displays settings for a new virtual device named 'Nexus 5X API 28'. The configuration includes a Nexus 5X device type at 5.2 inches with a resolution of 1080x1920 at 420dpi, running Android 9.0 (Pie) x86. The startup orientation is set to Portrait. Under 'Emulated Performance', 'Graphics' is set to 'Automatic'. A checkbox for 'Device Frame' is checked. At the bottom, there's a 'Show Advanced Settings' button and a 'Previous' button. On the right is the 'Android Virtual Device Manager' window titled 'Your Virtual Devices'. It lists the single device configuration: Type: Nexus 5X, Name: Nexus 5X API 28, Play Store: Not applicable, Resolution: 1080 x 1920: 420dpi, API: 28, Target: Android 9.0 (Google Pie), CPU/ABI: x86, Size on Disk: 513 MB. Below the table is a '+ Create Virtual Device...' button.

Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
Nexus 5X	Nexus 5X API 28		1080 x 1920: 420dpi	28	Android 9.0 (Google P...)	x86	513 MB	



에뮬레이터 실행

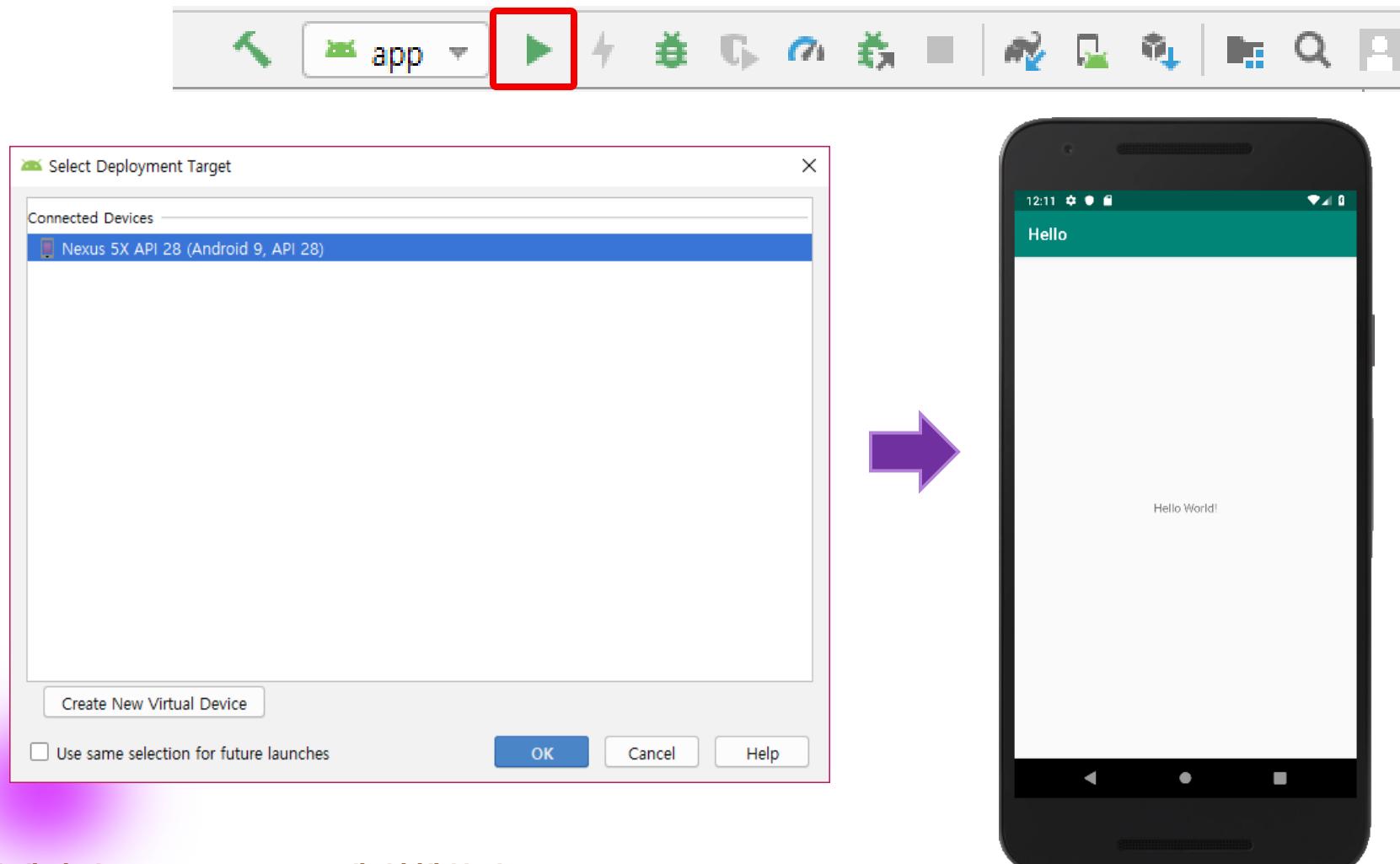
- 에뮬레이터 리스트에서 해당 항목의 [실행] 버튼 클릭





앱 실행하기

- 상단 툴바의 실행 버튼 클릭 → 연결되어 있는 단말 선택 → 앱 실행





첫번째 앱 살펴보기

- MainActivity.java 탭의 내용 – 자바 소스 코드

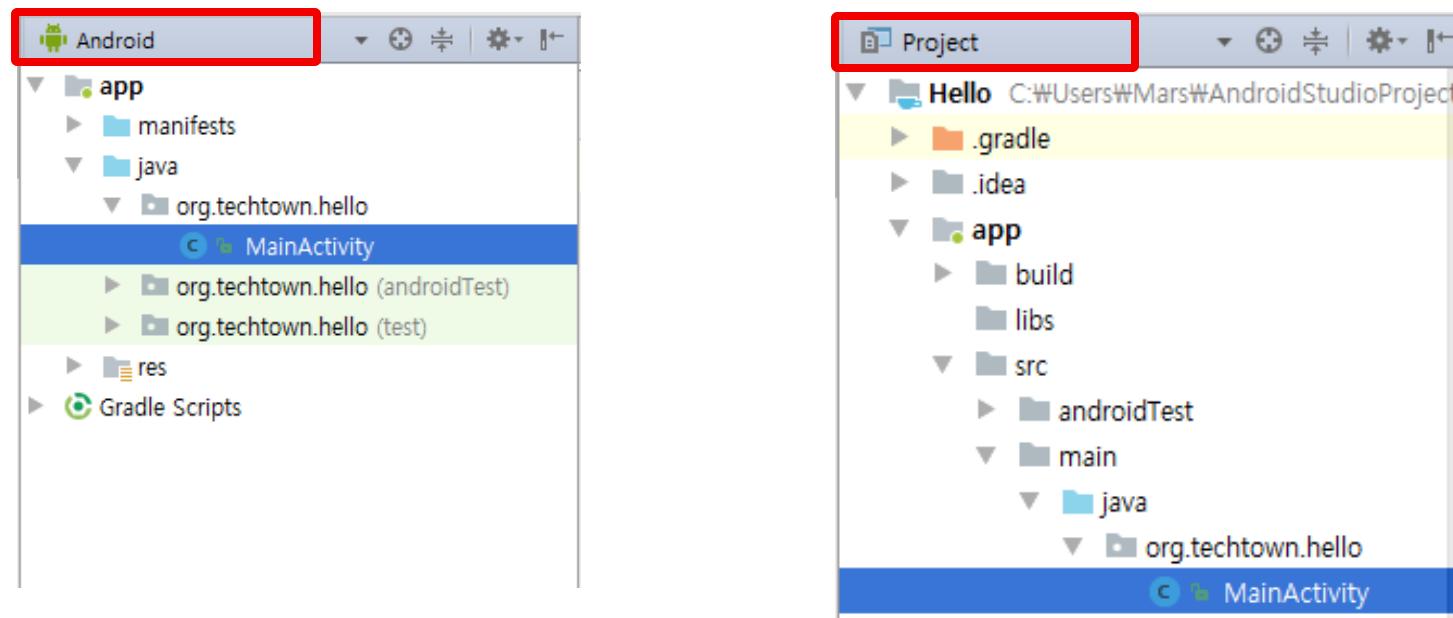
The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor has tabs for 'activity_main.xml' and 'MainActivity.java'. The 'MainActivity.java' tab is highlighted with a red rectangle. The code editor displays the following Java code:

```
1 package org.techtown.hello;
2
3 import ...
4
5
6 public class MainActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13 }
14
```



첫번째 앱 살펴보기

- 왼쪽 프로젝트 창에서 [Android] 탭 외에 [Project] 탭 선택해 보기





첫번째 앱 살펴보기

- 왼쪽 프로젝트 창에서 [res/layout] 폴더 안의 파일들과 activity_main.xml 파일 내용 살펴보기

The screenshot shows the Android Studio interface. On the left, the Project tool window displays the project structure under the 'app' module. The 'layout' folder contains 'activity_main.xml', which is highlighted and has a red border around it. Below the project tree, the 'Gradle Scripts' tab is selected.

In the center, the XML Editor shows the code for 'activity_main.xml'. The code defines a ConstraintLayout with a single TextView containing the text 'Hello World!'. The XML code is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    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"
    tools:context=".MainActivity">

    <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" />

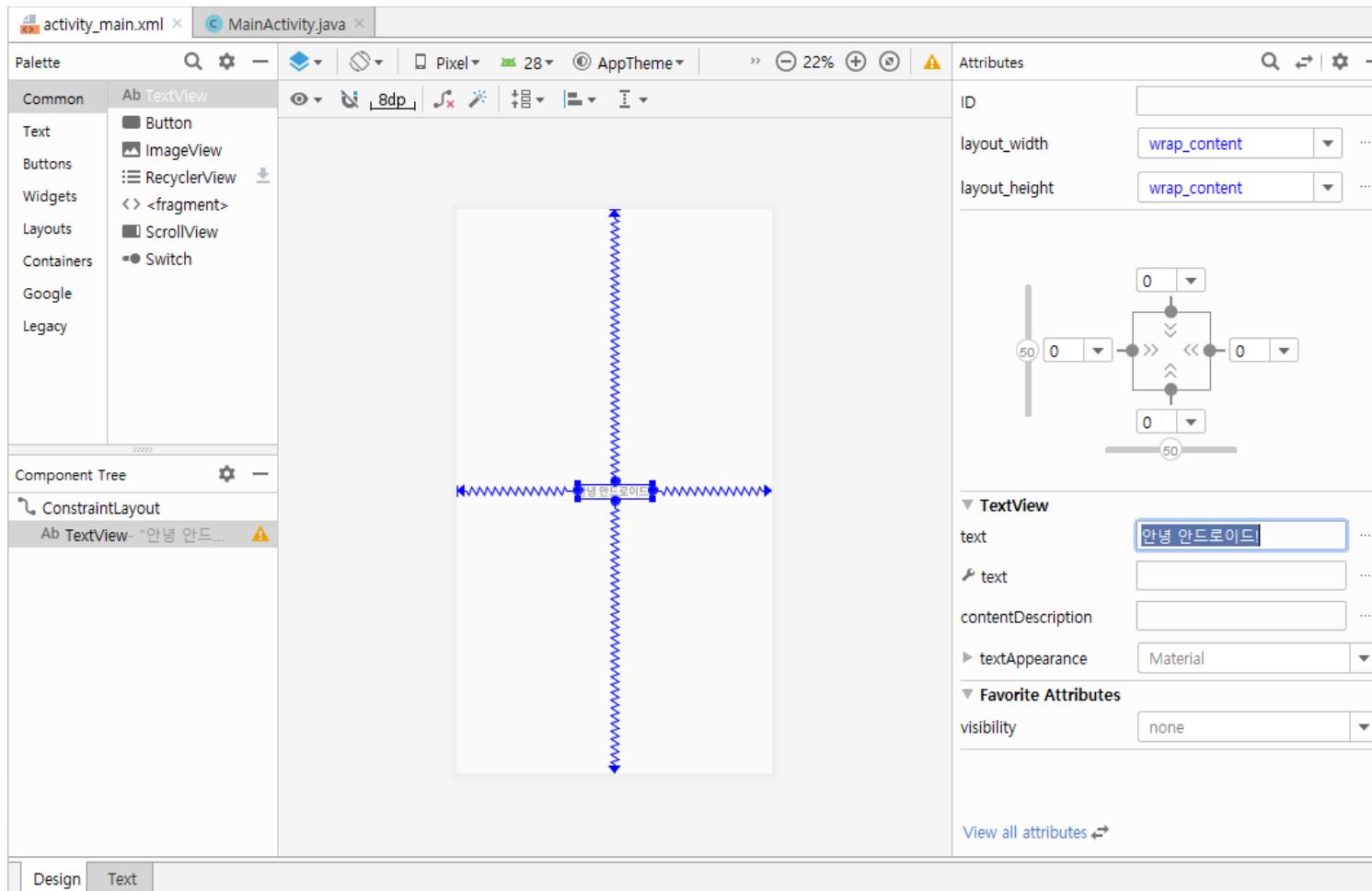
</android.support.constraint.ConstraintLayout>
```

On the right, the Preview pane shows a blue-bordered rectangular area with the text 'Hello World!' inside. At the bottom of the XML editor, there are two tabs: 'Design' and 'Text', with 'Text' being the active tab.



XML 파일에서 글자 바꾸어 보기

- TextView 태그의 text 속성을 바꾸면 글자를 바꿀 수 있음





안드로이드 스튜디오의 글자 크기 변경

- File > Settings 메뉴 → IDE Settings > Editor > Colors & Fonts > Font

The screenshot shows the Android Studio settings interface. The left sidebar has 'File' selected. In the main pane, 'Editor > Font' is selected. The 'Size' field is set to 22. A preview window shows text in a monospaced font at size 22. At the bottom are 'OK', 'Cancel', 'Apply', and 'Help' buttons.

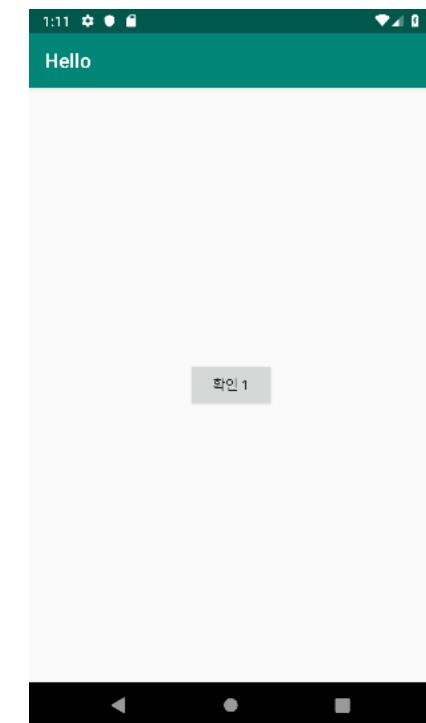
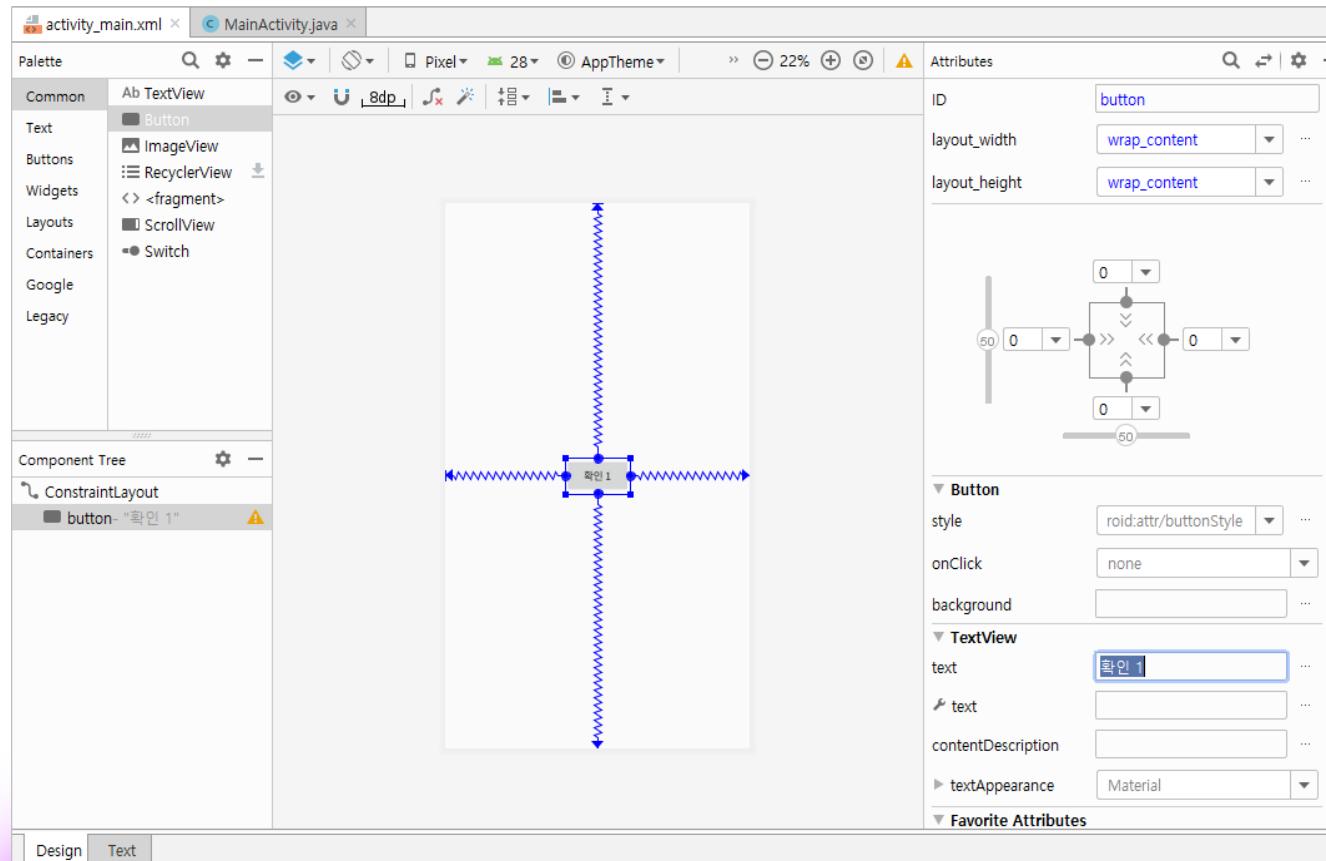
Font: Consolas
Size: 22
Line spacing: 1.0
Fallback font: <None>
Enable font ligatures

1 Android Studio is a full-featured IDE
2 with a high level of usability and outstanding
3 advanced code editing and refactoring support.
4
5 abcdefghijklmnopqrstuvwxyz 0123456789 (){}[]
6 ABCDEFGHIJKLMNOPQRSTUVWXYZ +-*/= .,:!?. #&\$%@|^



텍스트를 버튼으로 바꾸어 보기

- 빨레트에서 Button을 끌어다 화면 가운데 놓기





버튼을 눌렀을 때의 동작하도록 하기 위한 과정

● 방법 1 : 버튼에 onClick 속성 추가하고 소스에 메소드 추가

1. XML 레이아웃 파일에 들어있는 버튼에 onClick 속성 추가
2. 자바 코드에서 onClick 속성의 값으로 넣었던 메소드와 동일한 이름의 메소드 추가

● 방법 2 : 자바 소스에서 버튼 객체를 찾은 후

이벤트 처리를 위한 리스너 코드 추가

→ 둘째 마당에서 해 볼 내용

1. XML 레이아웃 파일에 들어있는 버튼에 ID 추가하기
2. 자바 코드에서 레이아웃 파일에 정의된 버튼 객체 참조하기
3. 이벤트 처리 코드 추가하기

* 화면을 위한 XML에 정의된 버튼과 기능을 소스 간의 연결 고리를 만들어주는 과정임



버튼 동작 방법 1

- XML 파일의 버튼에 onClick 속성 추가

[Hello>/res/layout/activity_main.xml]

```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:onClick="onButton1Clicked"  
    android:text="안녕 안드로이드!" />
```

The screenshot shows the Android Studio interface with the 'Design' tab selected. A ConstraintLayout contains a single button labeled "확인 1". The component tree on the left shows the button node.

The screenshot shows the 'Attributes' tab for the button. The 'onClick' attribute is highlighted with a red rectangle and set to "onButton1Clicked". Other attributes shown include ID (button), layout_width (wrap_content), layout_height (wrap_content), style (Widget.AppCompat.Button), background, and text (확인 1). The bottom section lists Favorite Attributes.



버튼 동작 방법 1

- 자바 파일에 onClick 속성에 추가했던 값과 똑같은 이름의 메소드 추가

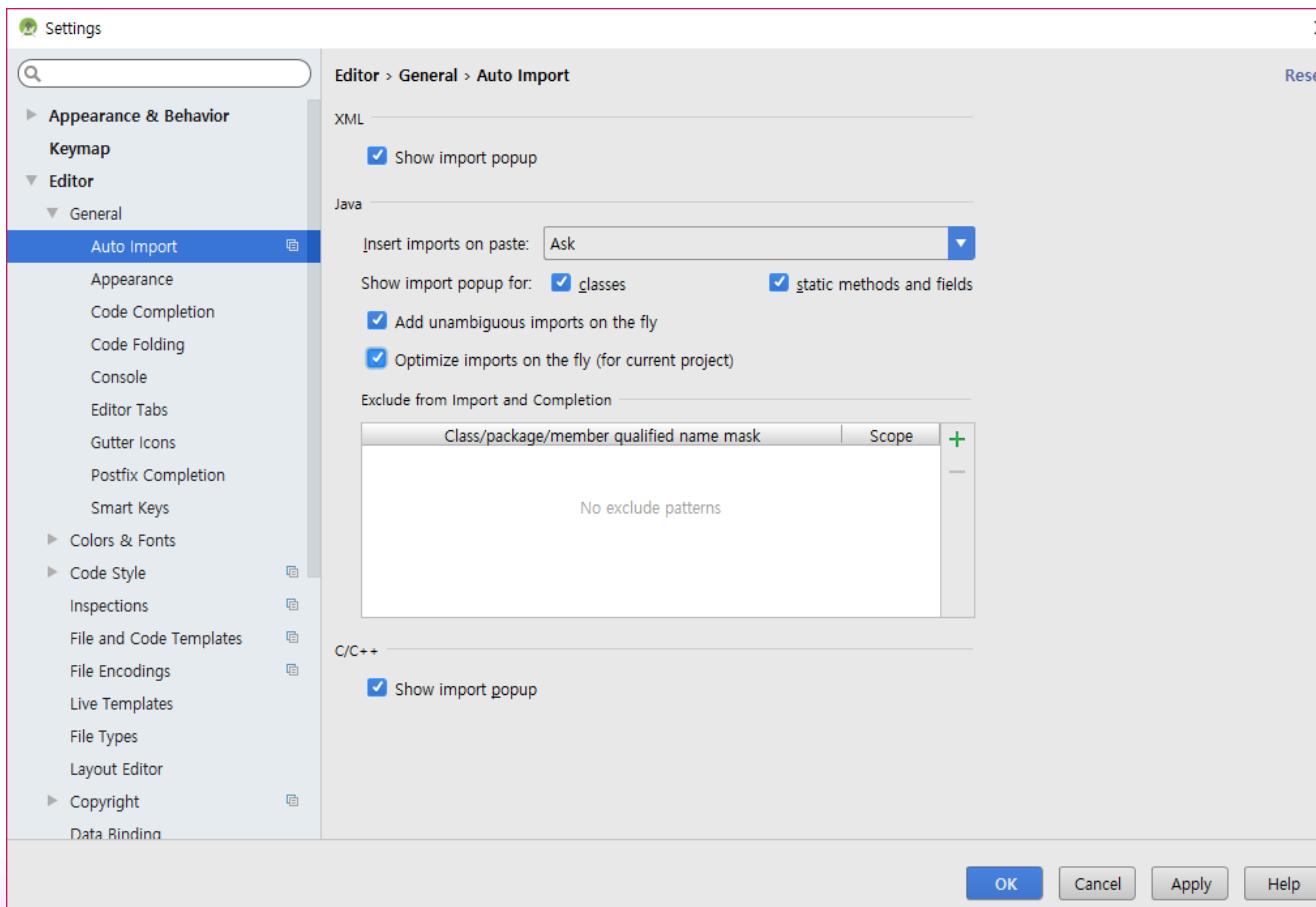
[Hello>/java/org.techtown.hello/MainActivity.java]

```
public void onButton1Clicked(View v) {  
    Toast.makeText(getApplicationContext(), "확인1 버튼이 눌렸어요.",  
        Toast.LENGTH_LONG).show();  
}
```



자동으로 import되도록 설정 변경

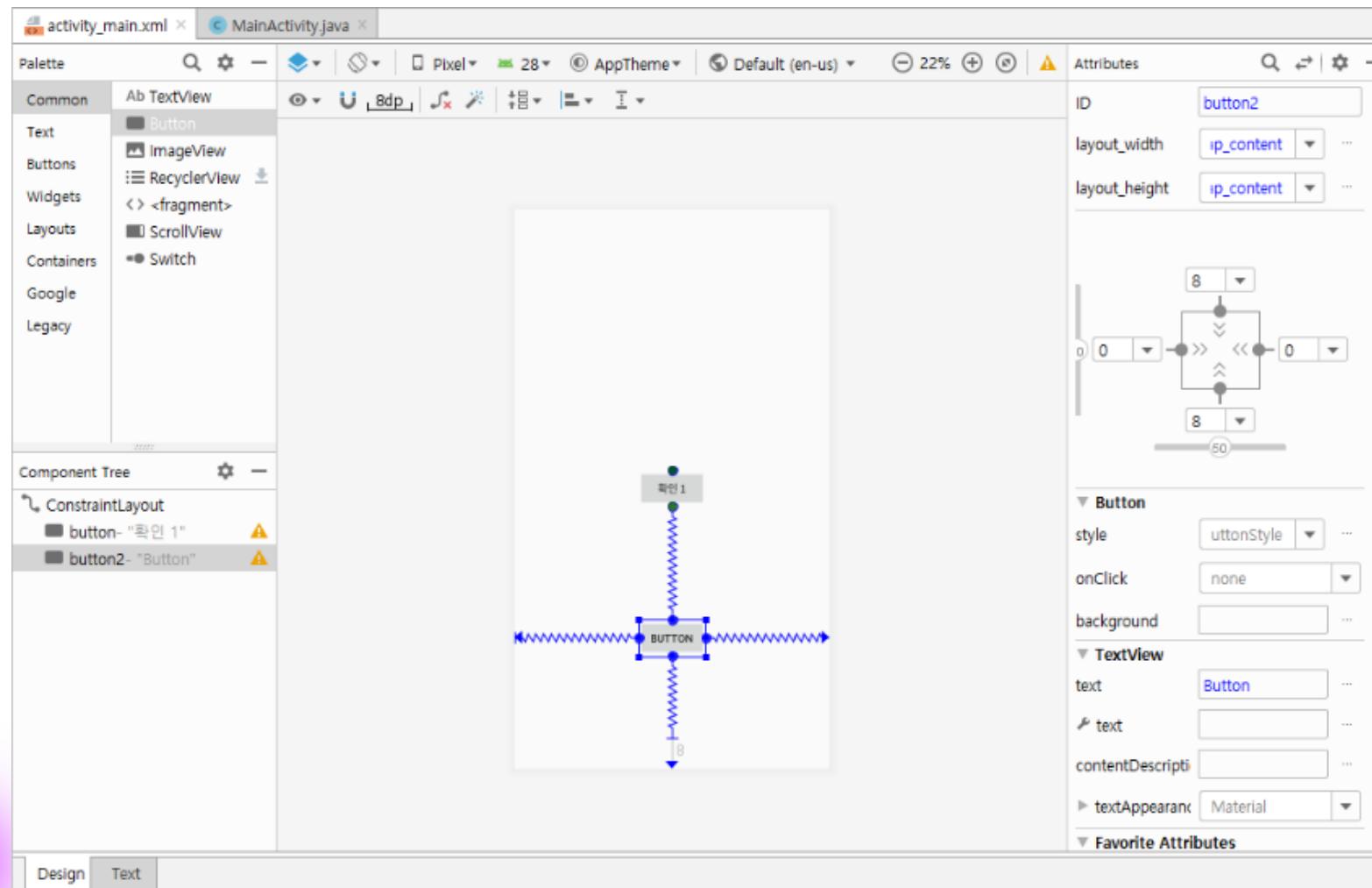
- File > Settings 화면 보면 IDE Settings > Editor > Auto Import





디자인 화면에서 버튼 추가하기

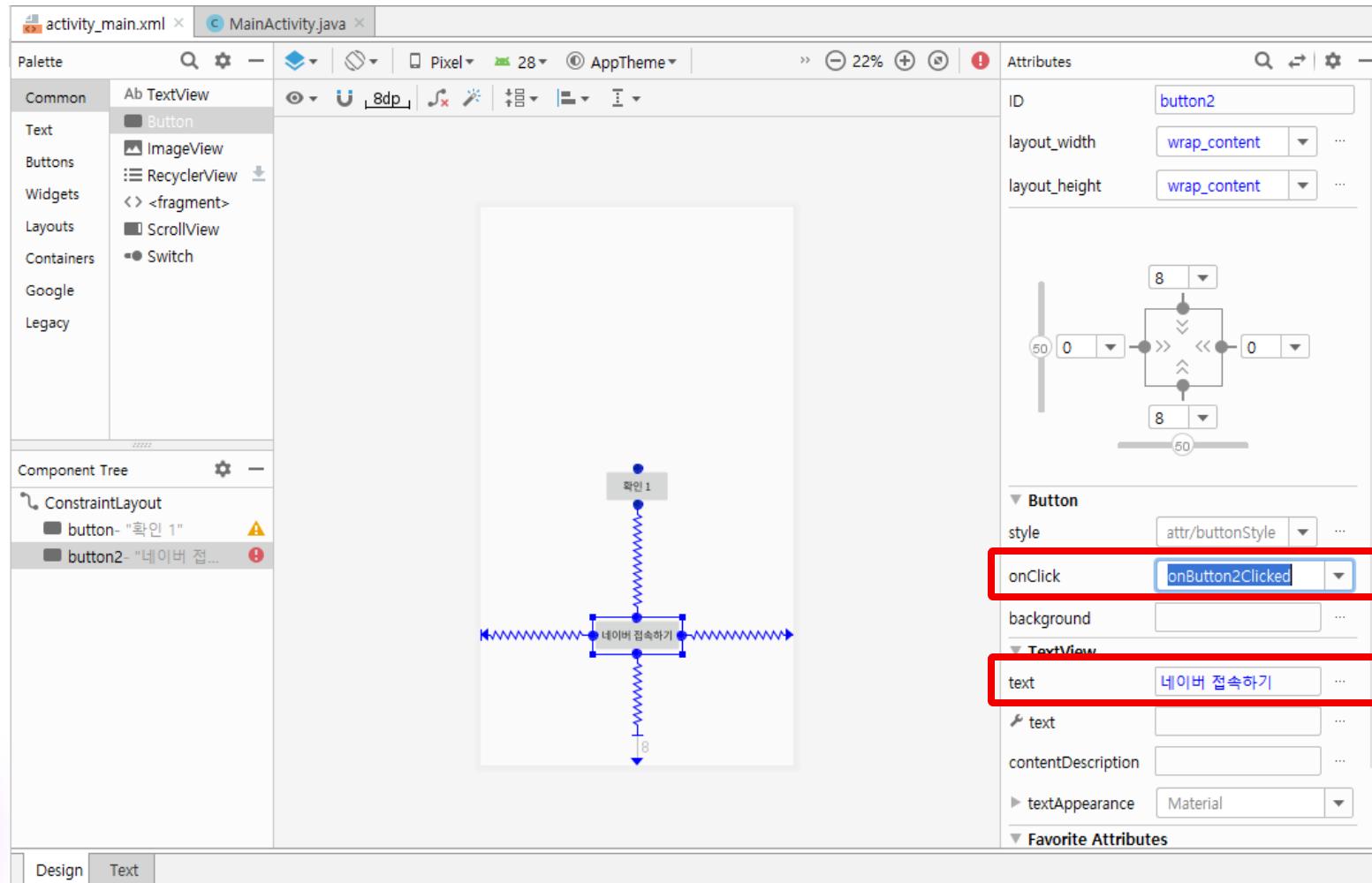
- ### • 두 번째 버튼 추가하기





text 속성과 onClick 속성 설정

- 오른쪽 Properties 영역에서 text 속성에 '네이버 접속하기', onClick 속성에 메소드 이름 설정





디자인 화면에서 버튼 추가하기

• 세 번째 버튼 추가하기

The screenshot shows the Android Studio interface with the Design tab selected. In the center is a ConstraintLayout containing three buttons labeled "확인 1", "네이버 접속하기", and "전화걸기". The "전화걸기" button has a tooltip "네이버 접속하기" above it. The "Buttons" category in the Palette is selected, and the "Button" tool is active. The Attributes panel on the right shows the following settings for the third button:

- ID: button3
- layout_width: wrap_content
- layout_height: wrap_content
- Button style: attr/buttonStyle
- onClick: onButton3Clicked
- Text: 전화걸기

The Component Tree panel shows the three buttons under a ConstraintLayout node.



자바 소스 파일에 메소드 추가

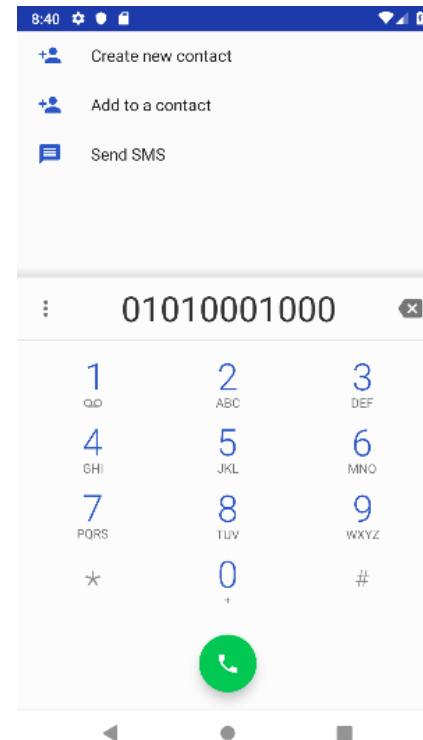
[Hello>/java/org.techtown.hello/MainActivity.java]

```
public void onButton1Clicked(View v) {  
    Intent myIntent = new Intent(Intent.ACTION_VIEW, Uri.parse("http://m.naver.com"));  
    startActivity(myIntent);  
}  
  
public void onButton2Clicked(View v) {  
    Intent myIntent = new Intent(Intent.ACTION_VIEW, Uri.parse("tel:010-1000-1000"));  
    startActivity(myIntent);  
}
```



버튼을 눌러 웹페이지 접속과 전화 걸기

- 인텐트(Intent)라는 것을 사용하면 “`http://...`”나 “`tel:...`”만으로도 웹 페이지 접속과 전화걸기가 가능



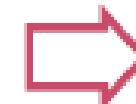


실제 단말 연결하기

드라이버 설치



케이블로 단말 연결



단말의 설정 바꾸기





단말기 제조사 사이트에서 드라이버 받기

- 삼성 단말기 https://local.sec.samsung.com/comLocal/support/down/kies_main.do?kind=usb
- LG 단말기 <https://www.lge.co.kr/lgekor/download-center/downloadCenterList.do>

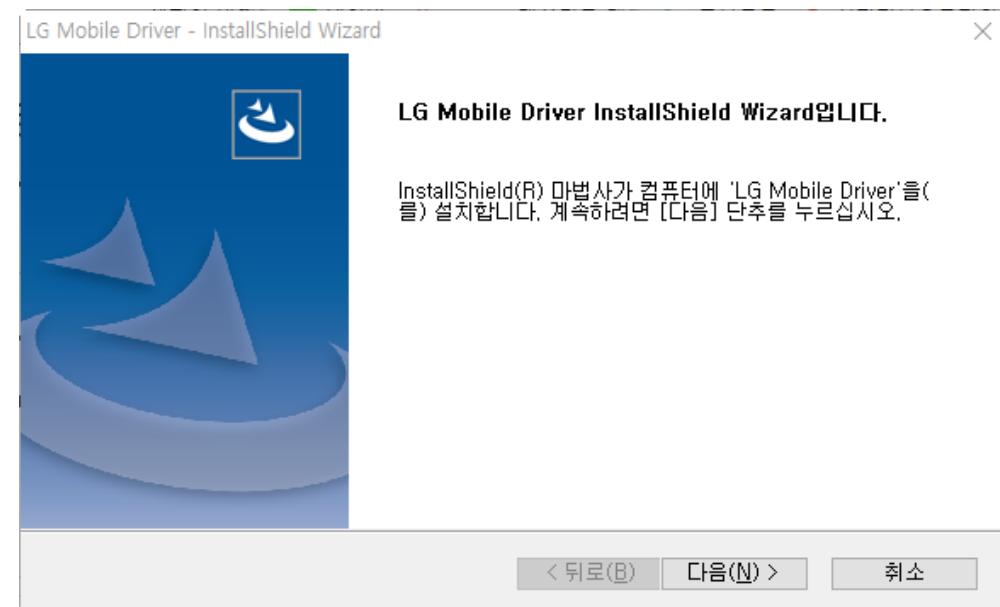
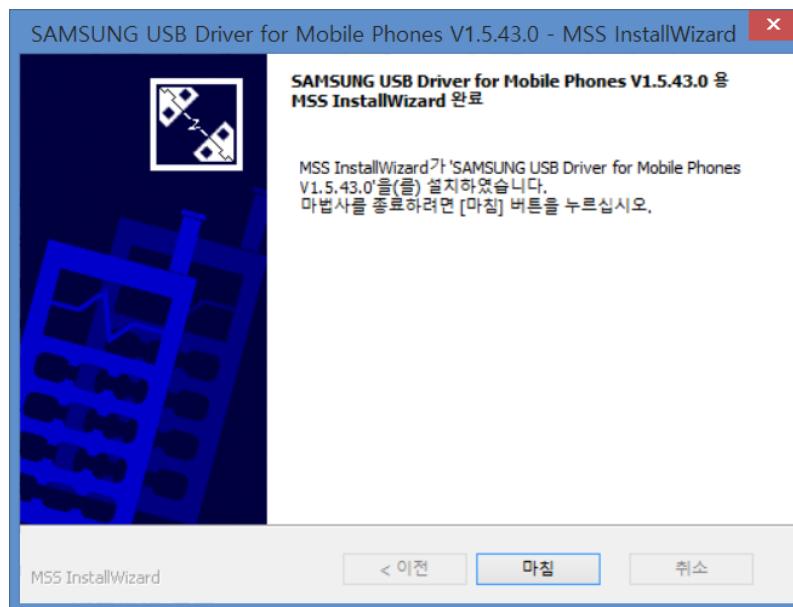
The screenshot shows a web browser window with the following details:

- Title Bar:** 통합 USB 드라이버 | 고객지원 | local.sec.samsung.com/comLocal/support/down/kies_main.do?kind=usb
- Header:** 모바일, TV, 가전, IT, 음향, **프로모션**, 큐레이션샵, SAMSUNG, EXPLORE, 고객지원, 비즈니스, 검색창
- Breadcrumbs:** Home / 고객지원 / 통합USB
- Section Title:** 통합 USB 드라이버
- Text:** 통합 USB 드라이버, 휴대폰관련 문의
- Image:** A blue icon representing the KIES software.
- Text:** NPS, Kies 설치시 기본적으로 함께 제공되며, 이에 따로 설치하실 필요는 없습니다.
별도 설치가 필요한 경우에만 이용하시기 바랍니다.
- Link:** 통합 USB 드라이버 >
- Section Title:** 통합 USB 드라이버 설치방법
- Text:** 자세한 내용은 통합 USB 드라이버 가이드를 클릭하여 확인 하실 수 있습니다.
- Text:** 1. 파일을 다운로드 하여 PC의 임의의 폴더에 저장합니다.
- Image:** A blue circular icon with a white 'beta' symbol.



드라이버 설치하기

• 일반 설치 과정과 같음





USB 케이블 연결

- 케이블 연결하면 장치 인식함
- 안드로이드 스튜디오 창에서 탐색기처럼 볼 수 있음

The screenshot shows the Android Studio interface with the following details:

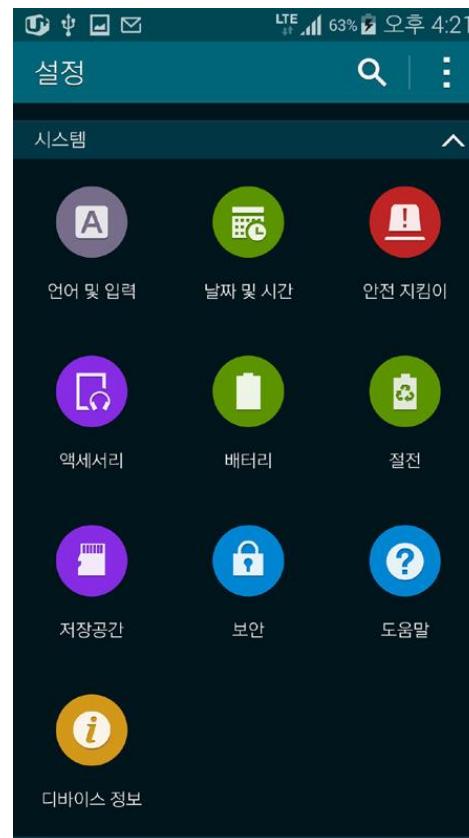
- Title Bar:** Hello [C:\Users\Mars\AndroidStudioProjects\Hello] - ...app\src\main\res\layout\activity_main.xml [app] - Android Studio
- Device File Explorer:** Shows the file structure of the emulator "Emulator Nexus_5X_API_28 Android 9, API 28".
- Activity Main XML Editor:** Displays the XML code for the activity_main.xml layout.
- Component Tree:** Shows the visual hierarchy of the layout, including a ConstraintLayout containing a button labeled "확인 1".
- Bottom Status Bar:** Includes Logcat, TODO, Terminal, Build, and Event Log tabs, along with a message: "daemon started successfully (moments ago)".

A large blue progress bar dialog is overlaid on the left side of the screen, indicating "장치 설치 중..." (Device installation in progress). It features a progress bar with a green segment and a message below it: "필요한 파일을 시스템에 설치하는 동안 잠시 기다려 주십시오. 몇 분 정도 걸릴 수 있습니다." (Please wait while files are installed in the system. This may take a few minutes.)



개발자 모드 설정

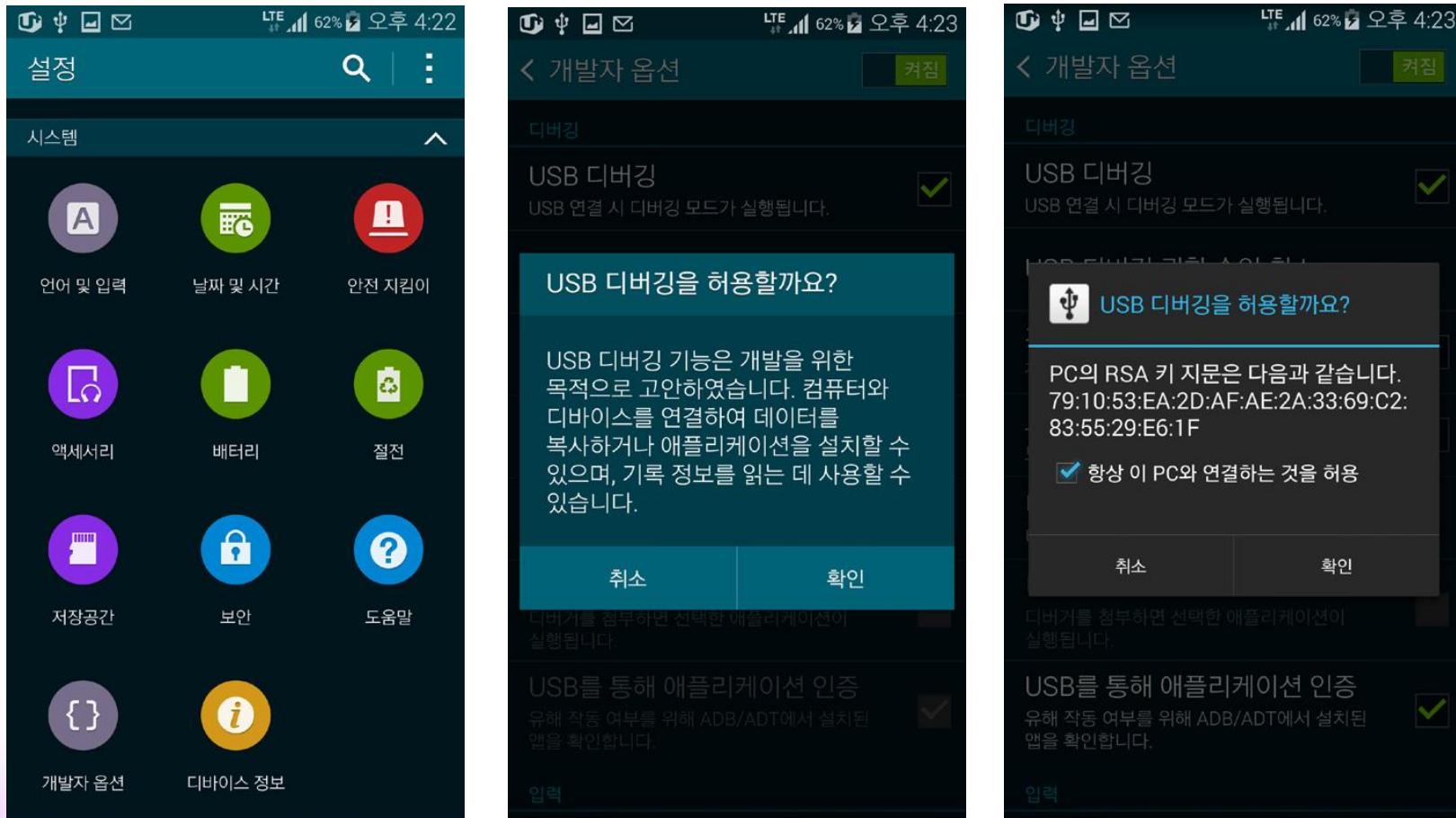
- 설정의 디바이스 정보에서 빌드 번호 여러 번 누르면 개발자 모드로 실행됨





개발자 모드 설정

• 개발자 옵션 메뉴가 보이면 해당 메뉴에서 USB 디버깅 허용 설정 변경함





안드로이드 스튜디오의 기본 사용법을 익히고 화면을 만들어봐요.



- 1 안드로이드 스튜디오 제대로 익히기
- 2 뷰와 뷰의 크기속성 이해하기
- 3 레이아웃 기초 익히기



안드로이드 스튜디오에 대해 먼저 알고 싶나요?

- 안드로이드 스튜디오 살펴보기



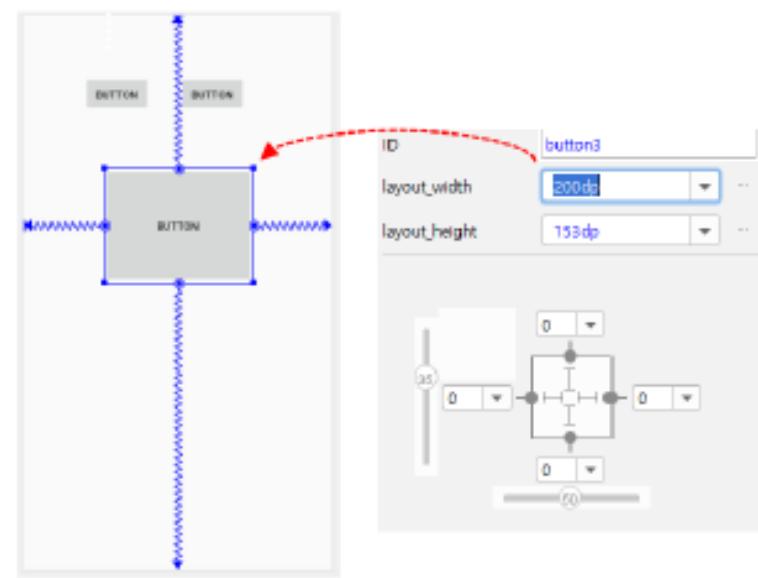
뷰란 무엇일까요?

- 뷰와 뷰의 크기 속성 이해하기



처음 보이는 레이아웃으로 화면 배치를 해볼까요?

- 제약 레이아웃 익히기



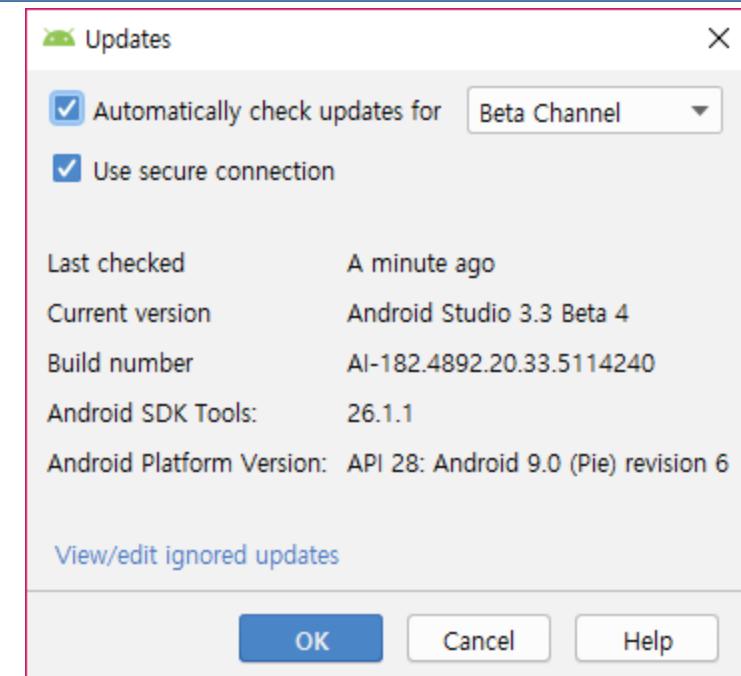
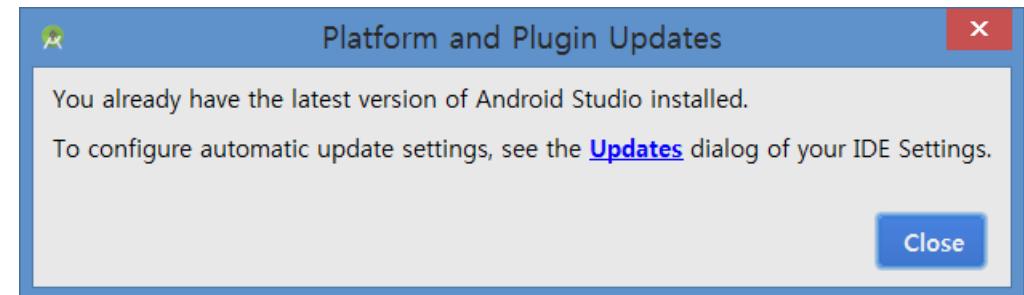
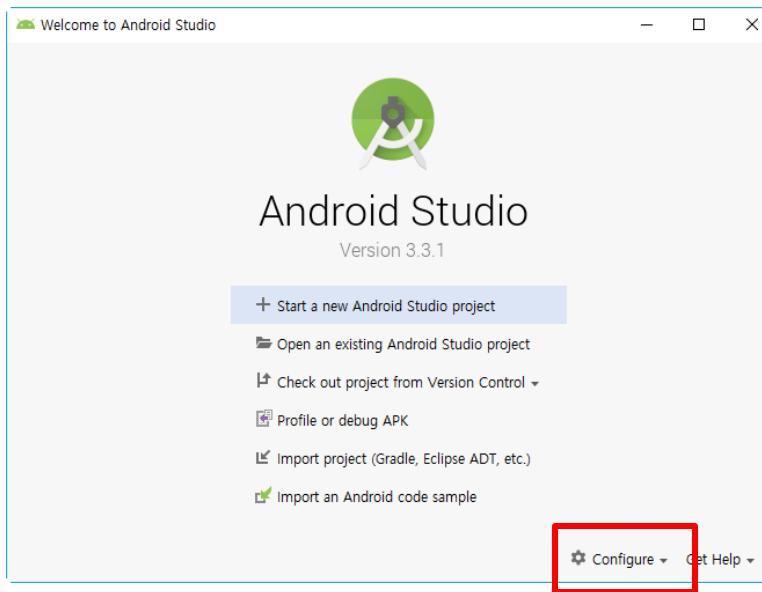


안드로이드 스튜디오 살펴보기



시작 화면과 업데이트 여부 확인

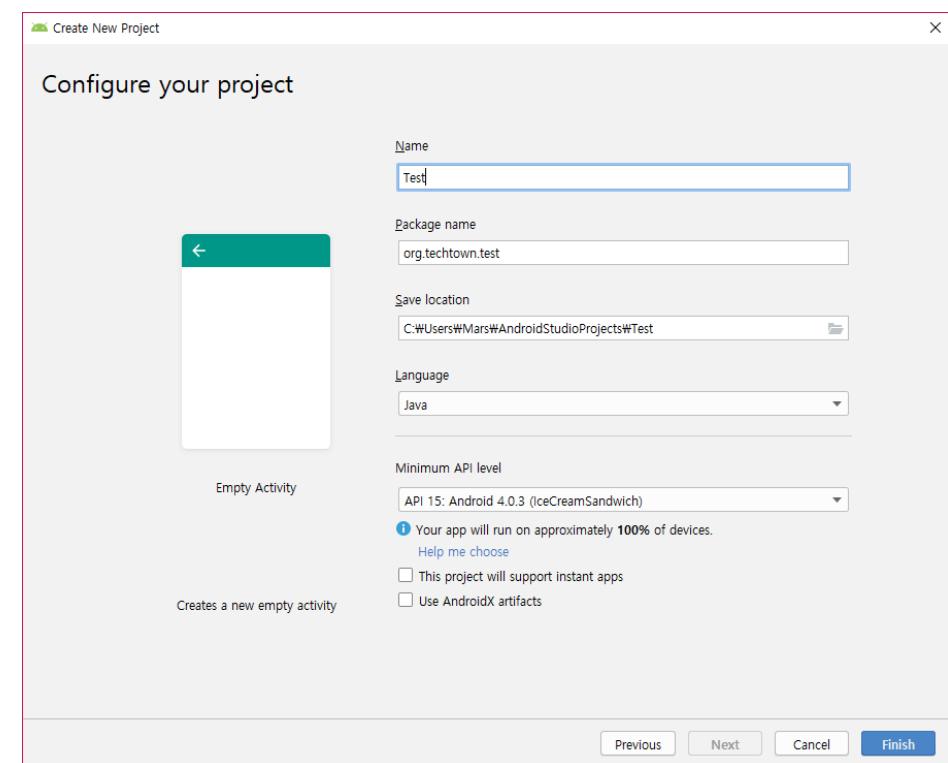
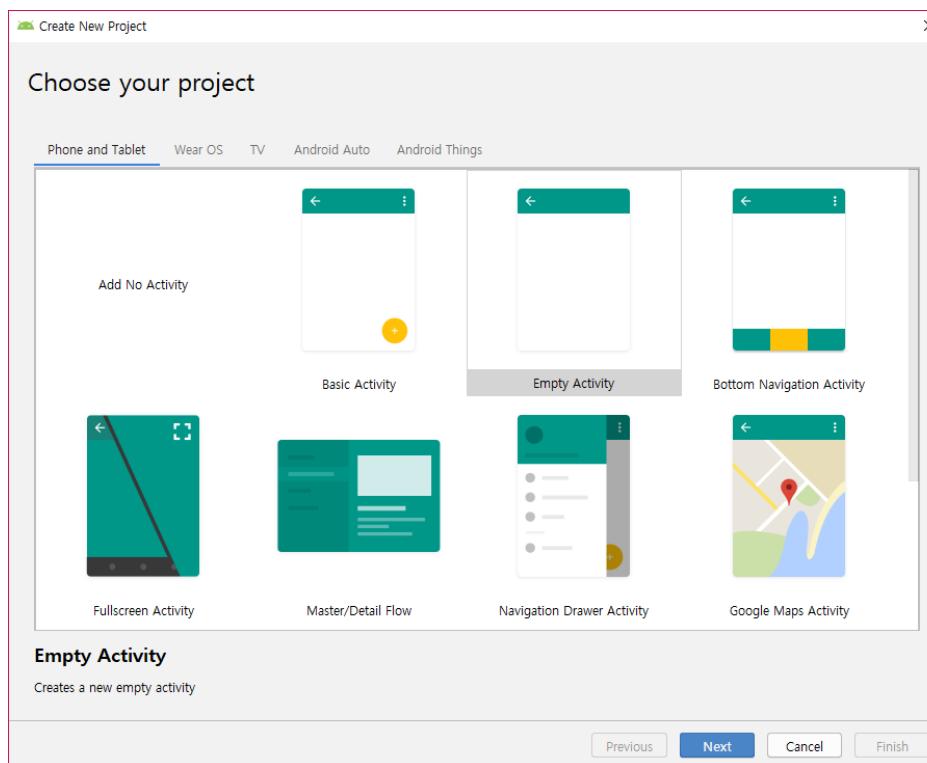
- 시작 화면 하단의 링크를 이용해 업데이트 여부 확인





새로운 프로젝트 생성

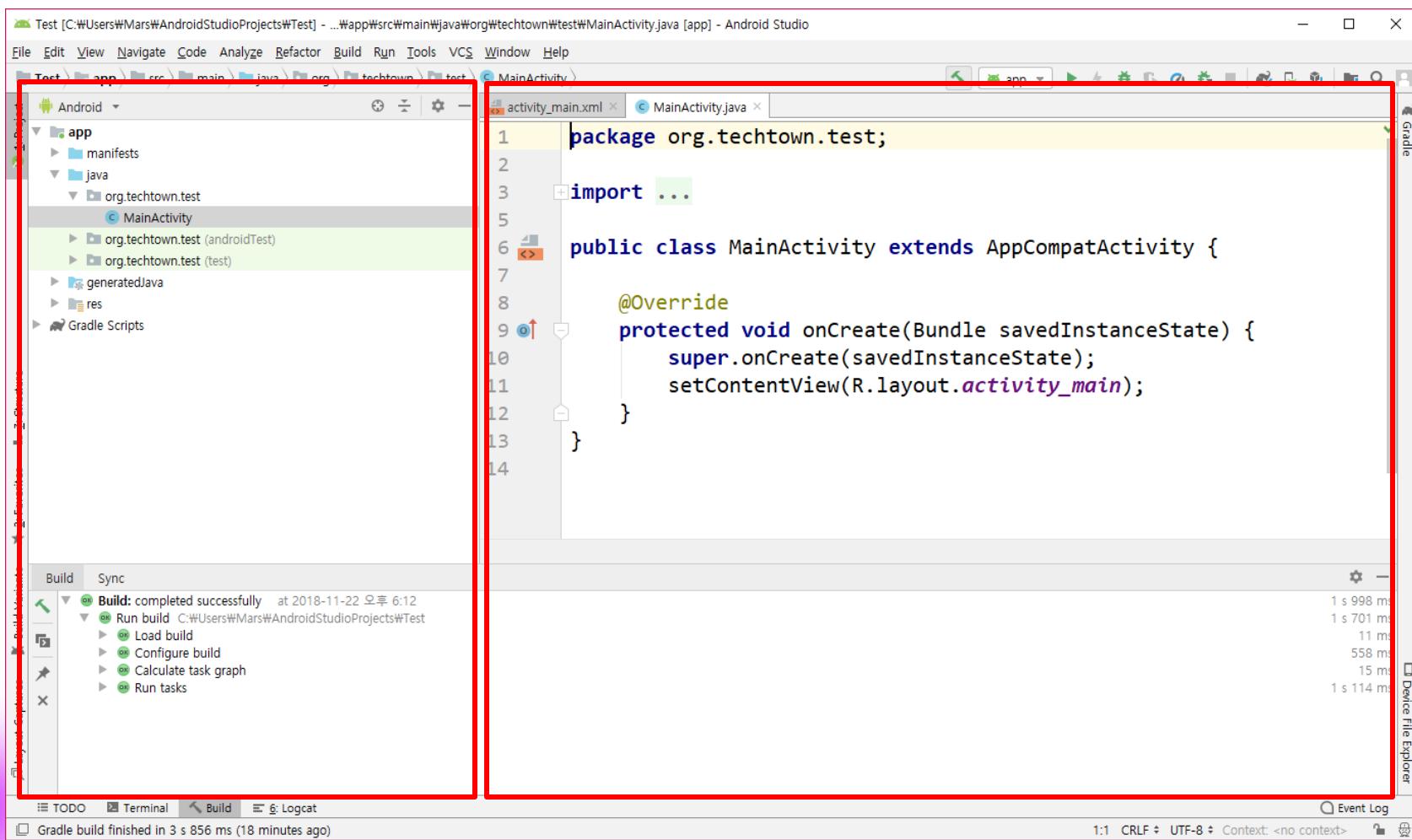
- Test라는 이름의 새로운 프로젝트 생성





프로젝트 창의 구성과 기능

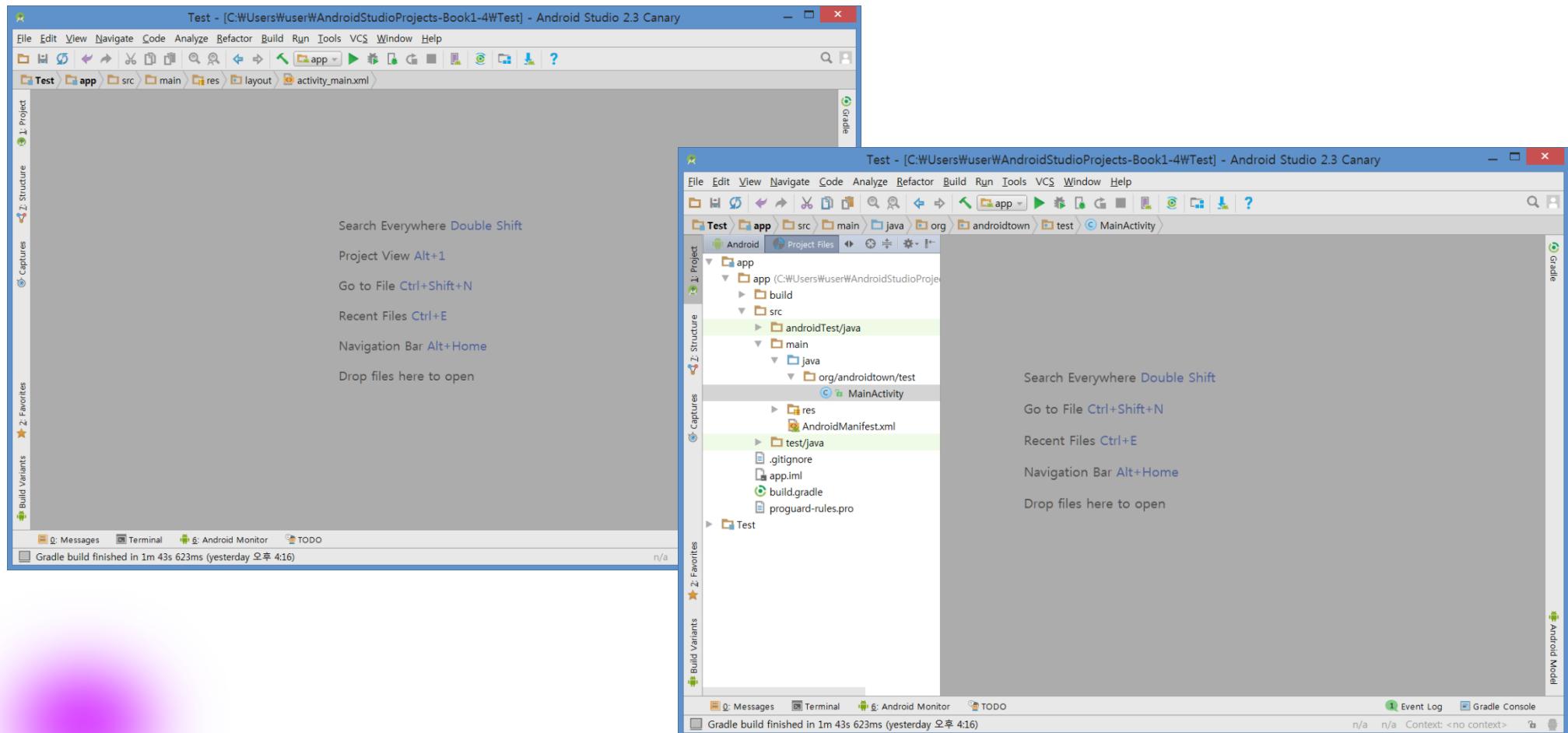
- 프로젝트 영역과 작업 영역





창 감추거나 보이기

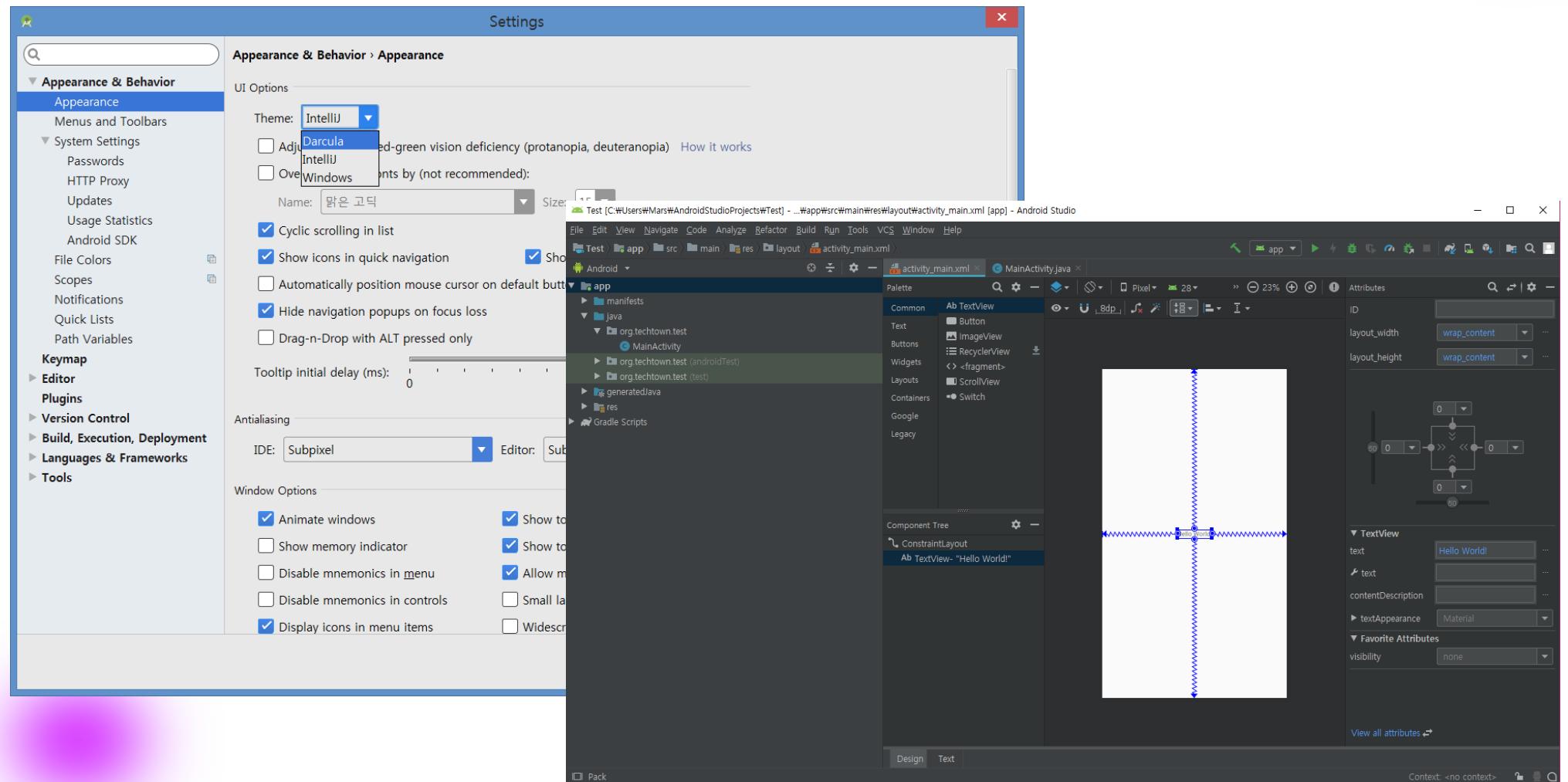
- 창을 감추려면 창 상단의 아이콘 클릭하면 되고 다시 보이게 하려면 벽면의 텁 클릭





테마 변경하기

- 어두운 테마로 변경하고 싶은 경우





코드 편집기

- 코드에서 발생한 에러 표시

The screenshot shows the Android Studio interface with the code editor open. The project navigation bar at the top shows 'androidtown' and 'test' under 'src', and 'MainActivity' selected. Below it, the code editor shows 'MainActivity.java'. The code is:

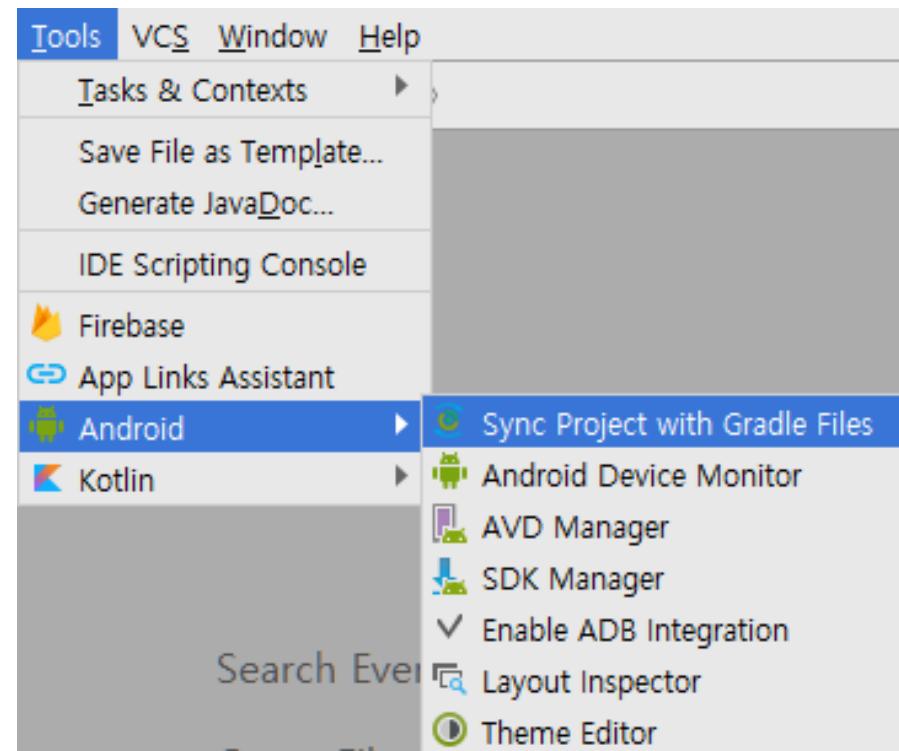
```
1 package org.androidtown.test;
2
3 import ...
5
6 public class MainActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13 }
```

Line 6 has a red underline and a yellow lightbulb icon, with a tooltip 'Unexpected token' pointing to the opening brace. Line 13 has a red underline and a small blue arrow icon.



파일 변경사항 반영

- Gradle을 빌드 도구로 사용하며 [Tools>Android>Sync ...] 메뉴를 누르면 파일 변경사항 반영하고 재빌드





코드 편집기

- 클래스 또는 메소드의 문서 참조 → Ctrl+Q

The screenshot shows the Android Studio interface. On the left, the code editor displays `MainActivity.java` with the following content:

```
1 package org.androidtown.test;
2
3 import ...
4
5
6 public class MainActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13 }
14 }
```

The cursor is positioned at the end of the `AppCompatActivity` class name. A tooltip window titled "Documentation for AppCompatActivity" appears, providing the following information:

Documentation for AppCompatActivity
appcompat-v7-23.1.1
android.support.v7.app
public class AppCompatActivity
extends android.support.v4.app.FragmentActivity
implements android.support.v7.app.AppCompatCallback, android.support.v4.app.TaskStackBuilder
Base class for activities that use the support library action bar features.
You can add an ActionBar to your activity when running on API level 7 or higher by extending



코드 편집기

- 후보 클래스와 파라미터의 자동 표시 → Ctrl+P

The screenshot shows two instances of the Java code editor in Android Studio. In the left window, the cursor is at the end of the word 'String' in the line 'String hello = new String();'. A completion dropdown menu is open, listing various constructors for the 'String' class. In the right window, the cursor is at the end of the word 'String' in the same line, and another completion dropdown menu is open, listing different constructor signatures for 'String'.

Left Window (Completion for 'String'):

```
1 package org.androidtown.test;
2
3 import ...
4
5
6 public class MainActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12
13         String
14             StringIndexOutOfBoundsException...
15             StrictMode (android.os)
16     }
17 }
```

Right Window (Completion for 'String hello = new String();'):

```
1 package org.androidtown.test;
2
3 import ...
4
5
6 public class MainActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12
13         String hello = new String();
14
15     }
16
17 }
```

Completion dropdown for 'String' (Left):

- <no parameters>
- byte[] data
- byte[] data, int high
- byte[] data, int offset, int byteCount
- byte[] data, int high, int offset, int byteCount
- byte[] data, int offset, int byteCount, String charsetName
- byte[] data, String charsetName
- byte[] data, int offset, int byteCount, Charset charset
- byte[] data, Charset charset
- char[] data
- char[] data, int offset, int charCount
- String toCopy
- StringBuffer stringBuffer

Completion dropdown for 'String hello = new String();' (Right):

- <no parameters>
- byte[] data
- byte[] data, int high
- byte[] data, int offset, int byteCount
- byte[] data, int high, int offset, int byteCount
- byte[] data, int offset, int byteCount, String charsetName
- byte[] data, String charsetName
- byte[] data, int offset, int byteCount, Charset charset
- byte[] data, Charset charset
- char[] data
- char[] data, int offset, int charCount
- String toCopy
- StringBuffer stringBuffer

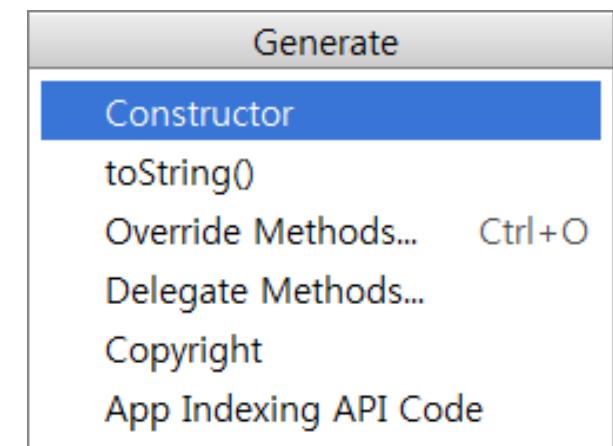


코드 편집기

- 부모 클래스의 메소드 재정의 → Generate... > Override Methods...

```
1 package org.androidtown.test;
2
3 import ...
4
5
6 public class MainActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13 }
14 }
```

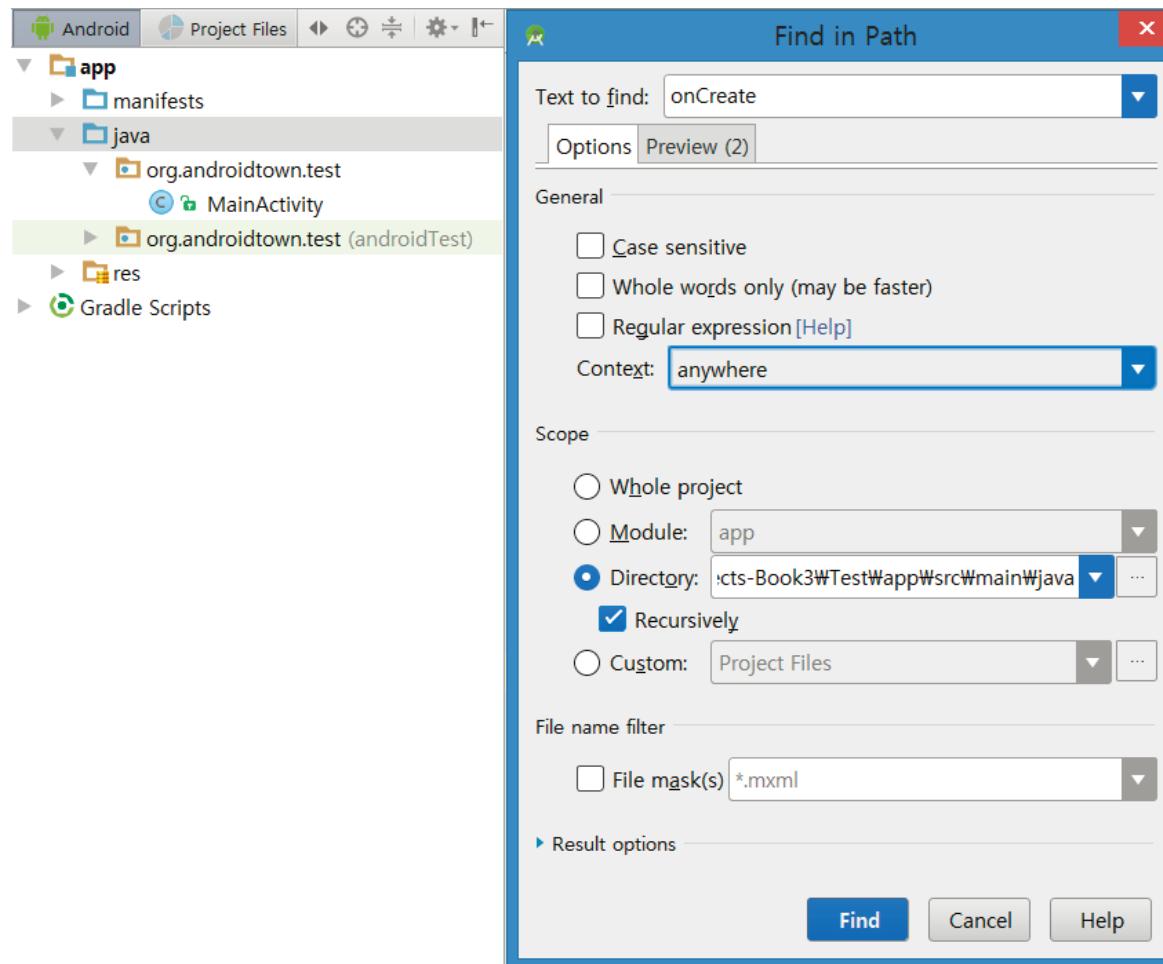
The screenshot shows the code editor for MainActivity.java. A context menu is open at the end of the closing brace of the onCreate method. The 'Generate...' option is highlighted in blue, indicating it is the selected action. Other options in the menu include Copy Reference, Paste, Paste from History..., Paste Simple, Column Selection Mode, Find Usages, Refactor, Folding, Analyze, Go To, Run 'MainActivity', Debug 'MainActivity', and Create 'MainActivity'.





코드 편집기

- 파일의 내용 검색 – Find in Path





- Design 탭과 Text 탭

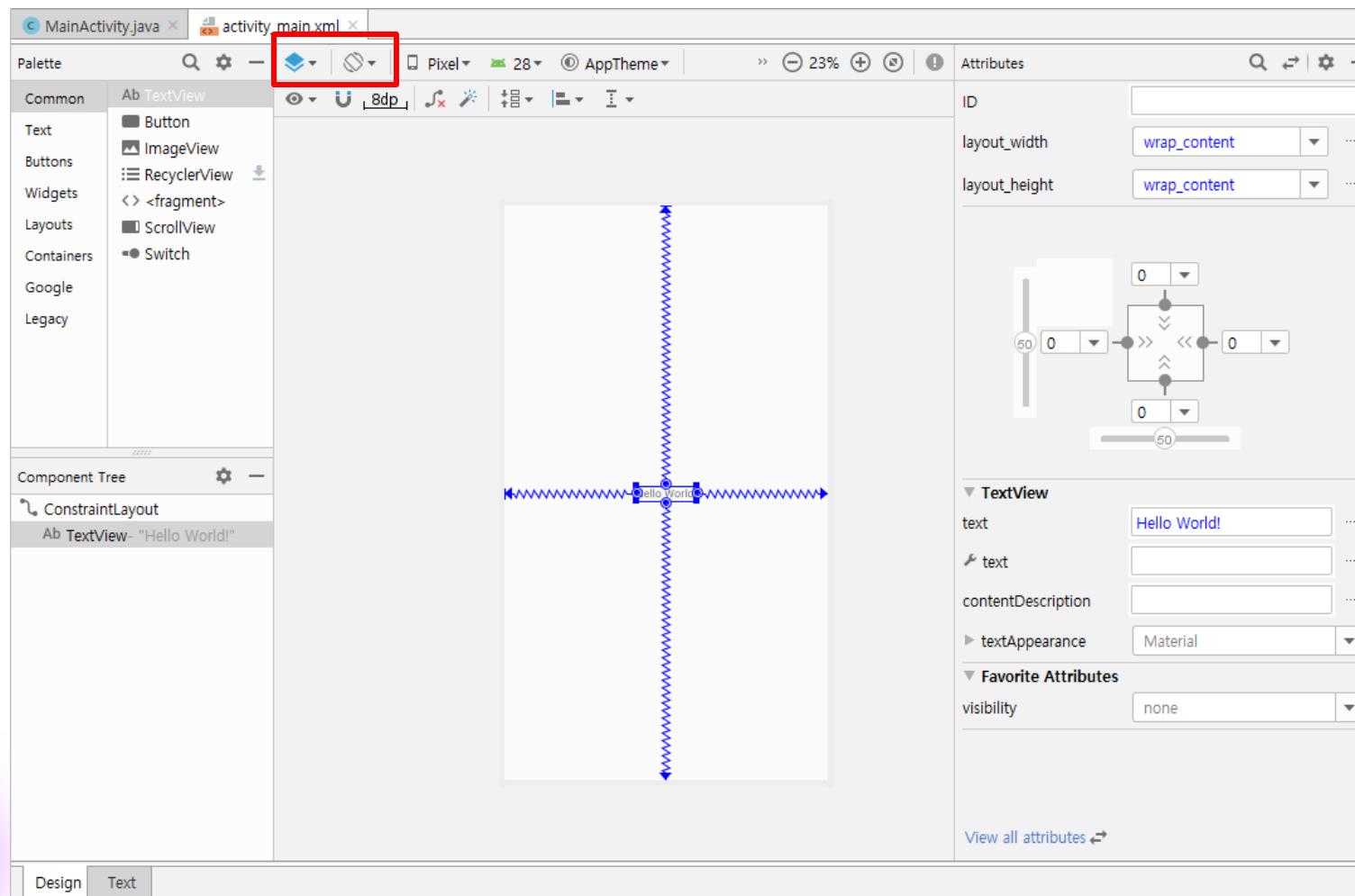
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="org.techtown.test.MainActivity">

    <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" />

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

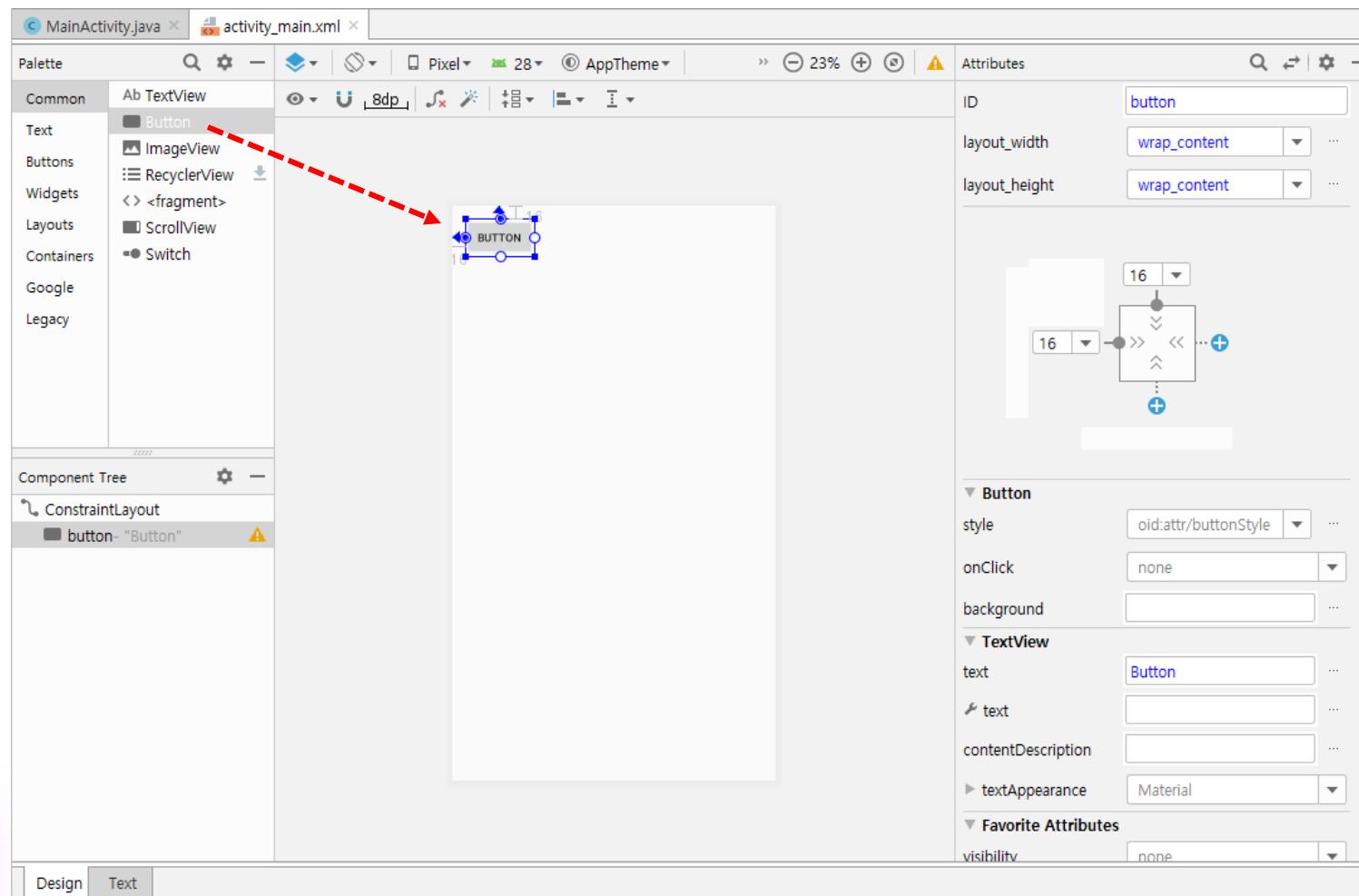


- 디자인 화면에서 [Show Design] 아이콘 선택





- 빨렛에서 위젯 끌어다 놓기와 위젯의 속성 설정하기

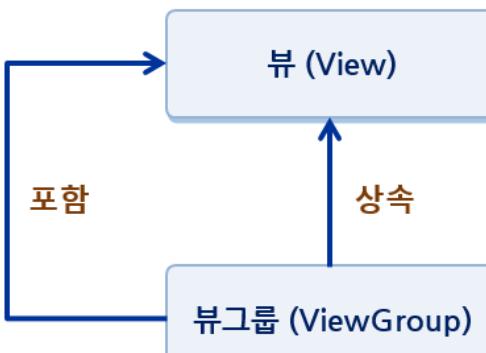




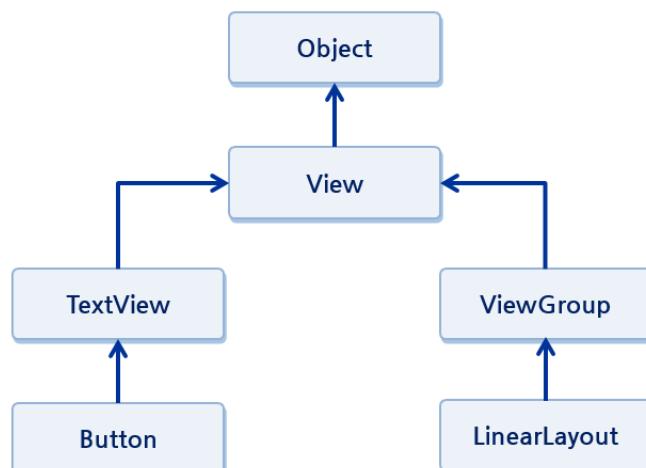
뷰와 뷰의 크기속성 이해하기



뷰와 뷰그룹의 정의



[뷰와 뷰 그룹의 관계]



[버튼과 리니어 레이아웃의 계층도]

• 뷰(View)

- 화면에 보이는 각각의 것들 (버튼, 텍스트 등등)
- 흔히 컨트롤(Control)이나 위젯(Widget)이라 불리는 UI 구성 요소

• 뷰 그룹(View Group)

- 뷰들을 여러 개 포함하고 있는 것
- 뷰 그룹도 뷰에서 상속하여 뷰가 됨. 즉, 위의 뷰는 버튼, 텍스트 뿐만 아니라 이것들을 포함하는 눈에 보이지 않는 영역을 포함함

• 위젯(Widget)

- 뷰 중에서 일반적인 컨트롤의 역할을 하고 있는 것
- 버튼, 텍스트 등등

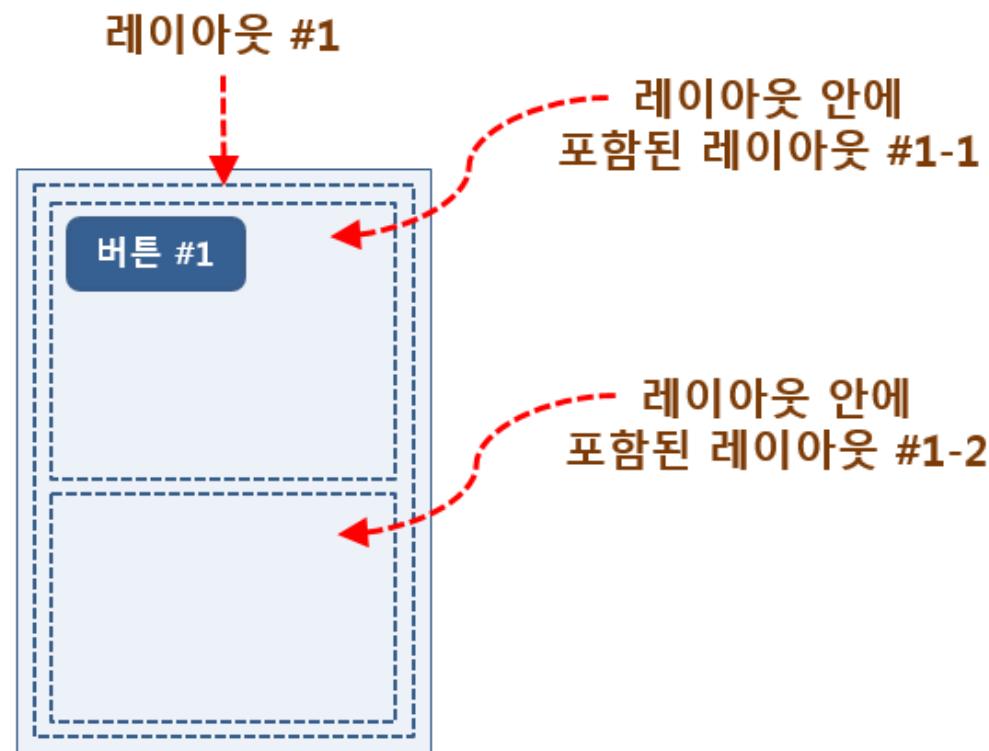
• 레이아웃(Layout)

- 뷰 그룹 중에서 내부에 뷰들을 포함하고 있으면서 그것들을 배치하는 역할을 하는 것



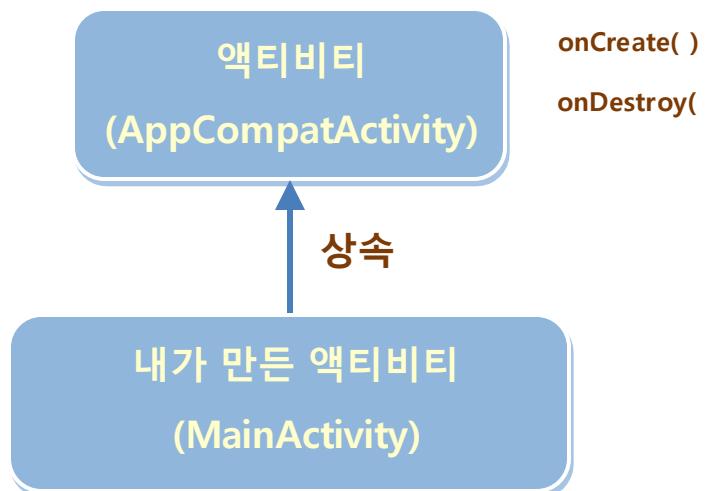
레이아웃 안에 레이아웃 포함시키기

- 레이아웃도 뷰이기 때문에 레이아웃 안에 레이아웃을 포함시킬 수 있음





상속에 대해 잘 몰라요!



• 상속

- 객체지향의 가장 기본적인 개념 중 하나
- 부모의 특성을 그대로 물려받는 것으로 변수나 메소드 재사용 가능

• 액티비티의 상속

- 처음 만들어 본 액티비티에서 extends 키워드 사용

```
public class MainActivity extends AppCompatActivity
```

• 부모 클래스의 메소드를 재정의

- onCreate() 메소드는 이미 부모 클래스에 정의되어 있음
- 기능을 추가하고 싶을 때 재정의(Override)

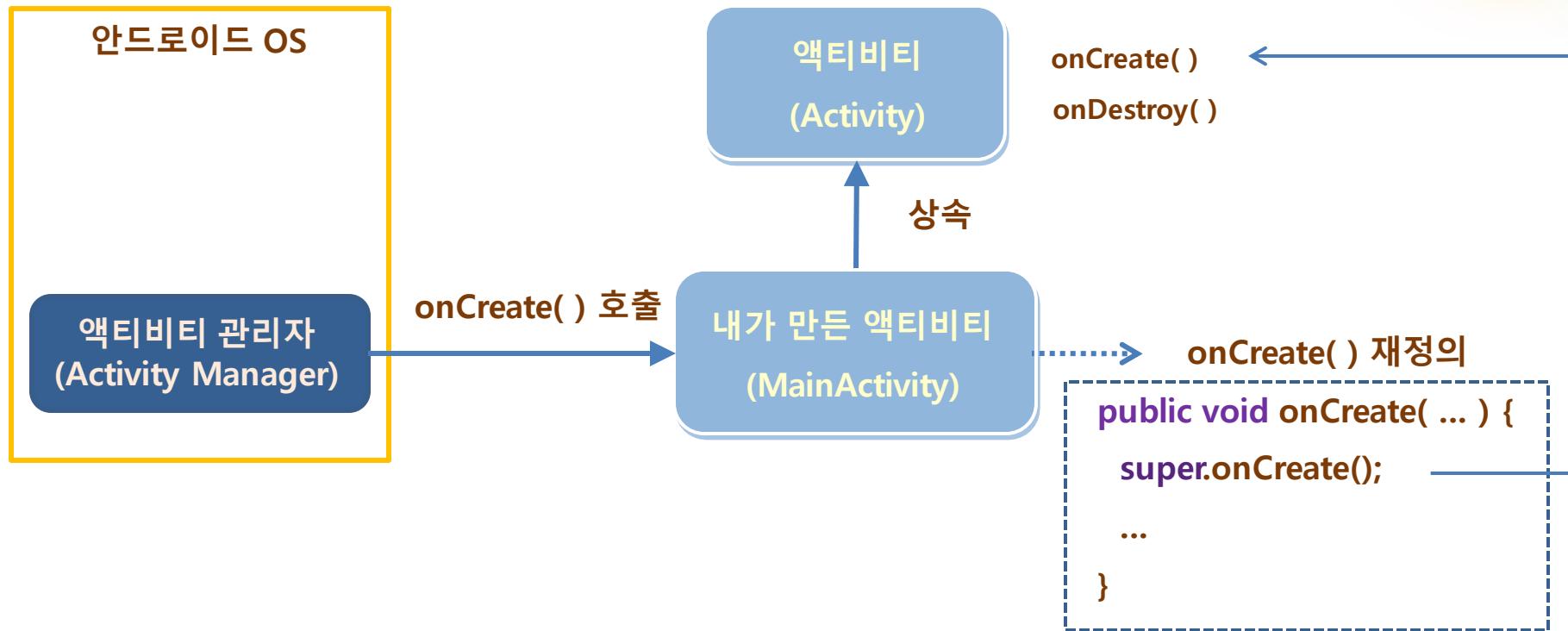
• this와 super

- 나 자신은 this, 부모는 super 를 사용하여 변수나 메소드 참조

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



재정의한 메소드를 시스템에서 호출하는 방법



- **액티비티가 만들어질 때 onCreate() 호출**

- 내가 만든 액티비티의 onCreate() 메소드를 호출
- super.onCreate() 를 호출하여 Activity 클래스에 정의된 onCreate() 메소드의 기능을 사용
- 그 아래에 추가적으로 필요한 기능을 코드로 추가



XML 레이아웃의 구성

- 뷰 태그와 속성으로 구성됨

```
<시작 태그  
    속성1="속성값1"  
    속성2="속성값2"  
    ...  
</끝 태그>
```

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Hello World!" />
```

```
<android.support.constraint.ConstraintLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    ...  
</android.support.constraint.ConstraintLayout>
```

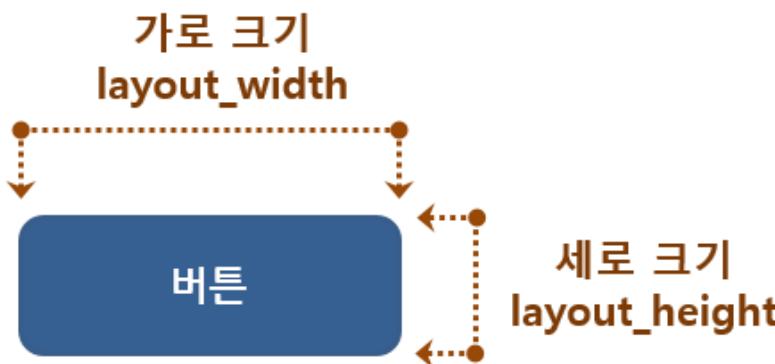
속성Prefix : 속성명1="속성값1"

```
    android:layout_width="wrap_content"  
    app:layout_constraintLeft_toLeftOf="parent"
```



가로 크기와 세로 크기 속성

- 가로 크기와 세로 크기는 뷰의 필수 속성
- 속성 제거 시 에러 표시



```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:app="http://schemas.android.com/apk
    xmlns:tools="http://schemas.android.com/t
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:marginStart="16dp"
        android:marginLeft="16dp"
        android:marginTop="16dp"
        android:text="Button"
        app:layout_constraintStart_toStartOf=
        app:layout_constraintTop_toTopOf="par

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



크기 지정 방법

• [필수] layout_width, layout_height

- 가장 기본적이면서 필수 속성으로 뷰의 폭과 높이를 설정함

(1) match_parent

뷰그룹에 남아 있는 여유 공간을 채움

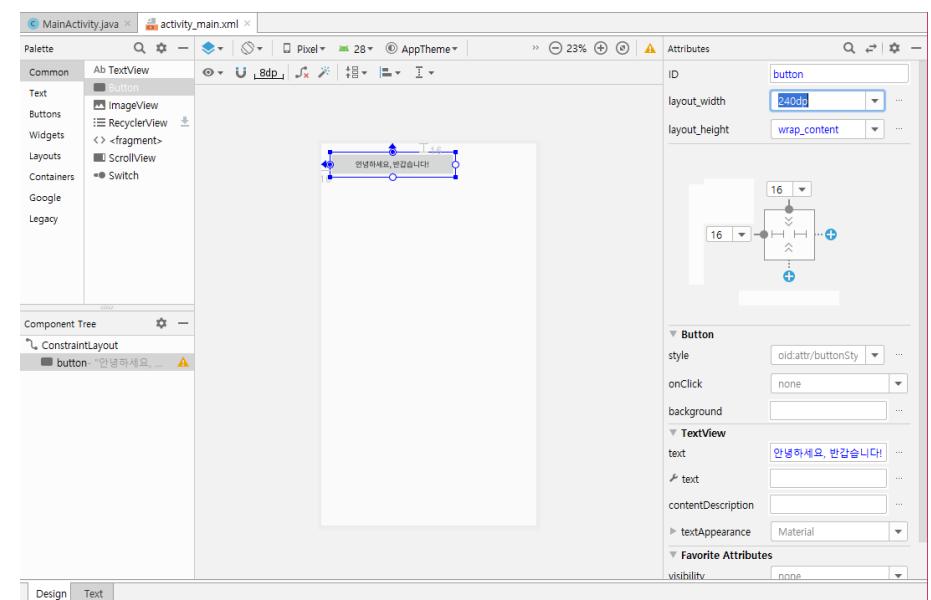
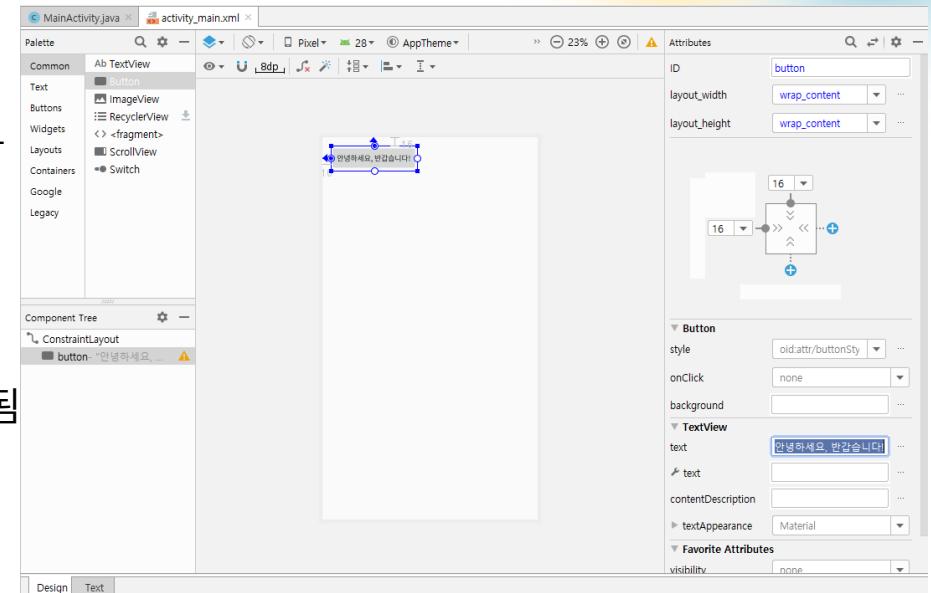
(2) wrap_content

뷰에 들어 있는 내용물의 크기에 따라 뷰의 크기가 결정됨

(3) 크기 값 지정

크기를 고정된 값으로 직접 지정하고 싶을 때 사용함

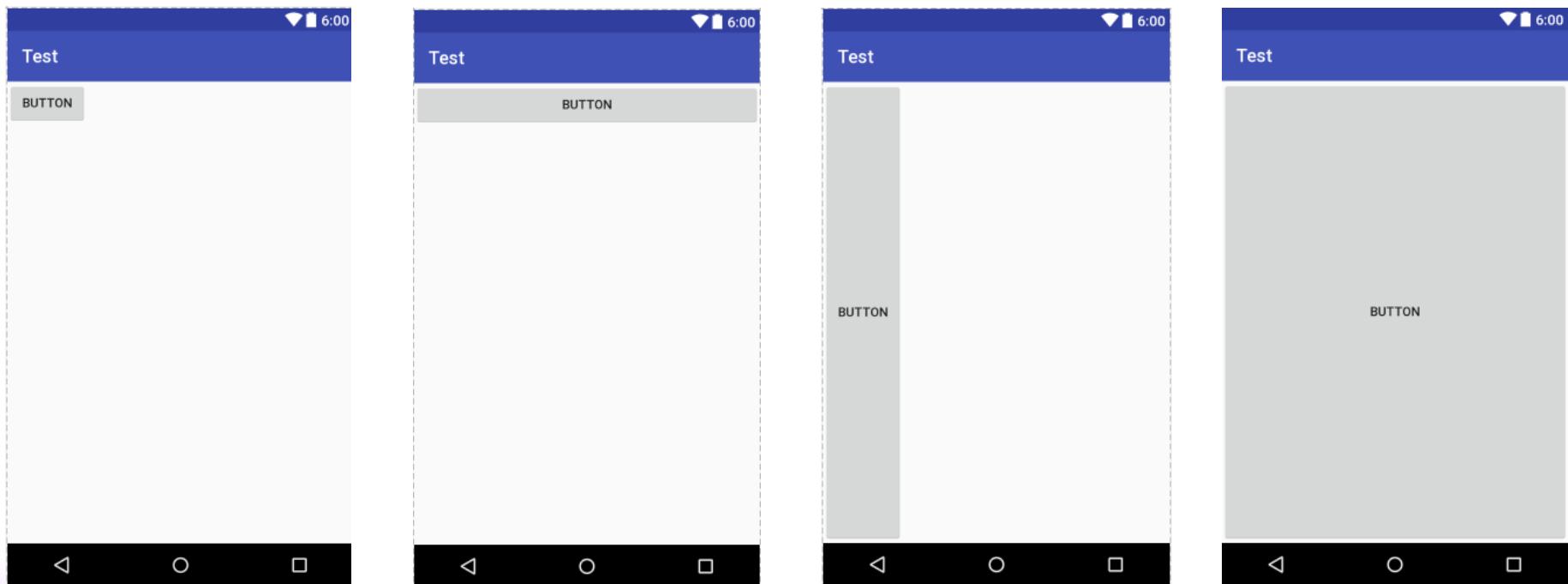
ex) "100px", "200dp"





최상위 레이아웃을 다른 것으로 변경한 경우

- `match_parent`와 `wrap_content` 적용 상태를 확실하게 구분할 수 있음





레이아웃 기초 익히기

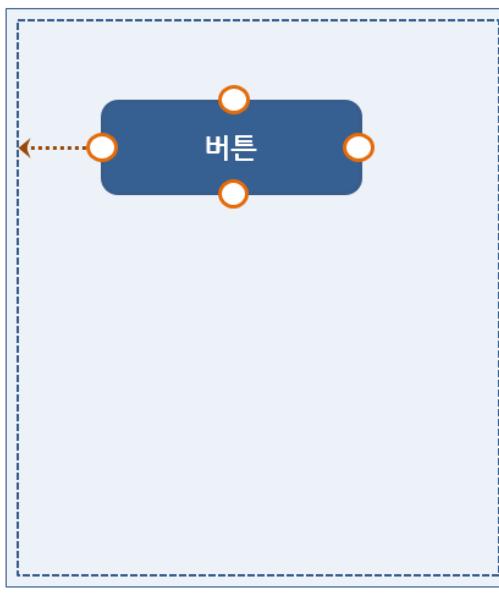


연결선으로 만드는 제약조건

- 뷰의 위, 아래, 왼쪽, 오른쪽의 연결점을 부모 레이아웃의 벽면과 연결하면 제약조건 생성
- 같은 레이아웃 안에 들어있는 다른 뷰와 연결 가능

제약조건(Constraint)

버튼의 왼쪽을 부모 레이아웃과 연결해 주세요.



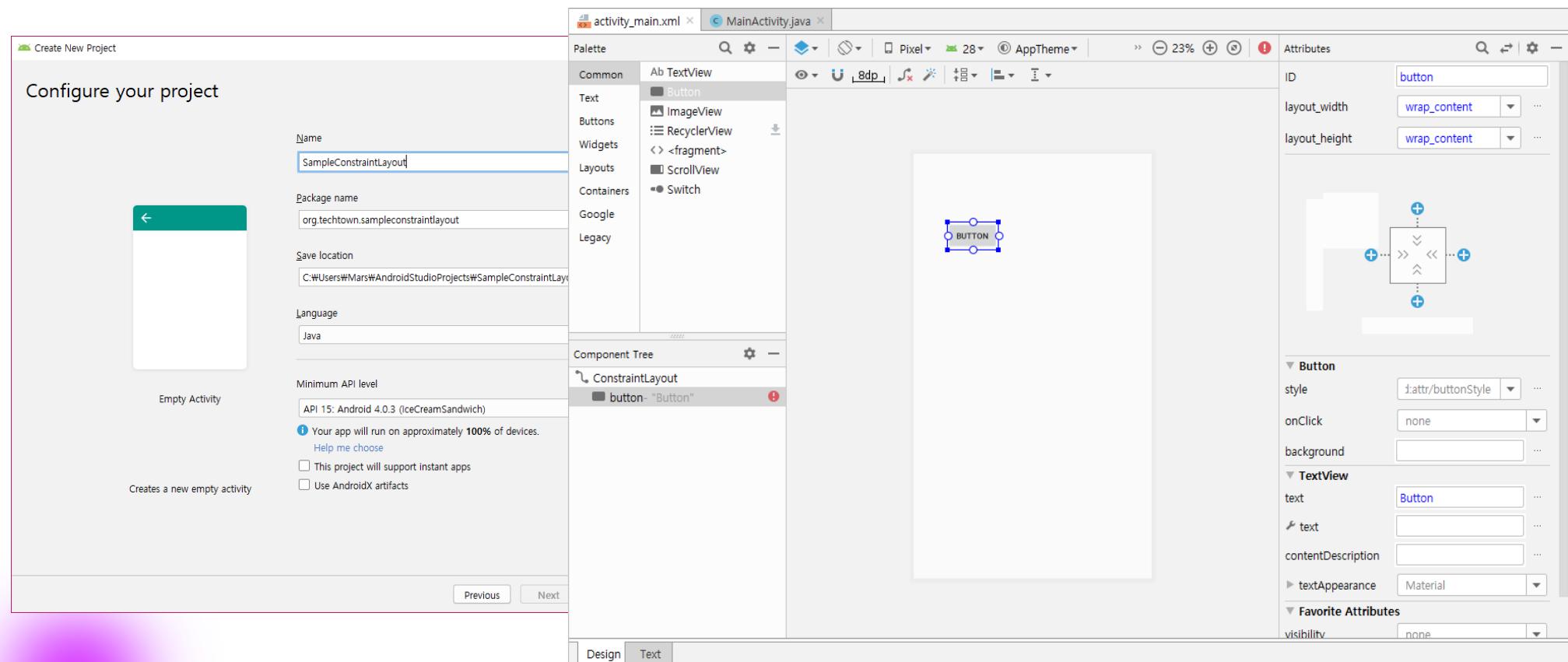
- 연결점을 연결할 수 있는 타깃

- 같은 레이아웃 안에 들어 있는 다른 뷰의 연결점
- 부모 레이아웃의 연결점
- 가이드라인(Guideline)



새로운 프로젝트 생성

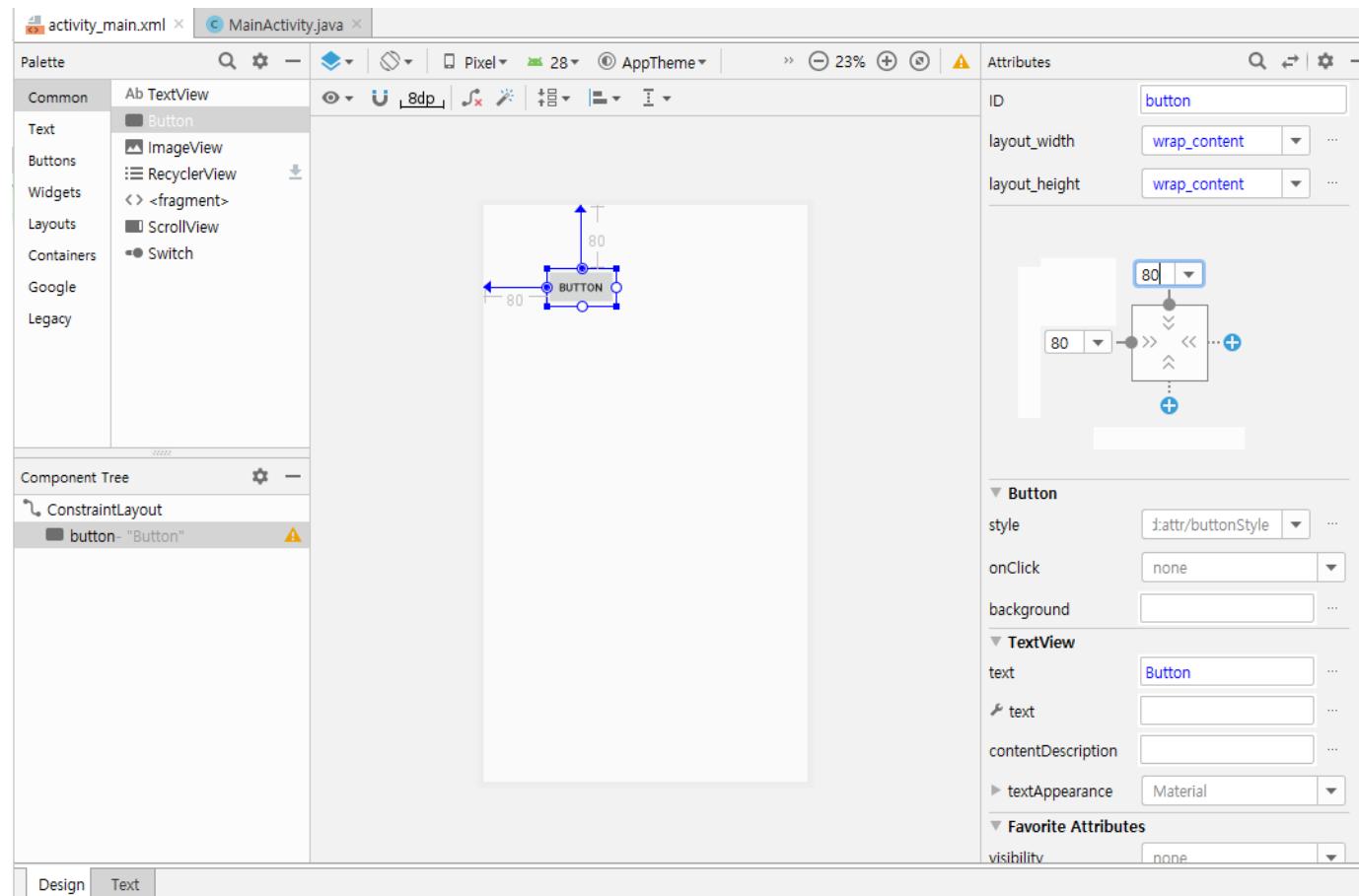
- SampleConstraintLayout이라는 이름으로 새로운 프로젝트 생성
- 화면 왼쪽 윗부분에 버튼 추가





연결선 생성

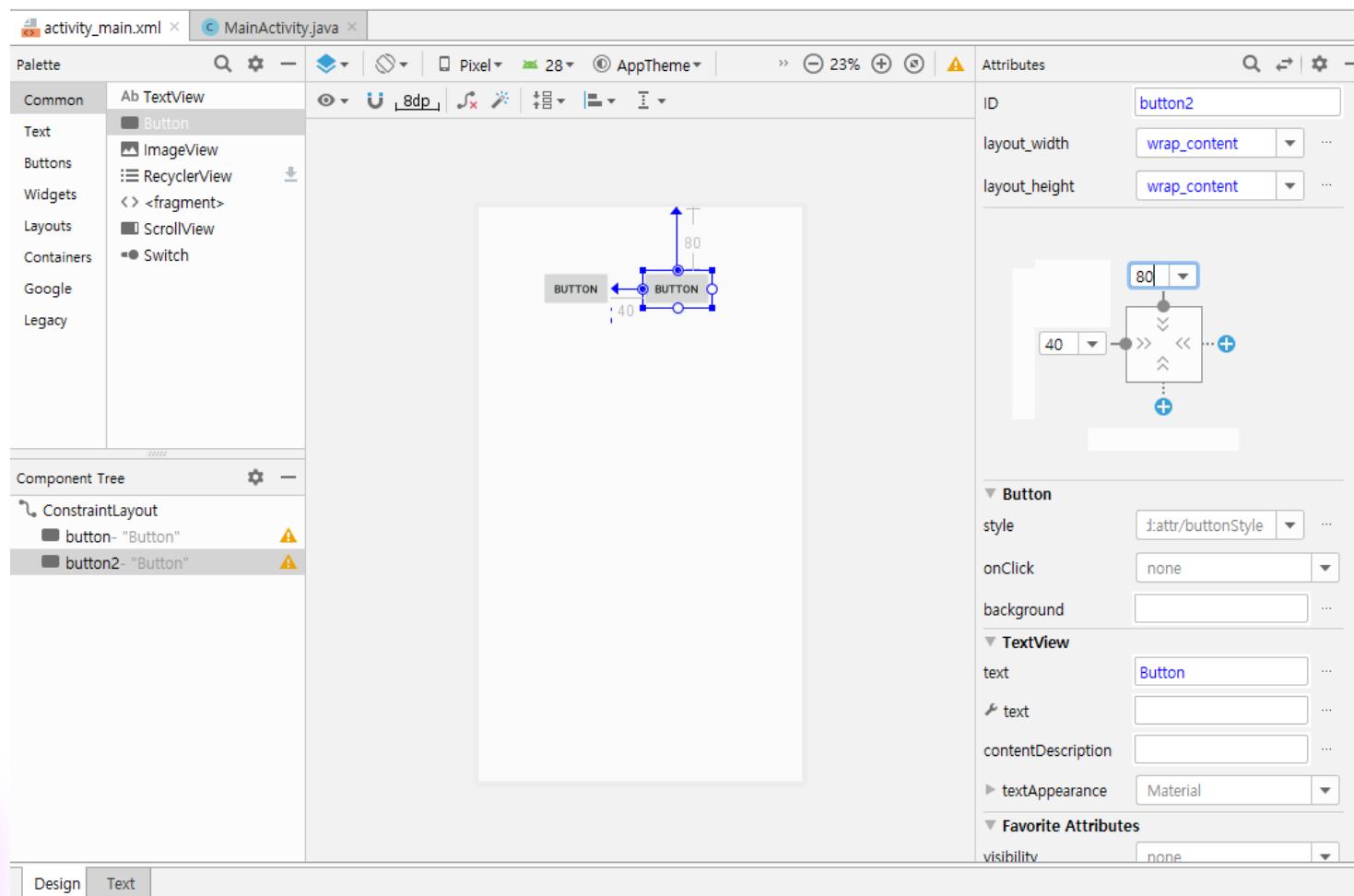
- 왼쪽과 위쪽에 있는 연결점을 부모 레이아웃의 벽면과 연결





버튼 하나 더 추가하고 연결선 생성

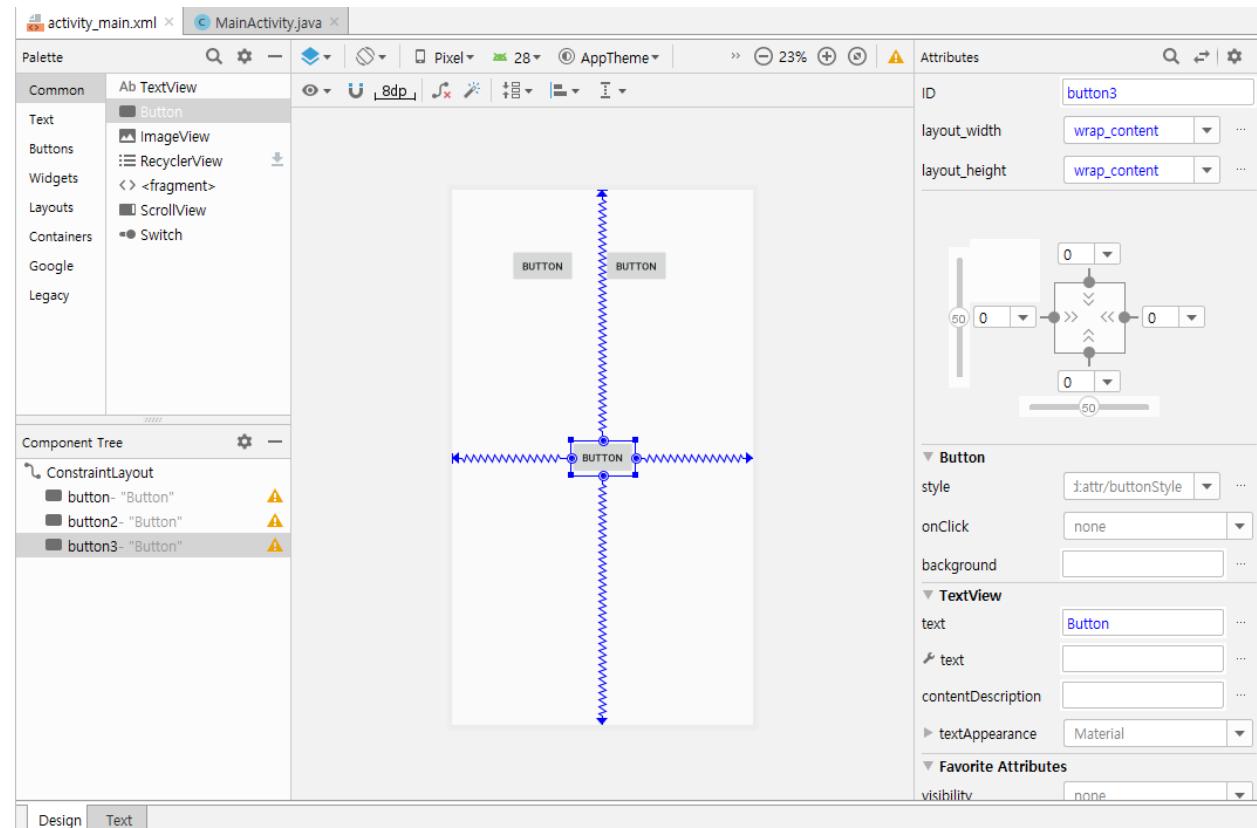
- 버튼 추가 후 부모 레이아웃 및 기존 버튼과 연결선 생성





화면 가운데에 뷰 배치하기

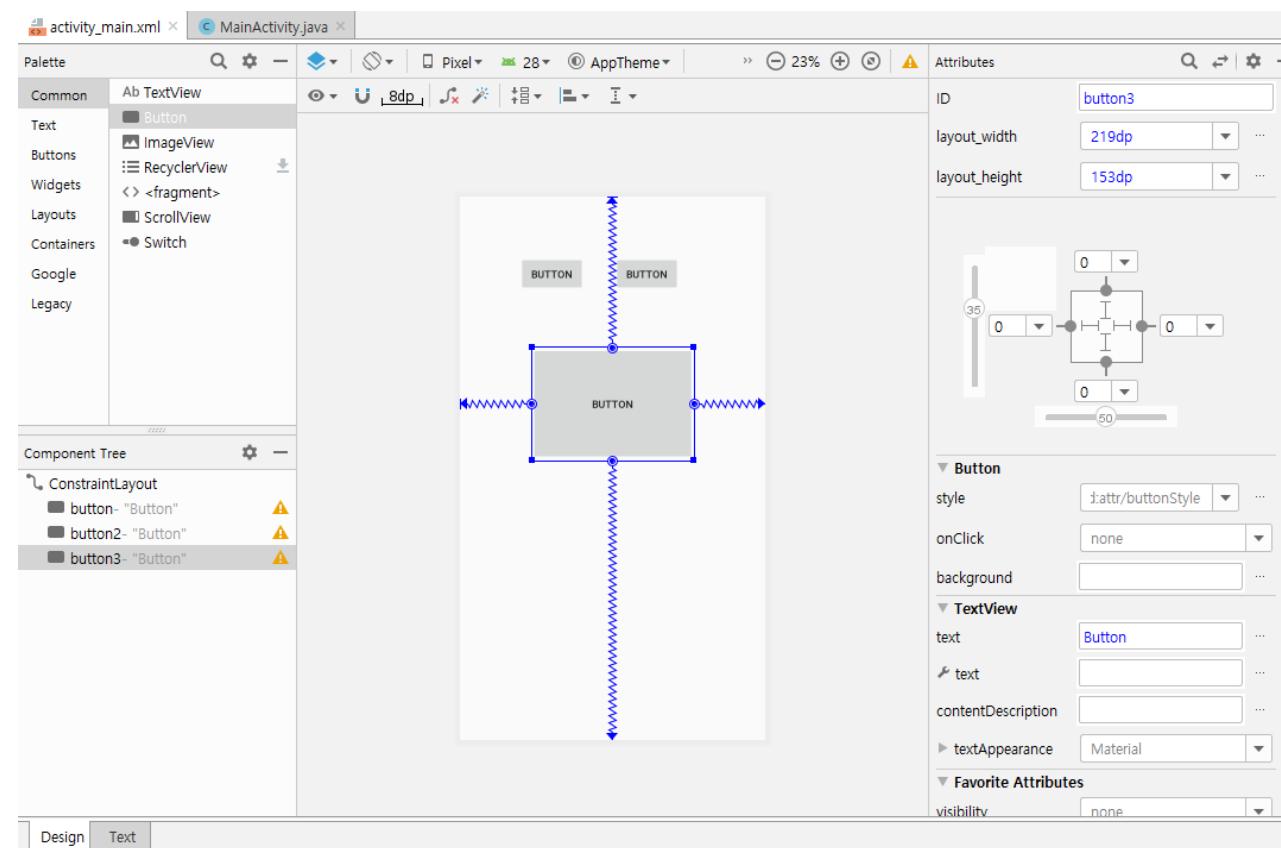
- 버튼 추가 후 위, 아래, 왼쪽, 오른쪽 모두 부모 레이아웃과 연결선 생성
- 좌우 또는 위, 아래를 쌍으로 연결하면 그 가운데에 배치됨





바이어스 사용하기

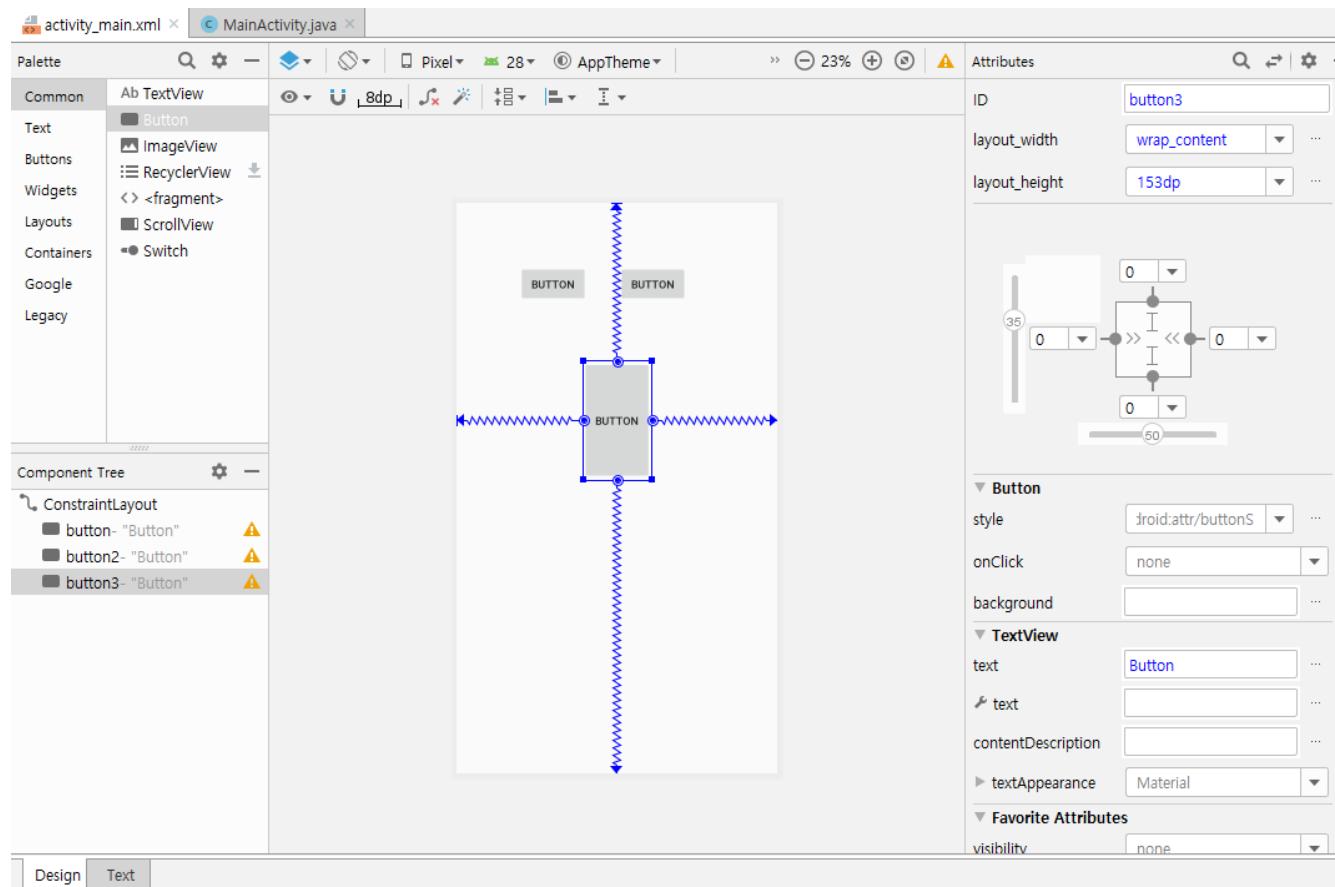
- 양쪽 또는 상하로 연결된 경우 오른쪽 제약조건 미리보기 창에서 바이어스(Bias)를 이용해 위치 조정 가능





뷰의 크기 조정

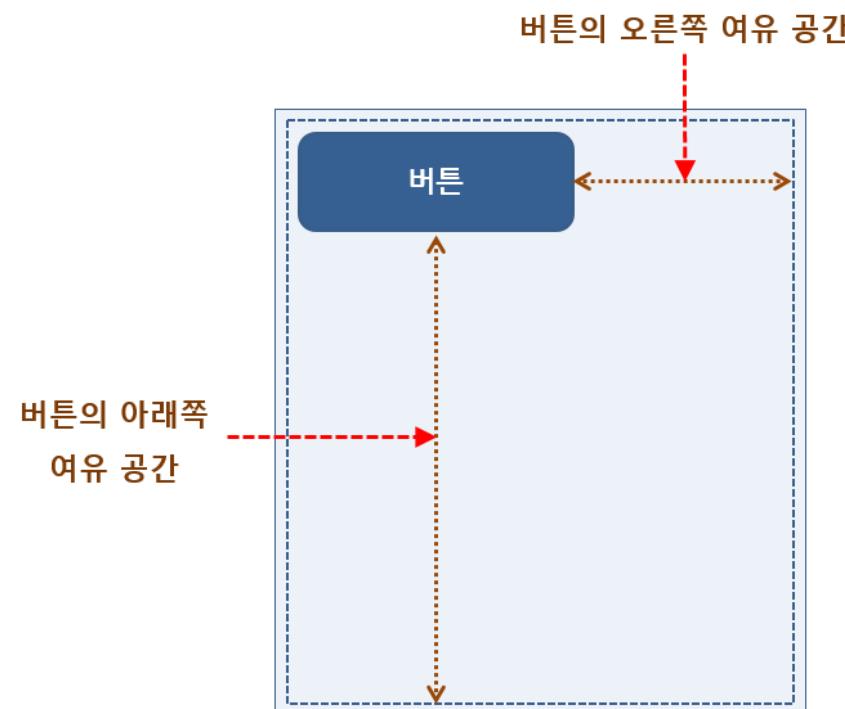
- 모서리에 있는 핸들(Handle)을 이용해 뷰의 크기 조정 가능





뷰가 차지할 수 있는 여유공간

- 부모 레이아웃 안에서 뷰가 차지할 수 있는 여유공간은 레이아웃에 의해 결정됨

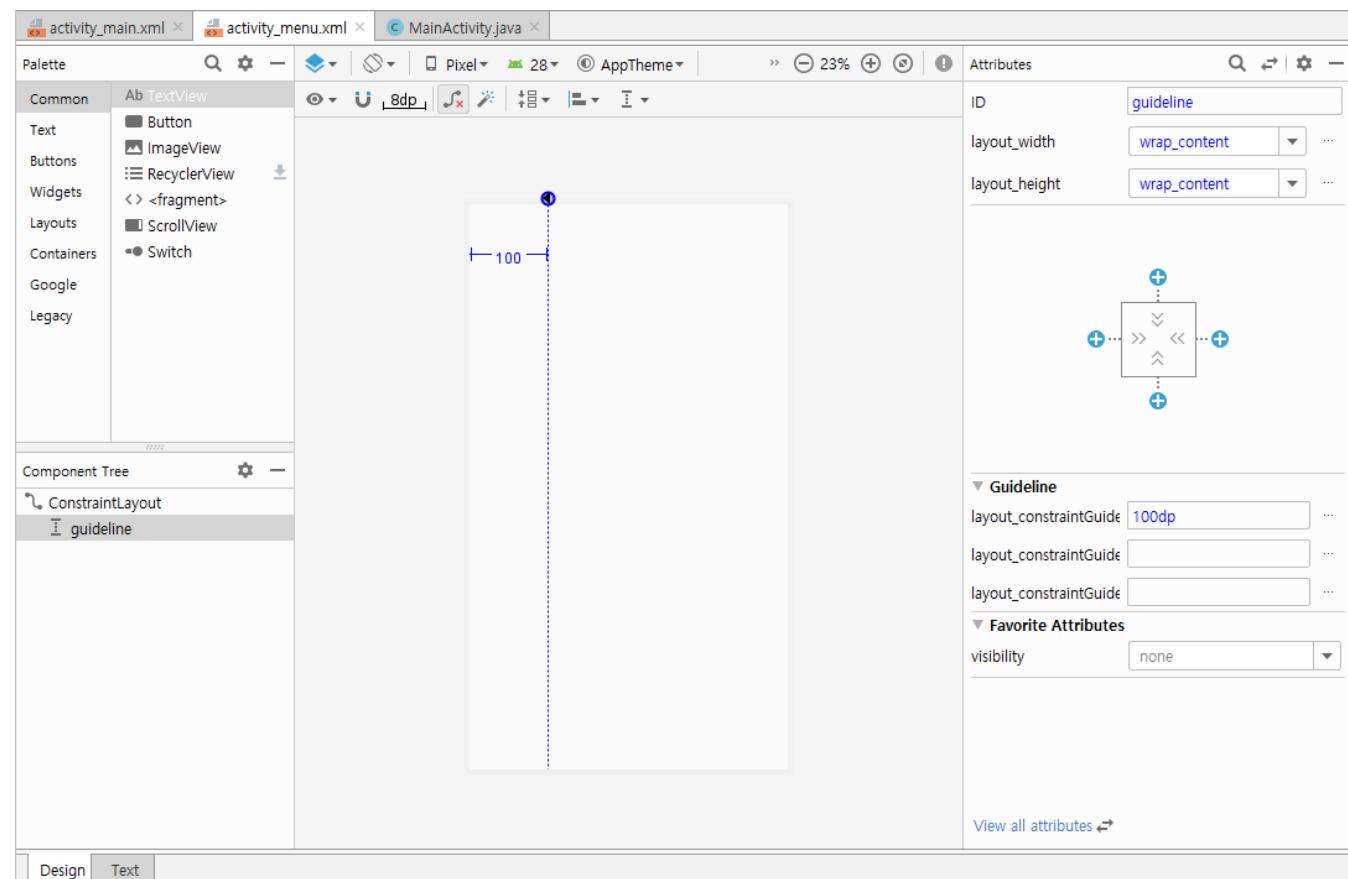
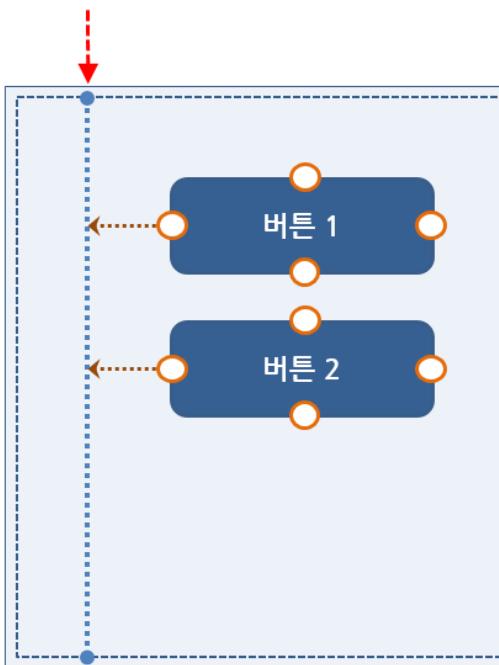




가이드라인 사용하기

- 일정 기준선으로 뷰를 정렬할 때나 기준선에 맞추어 추가할 때 사용됨

세로방향 가이드라인





XML 코드로 만들어지는 가이드라인

- XML 레이아웃 파일에서 원본 XML을 보면 Guideline 태그로 만들어짐

```
<android.support.constraint.Guideline  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/guideline"  
    android:orientation="vertical"  
    tools:layout_editor_absoluteY="0dp"  
    tools:layout_editor_absoluteX="20dp"  
    app:layout_constraintGuide_begin="100dp" />  
  
...
```



가이드라인의 원본 코드 살펴보기

- xmlns로 시작하는 속성, android로 시작하는 속성 등이 있음

```
<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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">  
  
    ...
```



XML 코드의 접두어 의미

- XML 레이아웃 코드에서 접두어가 가지는 의미가 있음

- (1) xmlns:android 안드로이드 기본 SDK에 포함되어 있는 속성을 사용합니다.
- (2) xmlns:app 프로젝트에서 사용하는 외부 라이브러리에 포함되어 있는 속성을 사용합니다.
- (3) xmlns:tools 안드로이드 스튜디오의 디자이너 도구 등에서 화면에 보여줄 때 사용합니다.

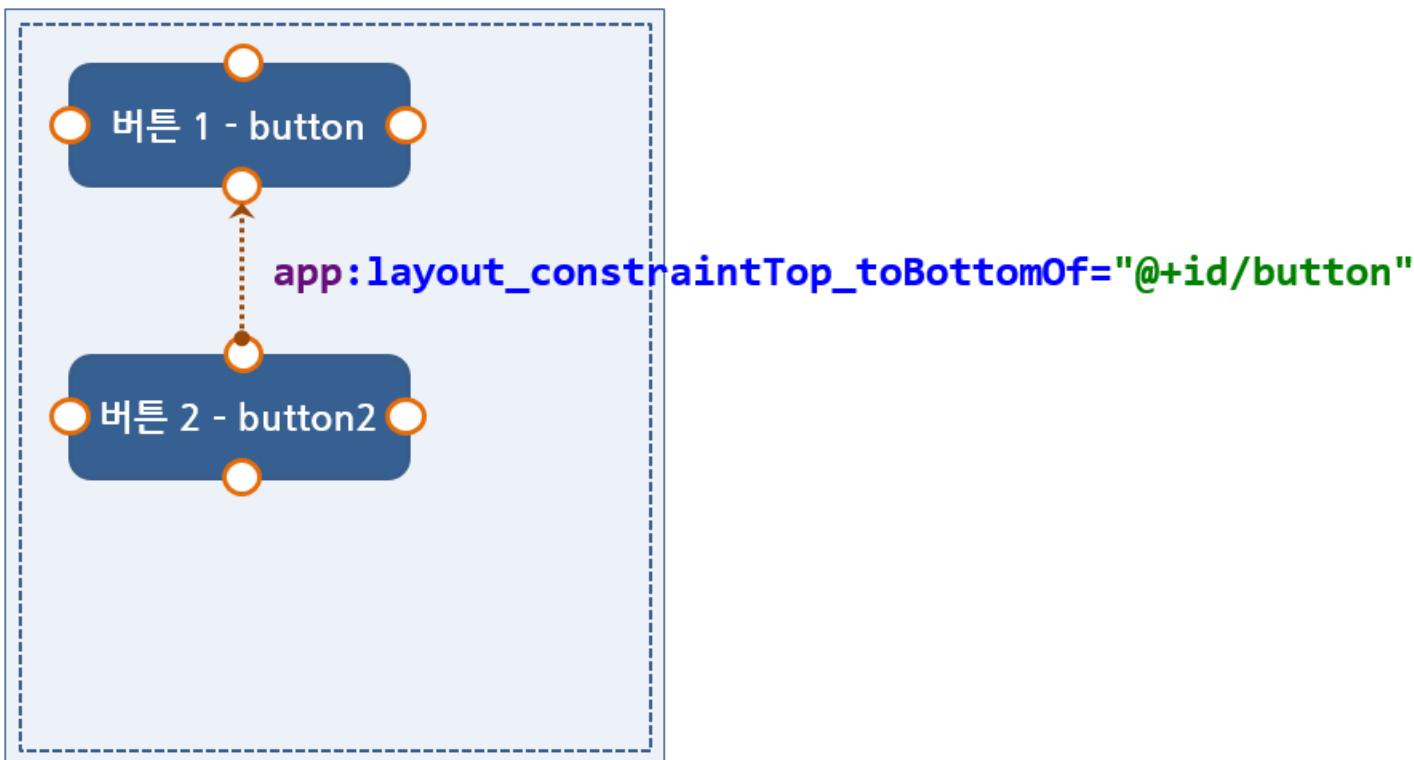
이 속성은 앱이 실행될 때는 적용되지 않고 안드로이드 스튜디오에서만 적용됩니다.



연결선이 만들어내는 XML 속성의 형식

- 일정한 규칙으로 만들어짐

```
layout_constraint[소스 뷰의 연결점]_[타깃 뷰의 연결점]=[타깃 뷰의 id]
```





뷰의 크기 지정에 사용되는 단위

단위	단위 표현	설명
px	픽셀	화면 픽셀
dp 또는 dp p	밀도 독립적 픽셀 (density independent pixel)	160dpi 화면을 기준으로 한 픽셀 예) 1인치 당 160개의 점이 있는 디스플레이 화면에서 1dp는 1px와 같음. 1인치 당 320개의 점이 있는 디스플레이 화면에서 1dp는 2px와 같음.
sp 또는 sip	축척 독립적 픽셀 (scale independent pixel)	가변 글꼴을 기준으로 한 픽셀로 dp와 유사하나 글꼴의 설정에 따라 달라짐
in	인치	1인치로 된 물리적 길이
mm	밀리미터	1밀리미터로 된 물리적 길이
em	텍스트 크기	글꼴과 상관없이 동일한 텍스트 크기 표시



대표적인 레이아웃 이해하기



- 1 대표적인 레이아웃 살펴보기
- 2 리니어 레이아웃 사용하기
- 3 상대 레이아웃 사용하기
- 4 테이블 레이아웃 사용하기
- 5 프레임 레이아웃과 뷰의 전환
- 6 스크롤뷰 사용하기



레이아웃에는 어떤 것들이 있는지 알아볼까요?

- 대표적인 레이아웃 살펴보기



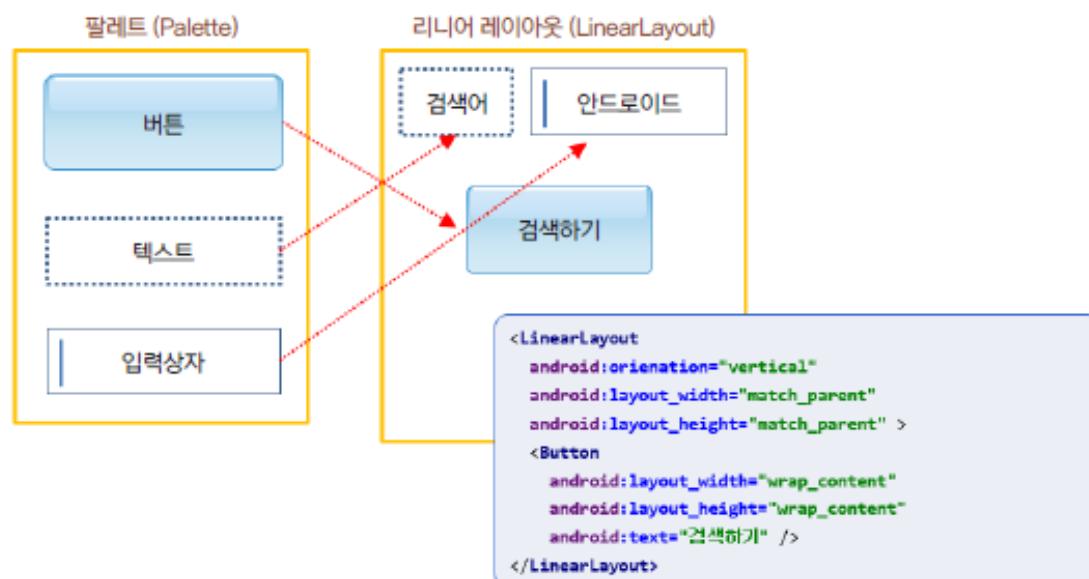
대표적인 레이아웃을 사용해 볼까요?

- 리니어 레이아웃 사용하기
- 상대 레이아웃 사용하기
- 테이블 레이아웃 사용하기
- 프레임 레이아웃 사용하기



스크롤을 만들어 볼까요?

- 스크롤 사용하기





대표적인 레이아웃 살펴보기

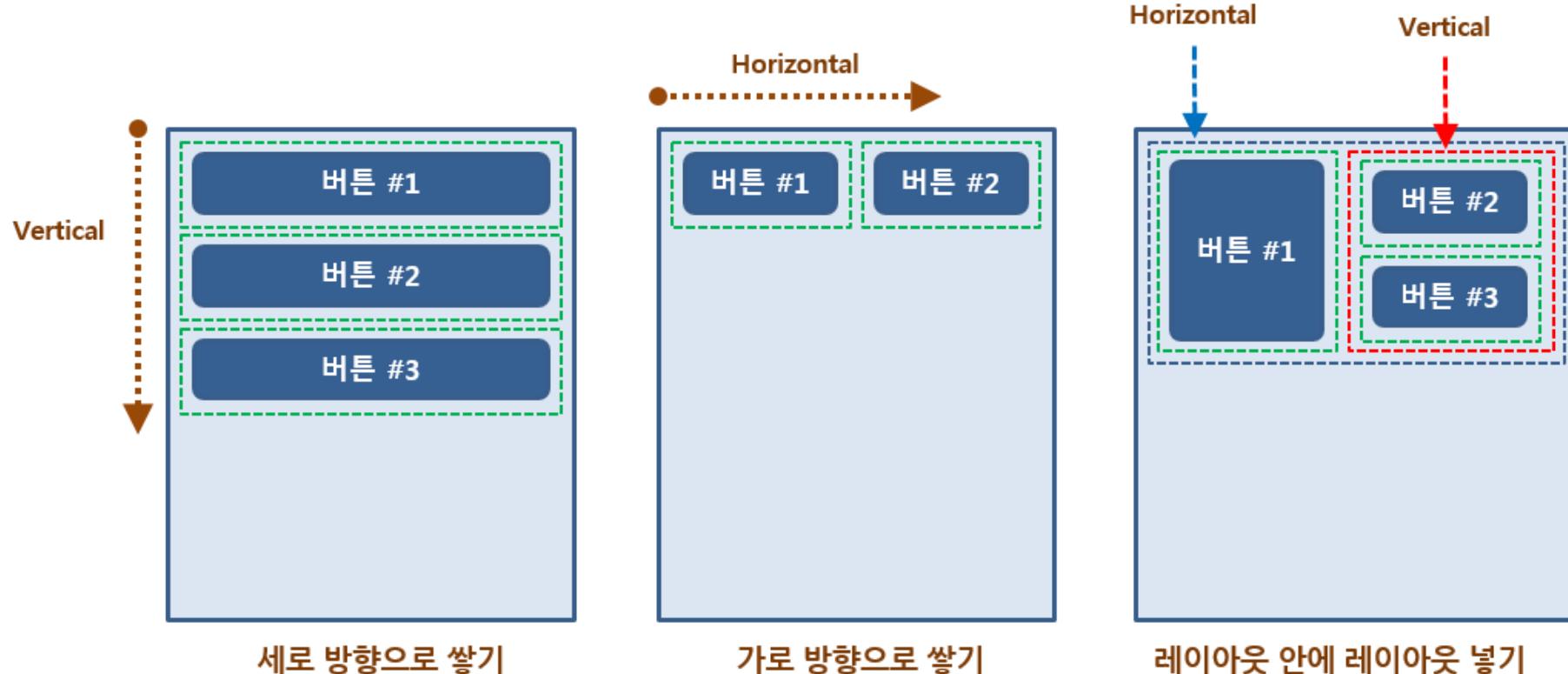


대표적인 레이아웃

레이아웃 이름	설명
제약 레이아웃 (ConstraintLayout)	제약 조건(Constraint) 기반 모델 제약 조건을 사용해 화면을 구성하는 방법 안드로이드 스튜디오에서 자동으로 설정하는 디폴트 레이아웃
리니어 레이아웃 (LinearLayout)	박스(Box) 모델 한 쪽 방향으로 차례대로 뷰를 추가하며 화면을 구성하는 방법 뷰가 차지할 수 있는 사각형 영역을 할당
상대 레이아웃 (RelativeLayout)	규칙(Rule) 기반 모델 부모 컨테이너나 다른 뷰와의 상대적 위치로 화면을 구성하는 방법
프레임 레이아웃 (FrameLayout)	싱글(Single) 모델 가장 상위에 있는 하나의 뷰 또는 뷰그룹만 보여주는 방법 여러 개의 뷰가 들어가면 중첩하여 쌓게 됨. 가장 단순하지만 여러 개의 뷰를 중첩한 후 각 뷰를 전환하여 보여주는 방식으로 자주 사용함
테이블 레이아웃 (TableLayout)	격자(Grid) 모델 격자 모양의 배열을 사용하여 화면을 구성하는 방법 HTML에서 많이 사용하는 정렬 방식과 유사하지만 많이 사용하지는 않음



리니어 레이아웃 사용방식





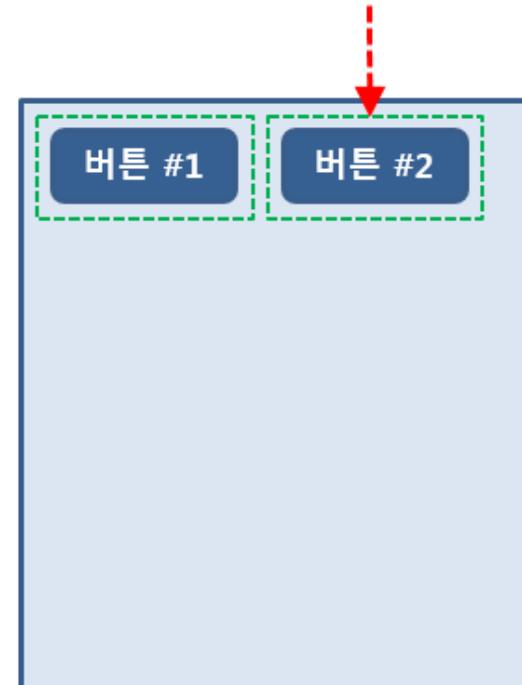
상대 레이아웃 사용방식

부모 컨테이너의 위쪽(Top)



부모 컨테이너와의 관계 이용

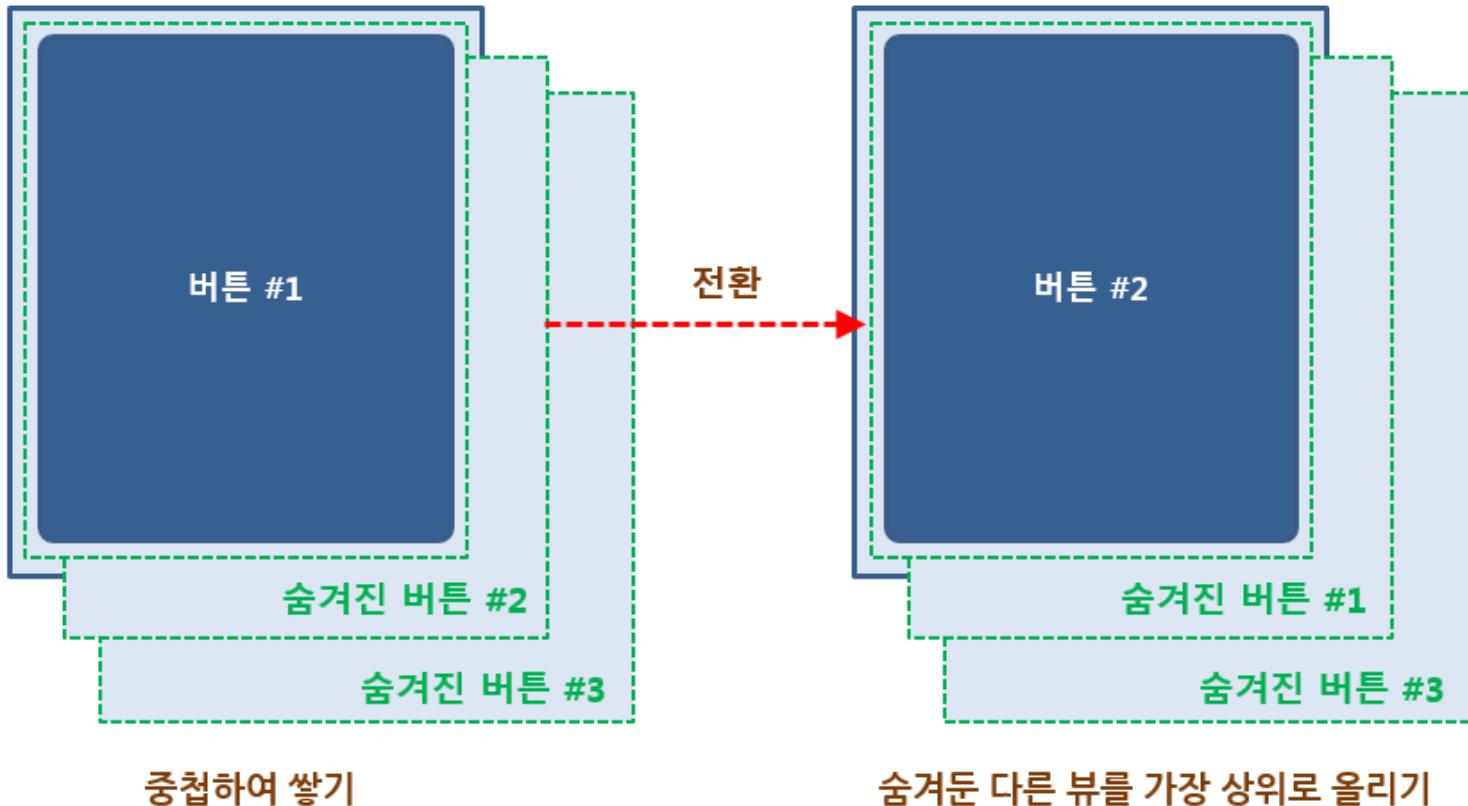
버튼 #1의 오른쪽(Right)



다른 뷰와의 관계 이용

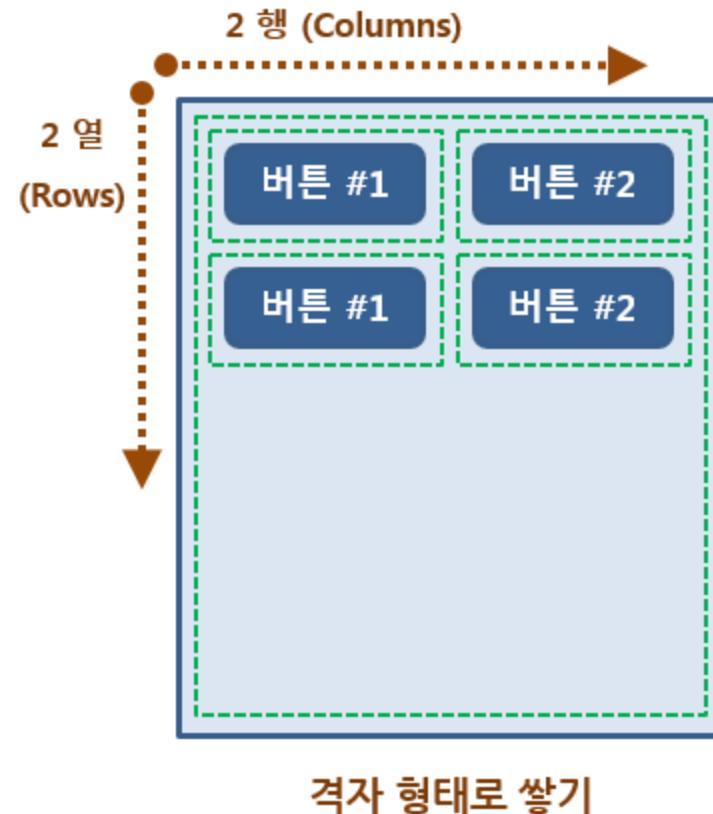


프레임 레이아웃 사용방식





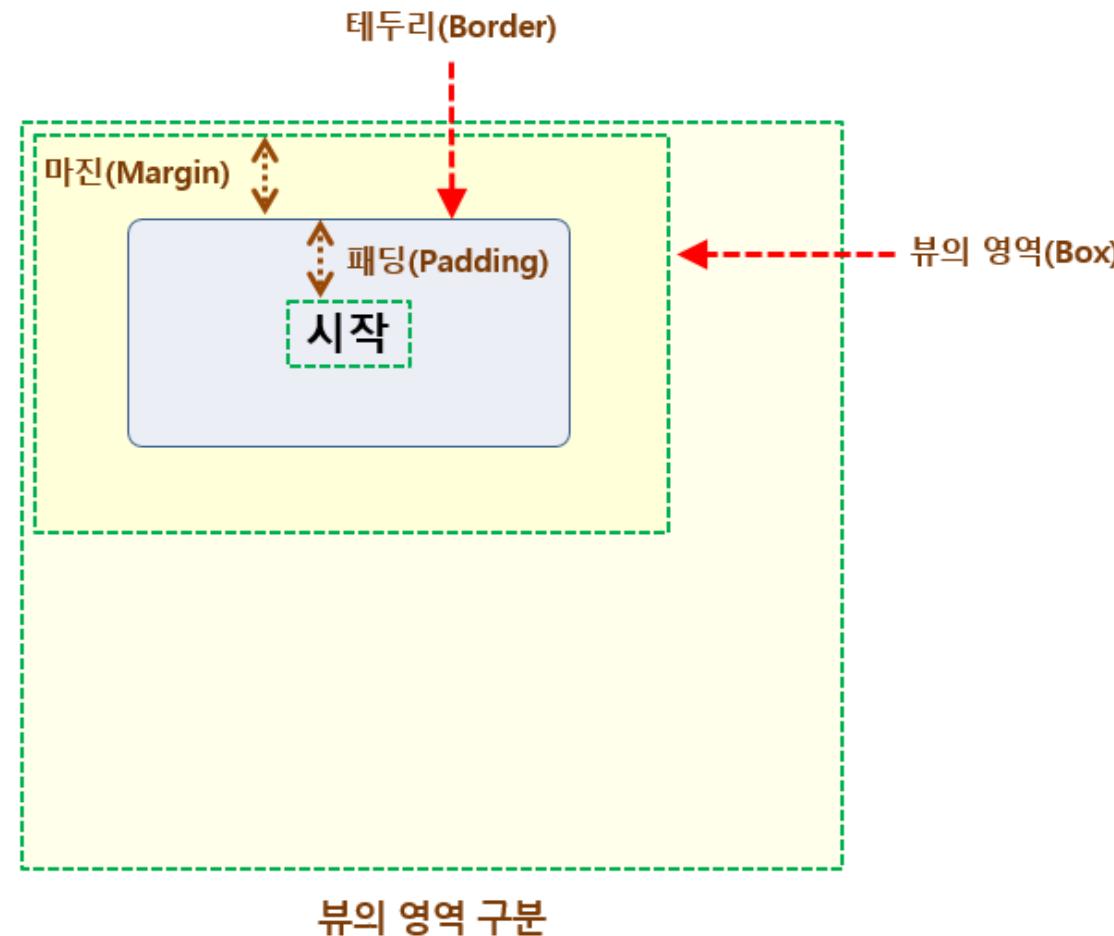
테이블 레이아웃 사용방식





뷰의 영역

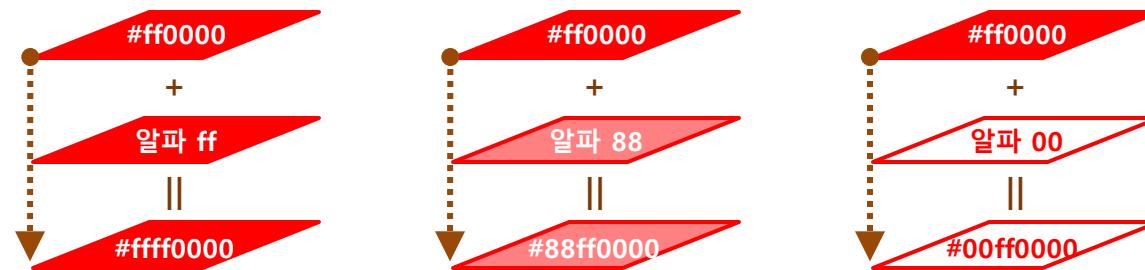
- 테두리를 기준으로 바깥쪽이 마진, 안쪽이 패딩임
- 뷰의 영역은 마진까지를 포함함





뷰의 배경색

- **background 속성으로 배경색 설정**
- 배경 이미지를 설정할 수도 있음
- 배경색은 ARGB를 기준으로 16진수 두 자리씩 할당하며 # 뒤에 코드를 붙임



```
android:background="@drawable/house"
```



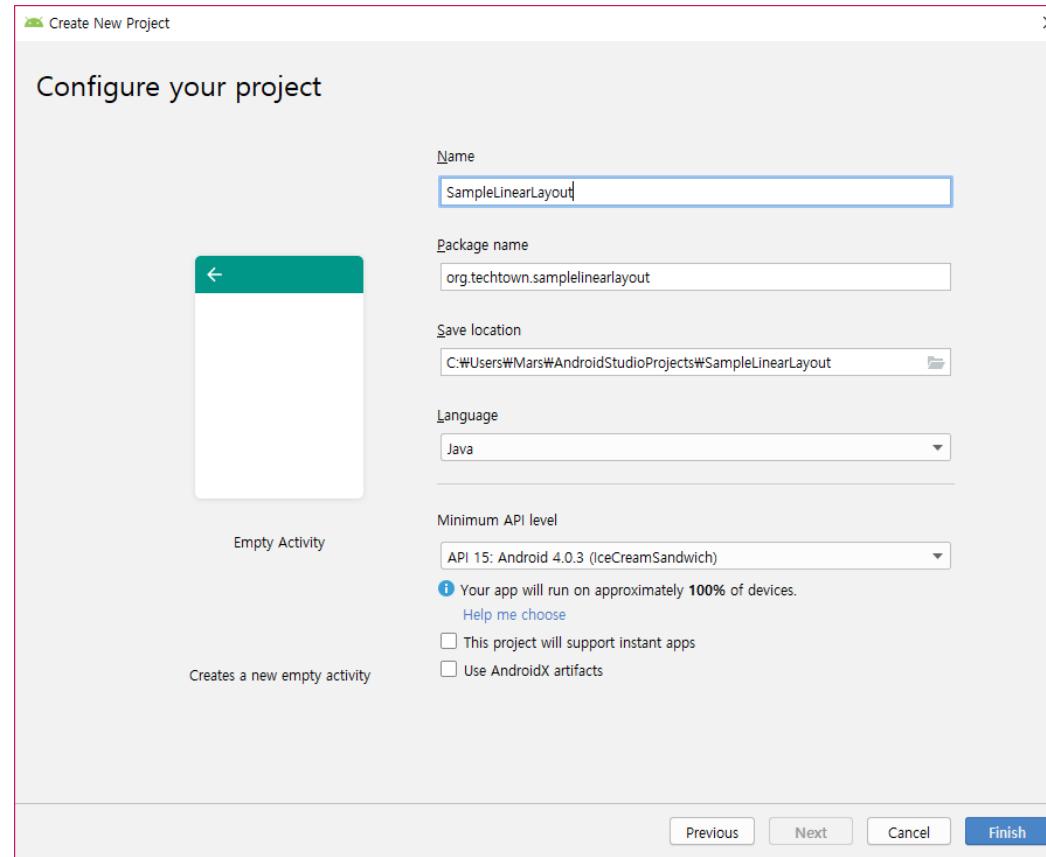
2.

리니어 레이아웃 사용하기



새로운 프로젝트 생성

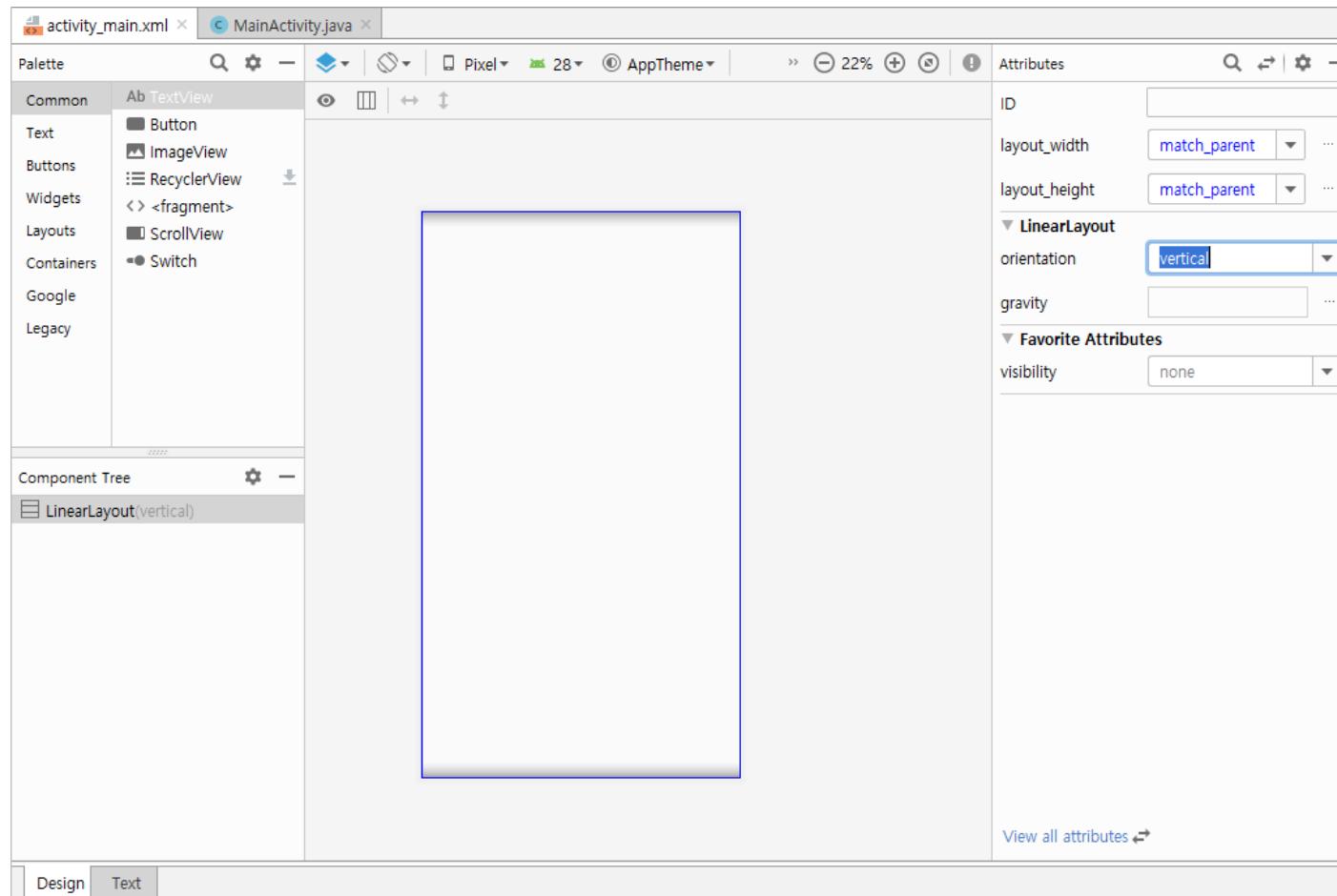
- SampleLinearLayout 이라는 이름으로 새로운 프로젝트 생성





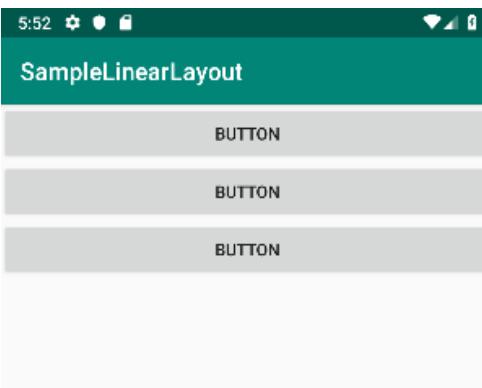
리니어 레이아웃 – 방향 설정하기

- 방향 속성은 리니어 레이아웃의 필수 속성임
- 가로(horizontal), 세로(vertical)

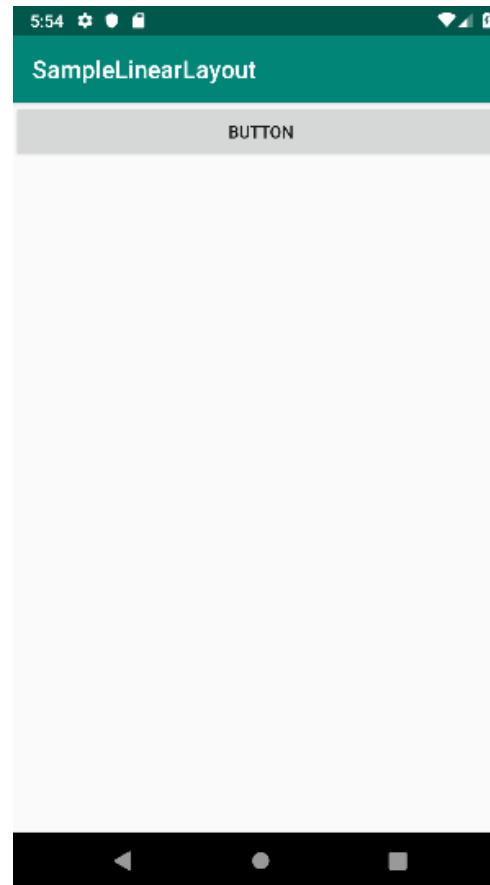




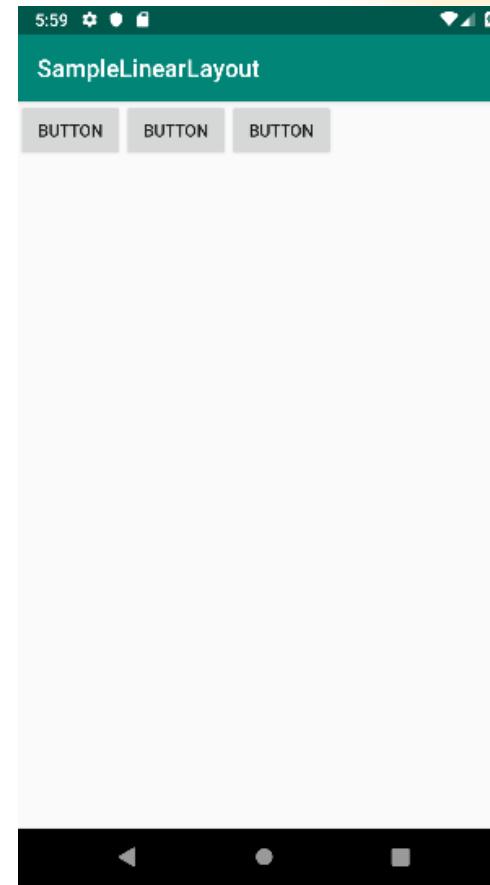
리니어 레이아웃 – 방향 설정하기 (계속)



[세로 방향으로 설정한 경우]



[가로 방향으로 설정을 바꾼 경우]



[버튼의 layout_width 속성을 wrap_content로 바꾼 경우]



리니어 레이아웃 – 자바 코드에서 구성하기

- 자바 코드에서 직접 레이아웃 객체를 만들고 파라미터 설정하는 방법

```
LinearLayout mainLayout = new LinearLayout(this);  
mainLayout.setOrientation(LinearLayout.VERTICAL);
```

1

레이아웃 객체 생성

```
LinearLayout.LayoutParams params = new LinearLayout.LayoutParams(  
    LinearLayout.LayoutParams.MATCH_PARENT,  
    LinearLayout.LayoutParams.WRAP_CONTENT);
```

2

파라미터 설정

```
Button button1 = new Button(this);  
button1.setText("Button 01");  
button1.setLayoutParams(params);  
mainLayout.addView(button1);
```

3

버튼 객체 생성하여 추가

```
setContentView(mainLayout);
```

4

화면 설정



리니어 레이아웃 – 뷰 정렬하기

[두 가지 정렬 속성]

정렬 속성	설명
layout_gravity	- [외부] 부모 컨테이너의 여유 공간에 뷰가 모두 채워지지 않아 여유 공간 안에서 뷰를 정렬할 때
gravity	- [내부] 뷰에서 화면에 표시하는 내용물을 정렬할 때 (텍스트뷰의 경우, 내용물은 글자가 되고 이미지뷰의 경우 내용물은 이미지가 됨)

• **layout_gravity**

- 뷰의 layout_width나 layout_height 속성이 match_parent가 아닐 경우에 같이 사용할 수 있음



리니어 레이아웃 – 뷰 정렬하기 (계속)

The screenshot shows the Android Studio Layout Editor with the following interface elements:

- Top Bar:** Displays "activity_main.xml" and "gravity.xml" tabs.
- Toolbar:** Includes icons for search, settings, orientation, pixel density, zoom level (28%), and theme (AppTheme).
- Attributes Panel:** Shows the properties for the selected view, which is a Button with ID "button4". The "layout_gravity" property is set to "[left]" and "center_horizontal" is checked.
- Component Tree:** Shows a "LinearLayout(vertical)" containing three buttons: "button4 - 'Left'", "button5 - 'Center'", and "button6 - 'Right'".
- Design View:** Shows three buttons arranged vertically. The first button is labeled "LEFT" and has its gravity set to "left". The second button is labeled "CENTER" and has its gravity set to "center". The third button is labeled "RIGHT" and has its gravity set to "right".
- Bottom Navigation:** Buttons for "Design" and "Text".



리니어 레이아웃 – 정렬을 위해 사용할 수 있는 값

[정렬을 위해 gravity 속성에 지정할 수 있도록 정의된 값]

정렬 속성값	설명
top	- 대상 객체를 위쪽 끝에 배치하기
bottom	- 대상 객체를 아래쪽 끝에 배치하기
left	- 대상 객체를 왼쪽 끝에 배치하기
right	- 대상 객체를 오른쪽 끝에 배치하기
center_vertical	- 대상 객체를 수직 방향의 중앙에 배치하기
center_horizontal	- 대상 객체를 수평 방향의 중앙에 배치하기
fill_vertical	- 대상 객체를 수직 방향으로 여유 공간만큼 확대하여 채우기
fill_horizontal	- 대상 객체를 수평 방향으로 여유 공간만큼 확대하여 채우기
center	- 대상 객체를 수직 방향과 수평 방향의 중앙에 배치하기
fill	- 대상 객체를 수직 방향과 수평 방향으로 여유 공간만큼 확대하여 채우기
clip_vertical	- 대상 객체의 상하 길이가 여유 공간보다 클 경우에 남는 부분을 잘라내기 - top clip_vertical로 설정한 경우 아래쪽에 남는 부분 잘라내기 - bottom clip_vertical로 설정한 경우 위쪽에 남는 부분 잘라내기 - center_vertical clip_vertical로 설정한 경우 위쪽과 아래쪽에 남는 부분 잘라내기
clip_horizontal	- 대상 객체의 좌우 길이가 여유 공간보다 클 경우에 남는 부분을 잘라내기 - right clip_horizontal로 설정한 경우 왼쪽에 남는 부분 잘라내기 - left clip_horizontal로 설정한 경우 오른쪽에 남는 부분 잘라내기 - center_horizontal clip_horizontal로 설정한 경우 왼쪽과 오른쪽에 남는 부분 잘라내기

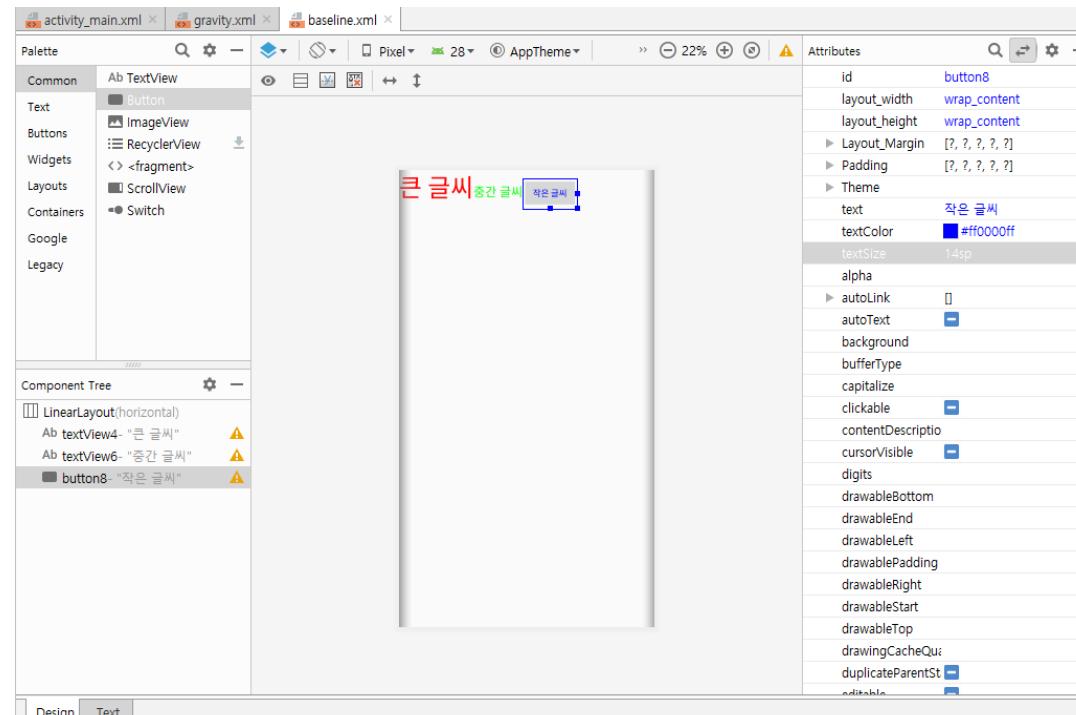


리니어 레이아웃 – 글자 아랫줄 정렬

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

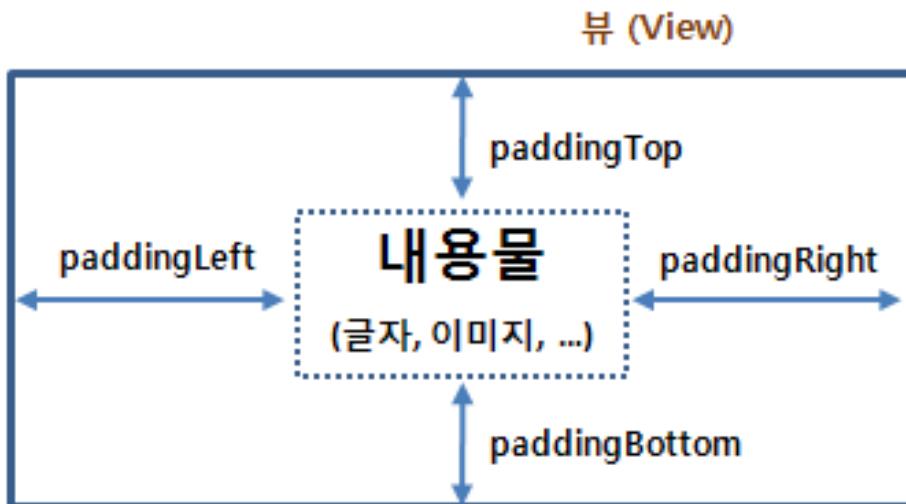
1

글자의 아랫줄 맞추기

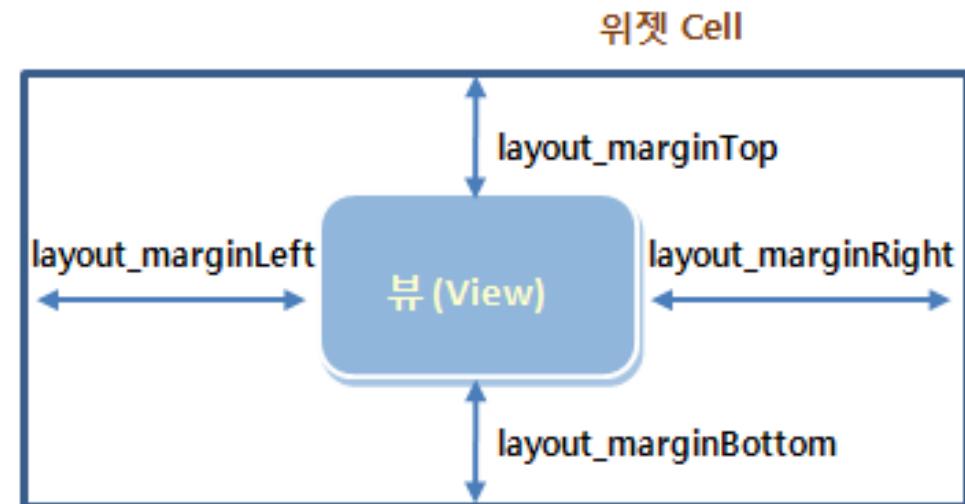




리니어 레이아웃 – 마진과 패딩 설정하기



[padding을 이용한 뷰 내부의 여백 주기]



[layout_margin을 이용한 부모 여유공간과의 여백 주기]

- **padding 속성**

- 뷰 안의 내용물인 텍스트나 이미지와 뷰 안의 영역 사이의 여백을 줄 수 있는 방법

- **layout_margin 속성**

- 부모 컨테이너의 여유 공간과 뷰 사이의 여백을 줄 수 있는 방법

- **위젯 셀**

- 위젯이나 뷰들은 부모 컨테이너로부터 할당된 공간을 차지하게 되며 이를 '위젯 셀(cell)'이라고 부름



리니어 레이아웃 – 마진과 패딩 설정하기 (계속)

The screenshot shows the Android Studio interface with the following details:

- Toolbar:** Includes tabs for activity_main.xml, gravity.xml, baseline.xml, and padding.xml; zoom controls; and a theme dropdown set to AppTheme.
- Palette:** Shows categories like Common, Text, Buttons, Widgets, Layouts, Containers, Google, and Legacy. A specific `Ab TextView` item is selected.
- Component Tree:** Displays a `LinearLayout(horizontal)` containing three `Ab textView` items, each with a yellow background and red text.
- Design View:** Shows three `TextView` components arranged horizontally. The first has a yellow background, the second has a cyan background, and the third has a magenta background. They are separated by gaps.
- Attributes Panel:** Shows the following properties for the selected `textView8`:
 - `id`: `textView8`
 - `layout_width`: `wrap_content`
 - `layout_height`: `wrap_content`
 - `Layout_Margin`: `[?, ?, ?, ?, ?]` (with sub-options for all, bottom, left, right, top)
 - `Padding`: `[20dp, ?, ?, ?, ?]` (with sub-options for all, bottom, left, right, top)
- Text Tab:** Shows XML code for the `padding.xml` file, which defines a `padding` style with `padding: 20dp`.

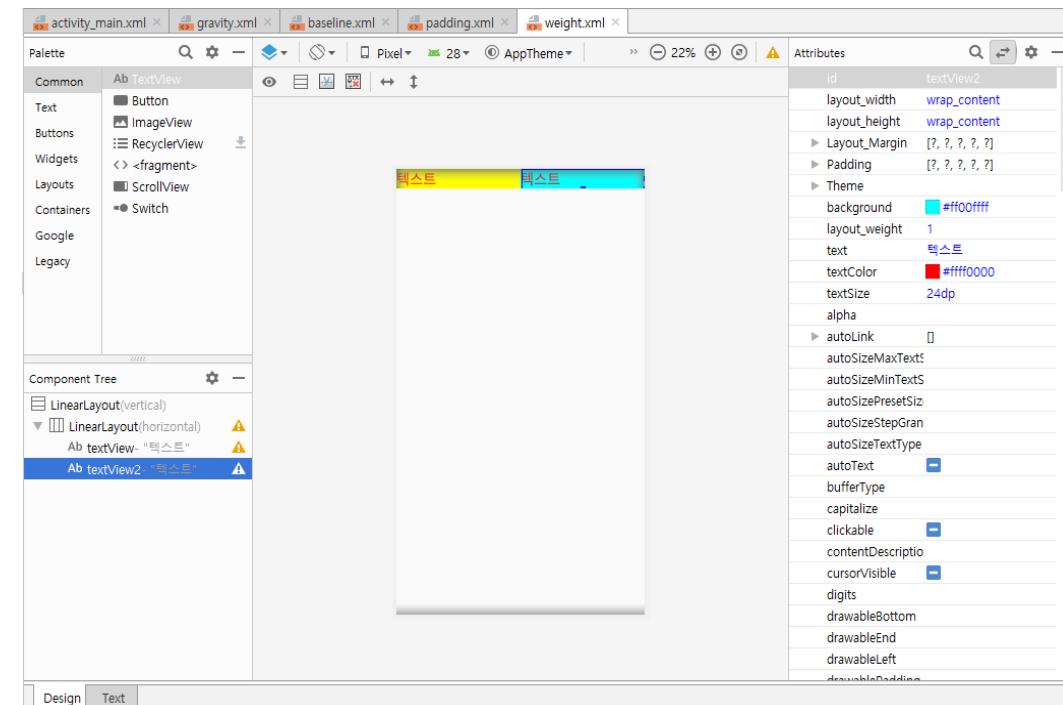
The screenshot shows the Android Studio interface with the following details:

- Toolbar:** Same as the first screenshot.
- Palette:** Same as the first screenshot.
- Component Tree:** Same as the first screenshot.
- Design View:** Shows the same three `TextView` components, but the gaps between them are significantly larger, indicating the effect of the increased padding.
- Attributes Panel:** Shows the following properties for the selected `textView8`:
 - `id`: `textView8`
 - `layout_width`: `wrap_content`
 - `layout_height`: `wrap_content`
 - `Layout_Margin`: `[?, ?, ?, ?, ?]` (with sub-options for all, bottom, left, right, top)
 - `Padding`: `[50dp, ?, ?, ?, ?]` (with sub-options for all, bottom, left, right, top)
- Text Tab:** Shows XML code for the `padding.xml` file, which defines a `padding` style with `padding: 50dp`.



리니어 레이아웃 – 여유공간 분할하기

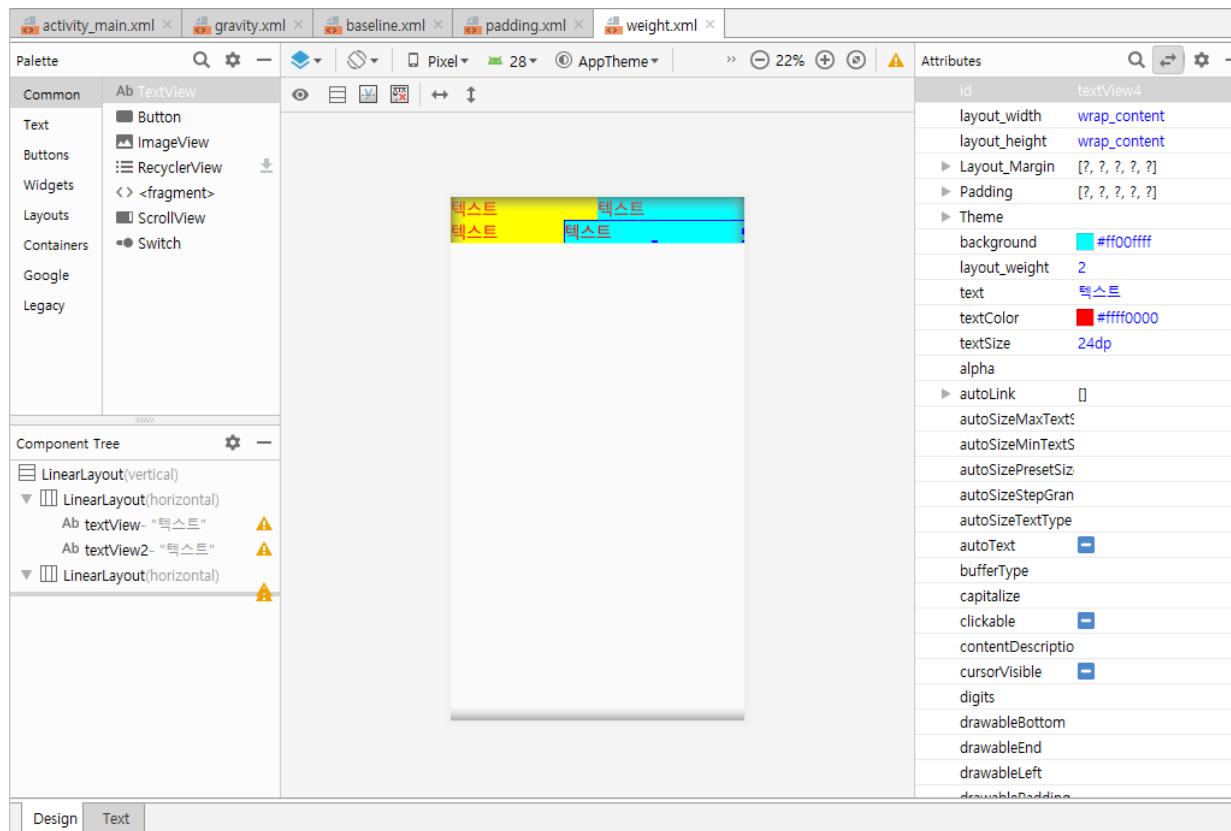
- **layout_weight** 속성은 같은 남아있는 여유공간을 얼마나 차지할 수 있는지를 비율로 지정하는 것
- **android:layout_weight** 속성 사용
- 두 개의 뷰에 모두 1 값을 설정한 경우





리니어 레이아웃 – 여유공간 분할하기

- 분할하려는 방향의 크기를 0dp로 하는 경우 공간을 분할하는 효과가 생김
- layout_width 속성값을 0dp로 설정한 후 하나는 1, 다른 하나는 2로 설정





리니어 레이아웃 – 레이아웃 안에 레이아웃 넣기

- 세로 방향의 리니어 레이아웃 안에 가로 방향의 리니어 레이아웃을 넣을 수 있음
- 레이아웃 안에 레이아웃을 계속 넣을 수 있으나 레벨이 너무 깊으면 시스템 부하 증가

```
<LinearLayout android:orientation="horizontal" >  
    <ImageView />  
    <LinearLayout android:orientation="vertical" >  
        <TextView />  
        <TextView />  
    </LinearLayout>  
</LinearLayout>
```

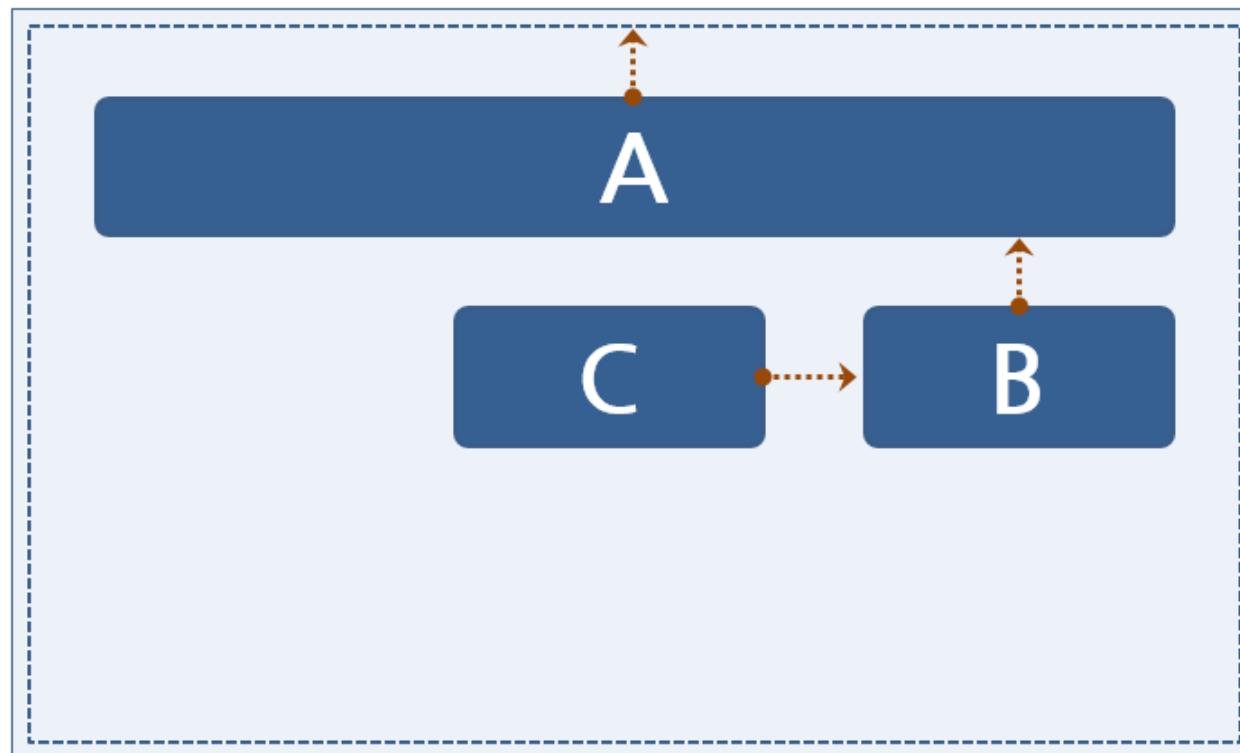


상대 레이아웃 사용하기



상대 레이아웃

- 상대 레이아웃은 다른 뷰나 부모 뷰와의 상대적인 위치를 이용해 뷰를 배치하는 방법

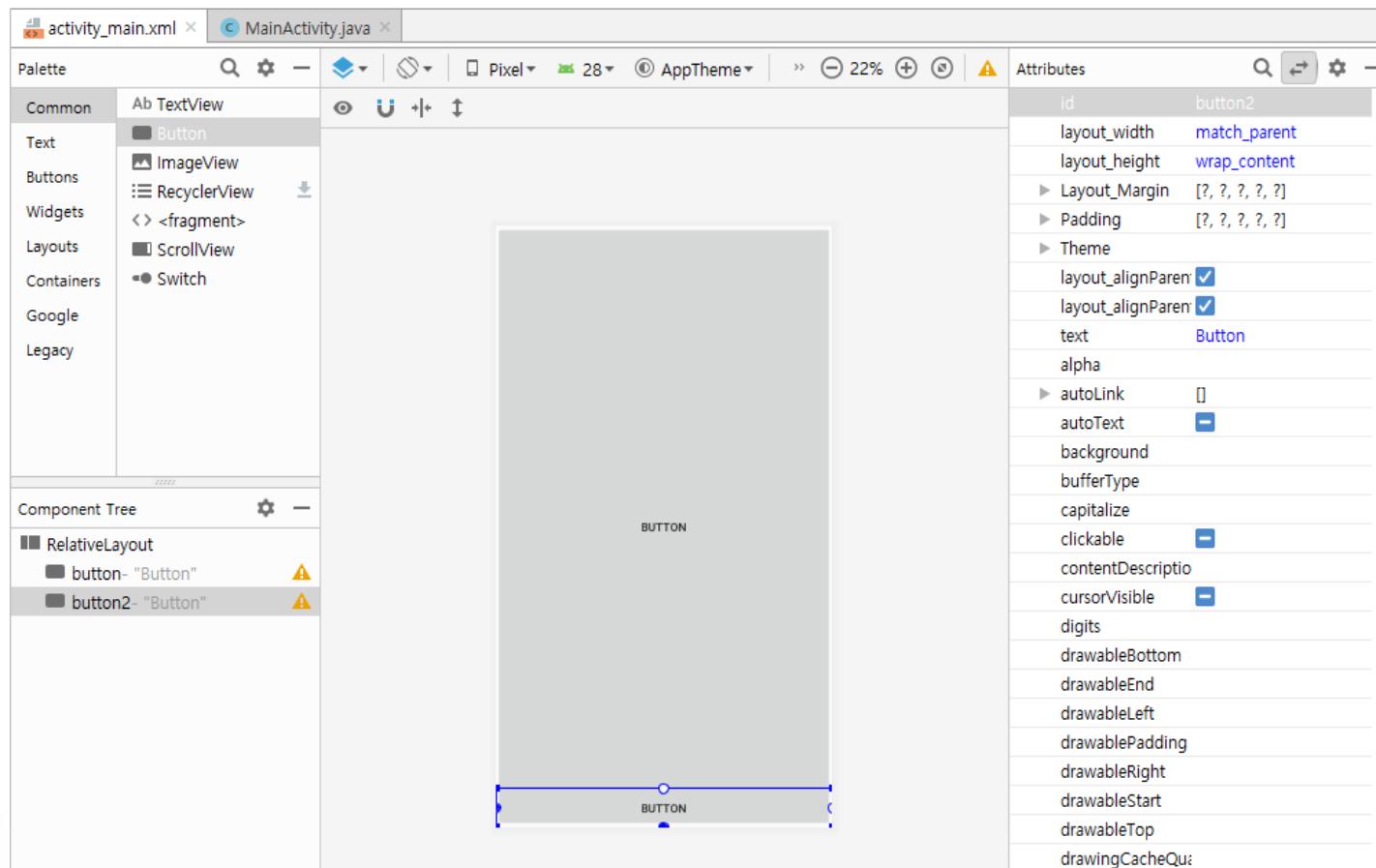


A : 부모 레이아웃의 위쪽
B : 뷰 A의 아래쪽
C : 뷰 B의 왼쪽



상대 레이아웃 화면 배치

- XML 레이아웃 파일에서 가운데 하나, 아래쪽에 하나 배치

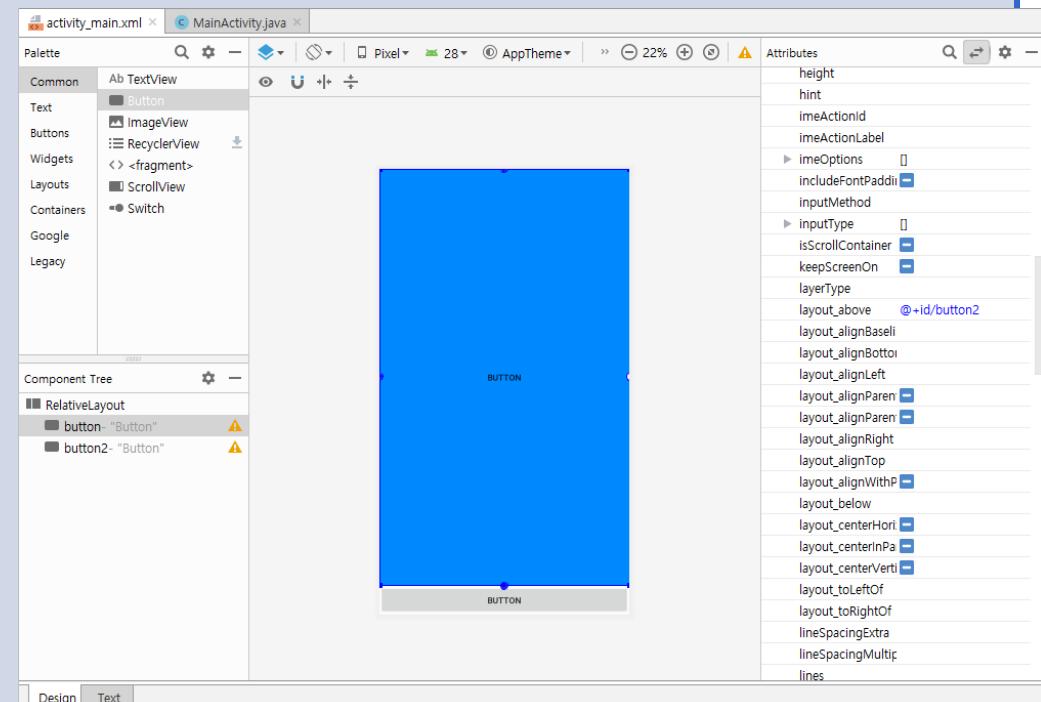




상대 레이아웃의 속성 사용

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

    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_above="@+id/button2"
        android:text="Button"
        android:background="#ff0088ff" />
```



Continued..



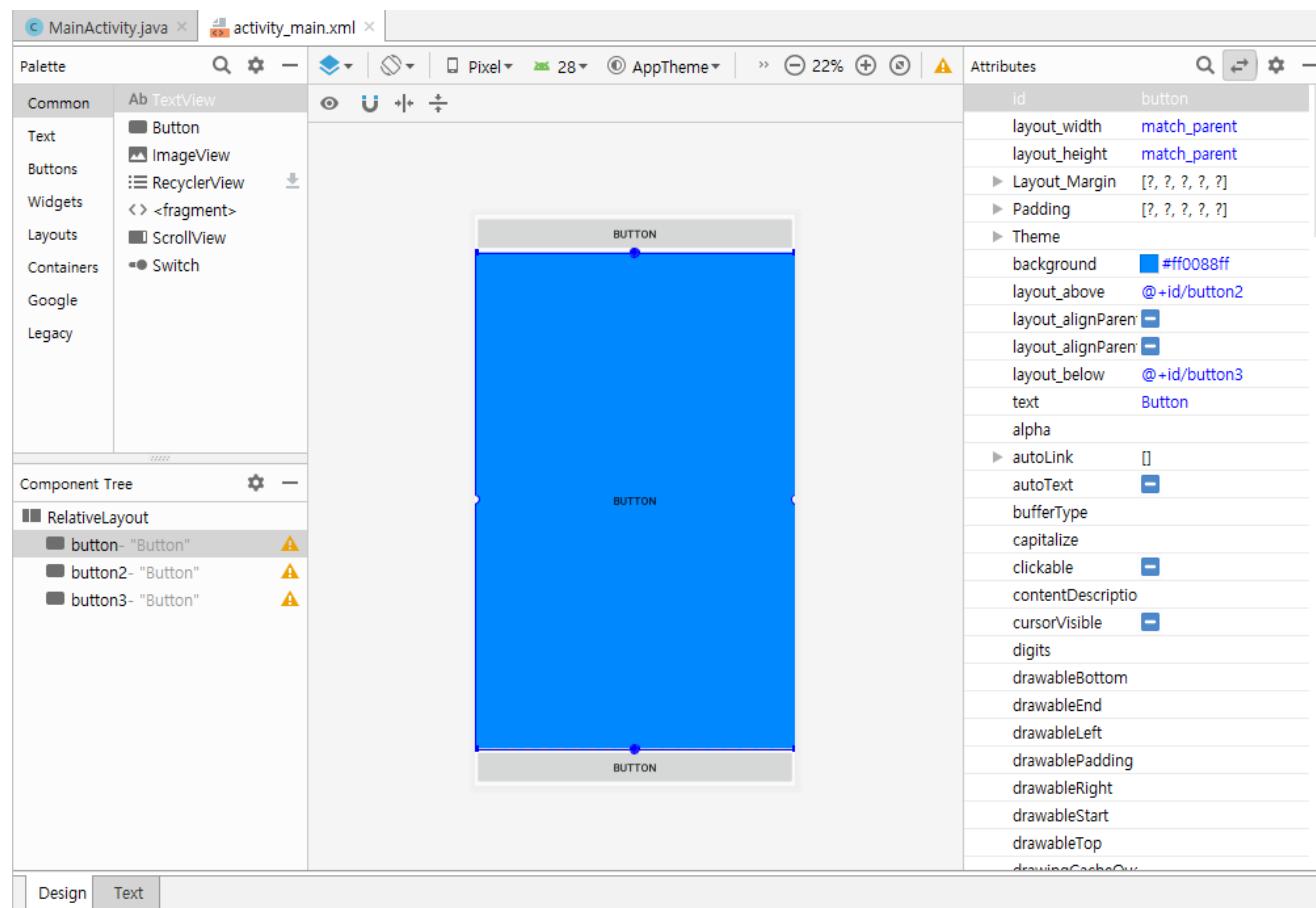
상대 레이아웃의 속성 사용 (계속)

```
<Button  
    android:id="@+id/button2"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Button"  
    android:layout_alignParentBottom="true"  
    android:layout_alignParentLeft="true"  
    android:layout_alignParentStart="true" />  
</RelativeLayout>
```



상대 레이아웃 화면 배치

- XML 레이아웃 파일에서 가운데 하나, 위와 아래쪽에 하나씩 배치





상대 레이아웃에서 사용할 수 있는 속성들

[상대 레이아웃에서 부모 컨테이너와의 상대적 위치를 이용하는 속성]

속성	설명
layout_alignParentTop	- 부모 컨테이너의 위쪽과 뷰의 위쪽을 맞춤
layout_alignParentBottom	- 부모 컨테이너의 아래쪽과 뷰의 아래쪽을 맞춤
layout_alignParentLeft	- 부모 컨테이너의 왼쪽 끝과 뷰의 왼쪽 끝을 맞춤
layout_alignParentRight	- 부모 컨테이너의 오른쪽 끝과 뷰의 오른쪽 끝을 맞춤
layout_centerHorizontal	- 부모 컨테이너의 수평 방향 중앙에 배치함
layout_centerVertical	- 부모 컨테이너의 수직 방향 중앙에 배치함
layout_centerInParent	- 부모 컨테이너의 수평과 수직 방향 중앙에 배치함



상대 레이아웃에서 사용할 수 있는 속성들

[상대 레이아웃에서 다른 뷰와의 상대적 위치를 이용하는 속성]

속성	설명
layout_above	- 지정한 뷰의 위쪽에 배치함
layout_below	- 지정한 뷰의 아래쪽에 배치함
layout_toLeftOf	- 지정한 뷰의 왼쪽에 배치함
layout_toRightOf	- 지정한 뷰의 오른쪽에 배치함
layout_alignTop	- 지정한 뷰의 위쪽과 맞춤
layout_alignBottom	- 지정한 뷰의 아래쪽과 맞춤
layout_alignLeft	- 지정한 뷰의 왼쪽과 맞춤
layout_alignRight	- 지정한 뷰의 오른쪽과 맞춤
layout_alignBaseline	- 지정한 뷰와 내용물의 아래쪽 기준선(baseline)을 맞춤



테이블 레이아웃 사용하기



테이블 레이아웃

- 테이블 레이아웃은 격자 모양으로 뷰를 배치하는 방법

<TableRow>

홍길동

20

<TableRow>

홍길순

24

[테이블 레이아웃을 이용한 뷰의 배치 방법]

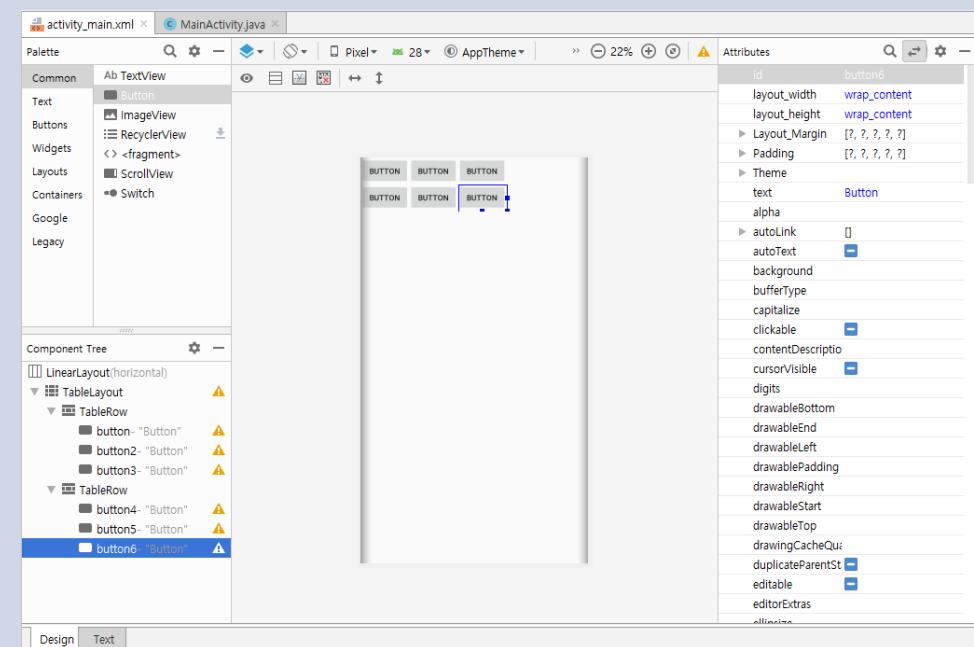


테이블 레이아웃

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

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

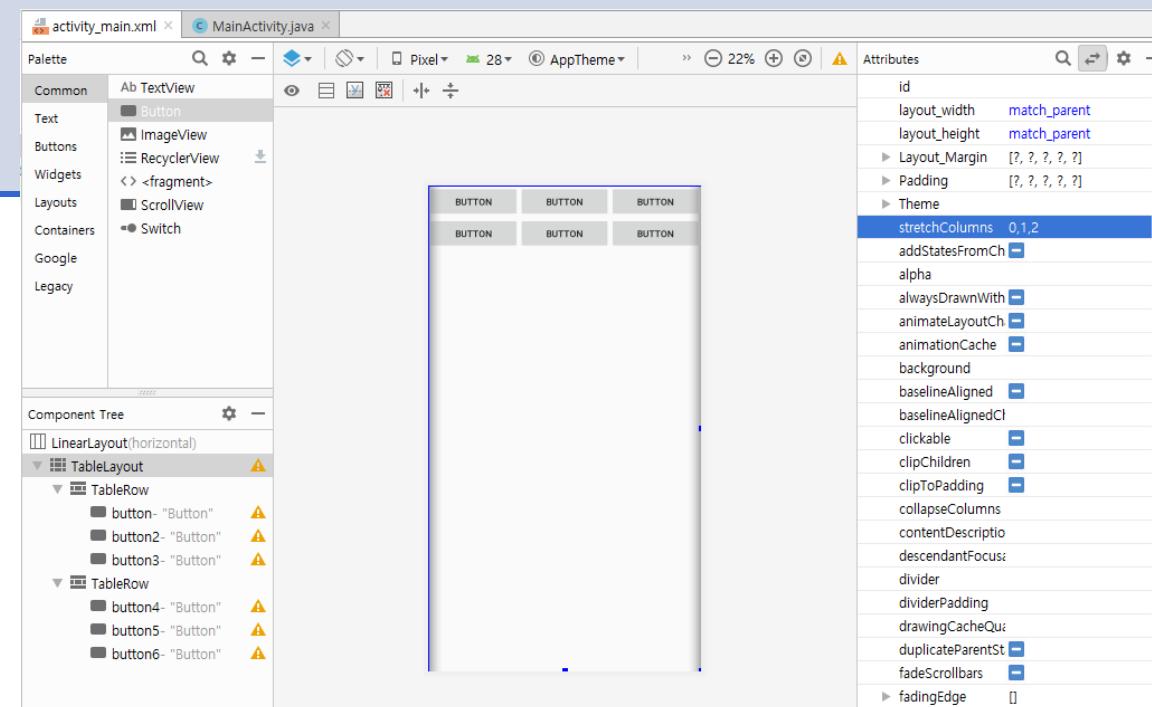
        <Button
            android:id="@+id/button3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button" />
    
```





stretchColumns 속성

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:stretchColumns="0,1,2" >
```





layout_span과 layout_column 속성의 사용

<EditText

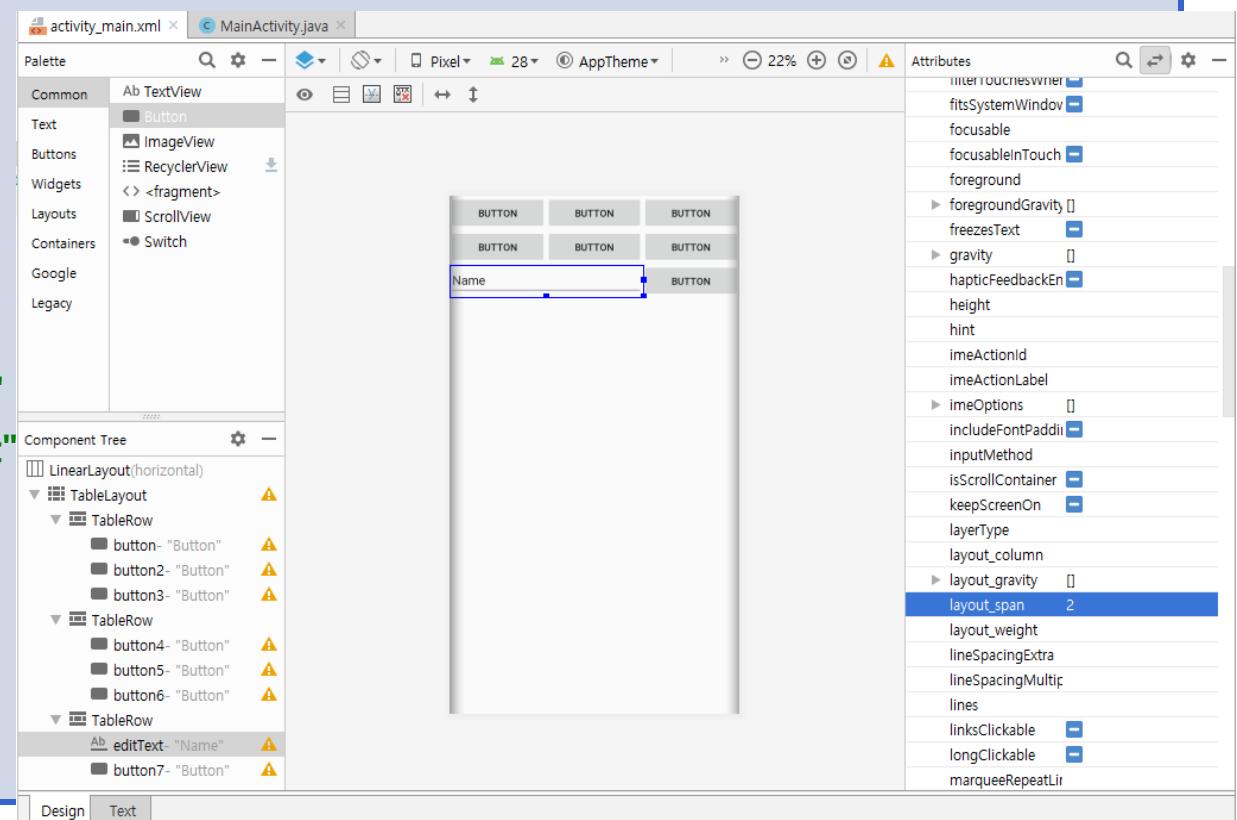
```
    android:id="@+id/editText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_span="3" />
```

...

<Button

```
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_column="2"
    android:text="아니오" />
```

...



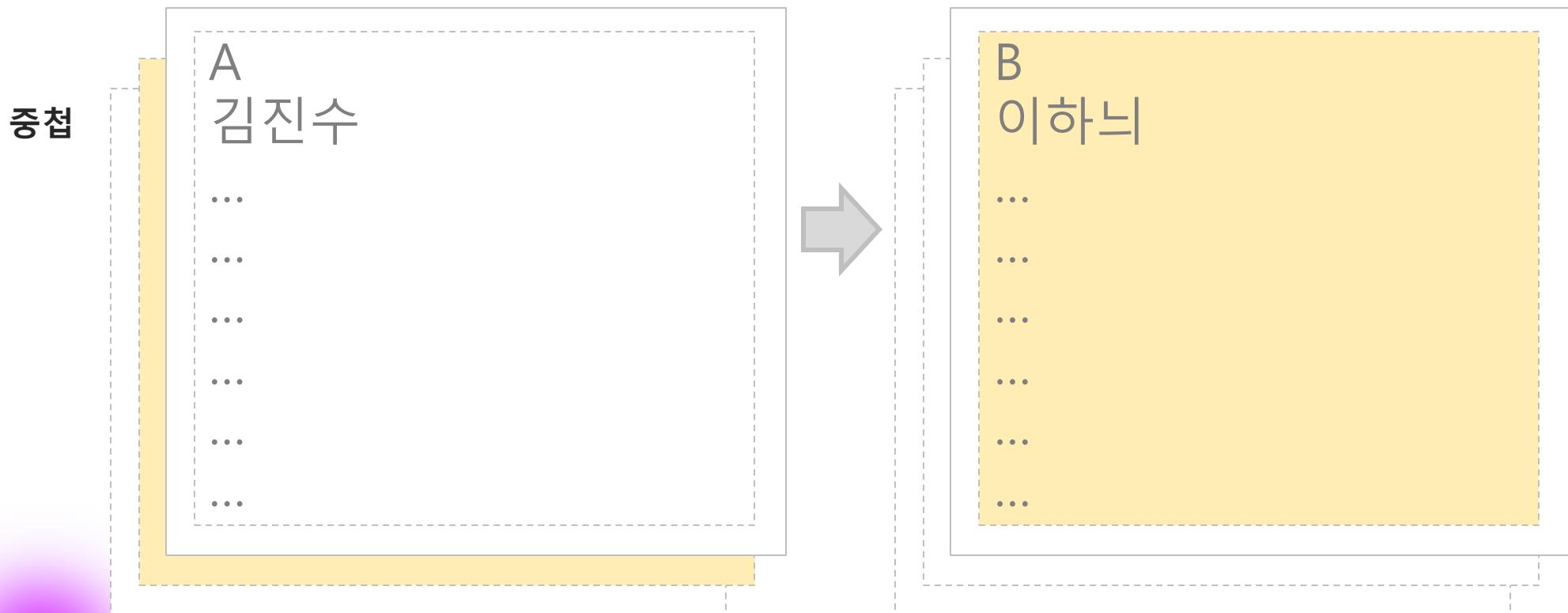


프레임 레이아웃과 뷰의 전환



프레임 레이아웃과 뷰의 전환

- 한 번에 하나의 뷰만 보여주며, 다른 뷰들은 그 아래에 중첩되어 쌓임
- 중첩되는 효과와 함께 뷰의 가시성(Visibility) 속성을 이용해 다양한 화면 구성이 가능함



[프레임 레이아웃과 가시성 속성의 사용]

뷰 A에서 뷰 B로 전환



가시성 속성 사용하기

- 사용 예 – XML 레이아웃

```
<LinearLayout  
    android:id="@+id/layout1"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical"  
    android:visibility="gone"  
>  
<TextView  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
/>  
</ScrollView>
```

- 사용 예 – 소스 코드

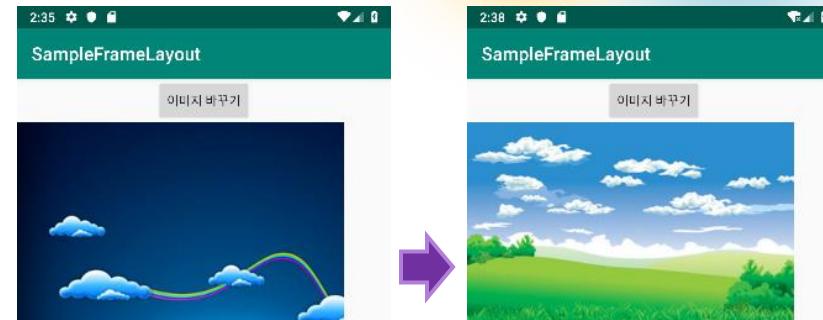
```
layout1.setVisibility(View.GONE);  
layout1.setVisibility(View.VISIBLE);  
layout1.setVisibility(View.INVISIBLE);
```



프레임 레이아웃과 뷰의 전환

뷰 전환 예제

- 프레임 레이아웃을 이용해 뷰를 중첩하여 만들기
- 버튼을 누르면 다른 이미지로 전환하기



XML 레이아웃

-레이아웃 코드 작성

메인 액티비티 코드

-메인 액티비티 코드 작성



프레임 레이아웃과 뷰의 전환 – XML 레이아웃

```
<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="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:text="이미지 바꾸기"  
>  
  
<FrameLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
>
```

1 전환 버튼

2 화면 채우기

Continued..



프레임 레이아웃과 뷰의 전환 – XML 레이아웃

```
<ImageView  
    android:id= "@+id/imageView"  
    android:layout_width= "wrap_content"  
    android:layout_height= "wrap_content"  
    android:src= "@drawable/dream01"  
    android:visibility= "invisible"  
/>  
<ImageView  
    android:id= "@+id/imageView2"  
    android:layout_width= "wrap_content"  
    android:layout_height= "wrap_content"  
    android:src= "@drawable/dream02"  
    android:visibility= "visible"  
/>  
</FrameLayout>  
</LinearLayout>
```

3 이미지 뷰 설정

4 이미지 뷰 설정



프레임 레이아웃과 뷰의 전환 – 메인 액티비티 코드

```
private void changeImage() {
    if (imageIndex == 0) {
        imageView1.setVisibility(View.VISIBLE);
        imageView2.setVisibility(View.INVISIBLE);
        imageIndex = 1;
    } else if (imageIndex == 1) {
        imageView1.setVisibility(View.INVISIBLE);
        imageView2.setVisibility(View.VISIBLE);
        imageIndex = 0;
    }
}
```

1 이미지 뷰 설정

2 이미지 뷰 설정

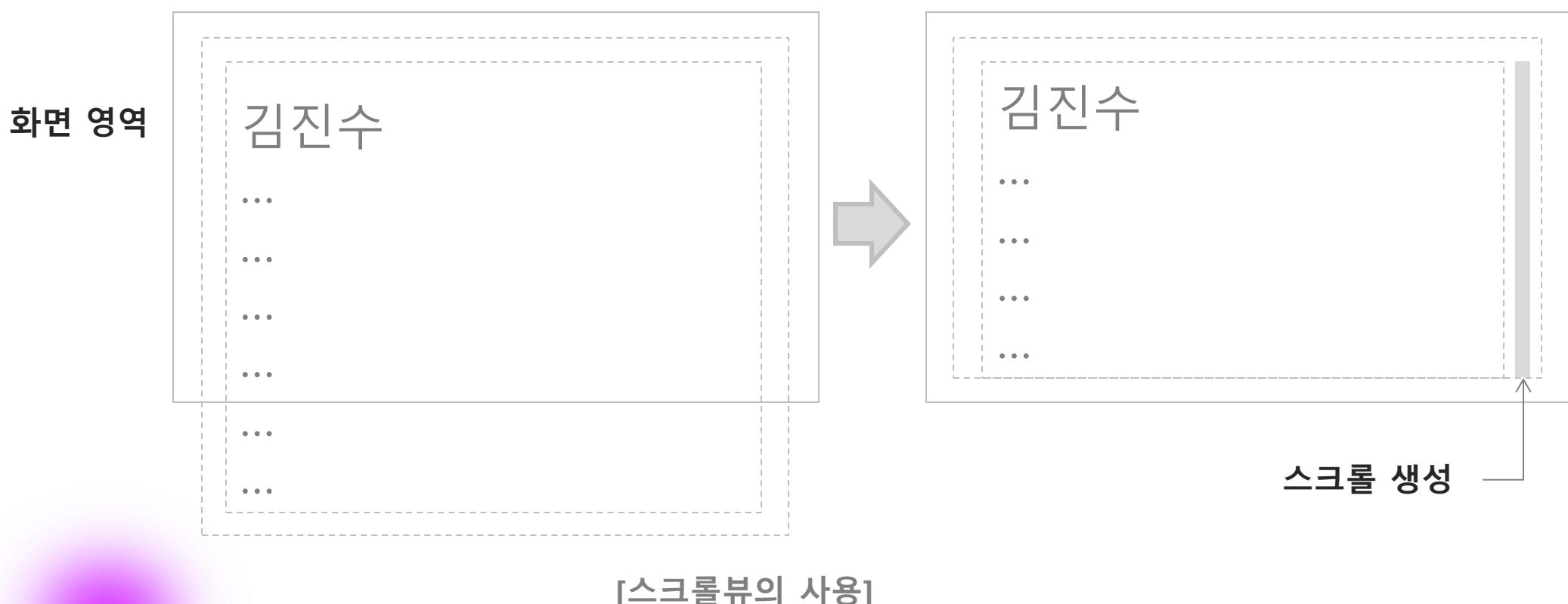


스크롤뷰 사용하기



스크롤뷰 사용하기

- 위젯의 내용이 화면 영역을 벗어나면 스크롤 표시가 자동으로 보임
- 스크롤뷰를 이용해 다른 뷰를 감싸주기만 하면 됨





스크롤뷰 사용하기

- 사용 예 – 텍스트뷰를 스크롤뷰로 감싸주는 경우

```
<ScrollView  
    android:layout_width= "match_parent"  
    android:layout_height= "match_parent"  
    >  
    <TextView  
        android:id= "@+id/textView"  
        android:layout_width= "match_parent"  
        android:layout_height= "match_parent"  
    />  
</ScrollView>
```



스크롤뷰 사용하기

스크롤뷰 예제

-주어진 영역보다 큰 이미지에 스크롤이 생기도록 만들기

XML 레이아웃

-레이아웃 코드 작성

메인 액티비티 코드

-메인 액티비티 코드 작성





스크롤뷰의 XML 레이아웃

```
    android:onClick="onButton1Clicked"  
    />  
  
<HorizontalScrollView  
    android:id="@+id/horScrollView"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent" >  
    <ScrollView  
        android:id="@+id/scrollView"  
        android:layout_width="match_parent"  
        android:layout_height="match_parent" >  
        <ImageView  
            android:id="@+id/imageView"  
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
        />
```

Continued..



스크롤뷰의 메인 액티비티 코드

```
...
scrollView = (ScrollView) findViewById(R.id.scrollView);
imageView = (ImageView) findViewById(R.id.imageView);

...
scrollView.setHorizontalScrollBarEnabled(true);
Resources res = getResources();
BitmapDrawable bitmap = (BitmapDrawable)
    res.getDrawable(R.drawable.system_architecture);
int bitmapWidth = bitmap.getIntrinsicWidth();
int bitmapHeight = bitmap.getIntrinsicHeight();
imageView.setImageDrawable(bitmap);
imageView.getLayoutParams().width = bitmapWidth;
imageView.getLayoutParams().height = bitmapHeight;
...

```

1 객체 참조

2 기능 설정

3 이미지 참조

4 크기 설정

Continued..



기본 위젯과 드로어블에 대해 이해하기



- 1 기본 위젯 다시 한 번 자세히 공부하기
- 2 드로어블 만들기
- 3 이벤트 처리 이해하기
- 4 토스트, 스낵바 그리고 대화상자 사용하기
- 5 프로그레스바 사용하기



버튼이나 텍스트뷰를 다루는 방법을 좀 더 알고 싶어요

- 기본 위젯들 살펴보기



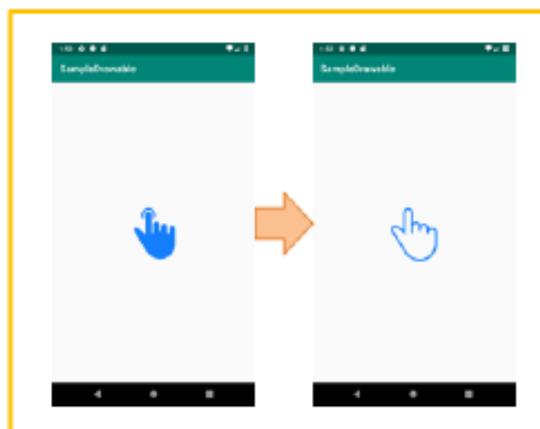
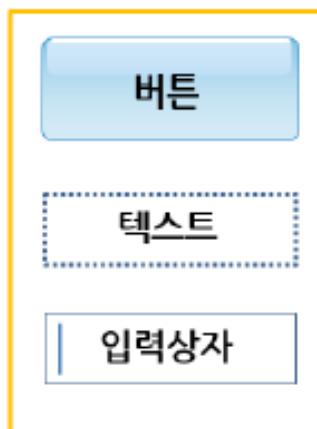
드로어블은 어떻게 만들 수 있나요?

- 배경 설정하기
- 드로어블 만들기



사용자에게 간단한 정보를 보여주고 싶어요

- 이벤트 이해하기
- 토스트, 스낵바 그리고 대화상자 사용하기
- [프로그래스바](#) 사용하기





1.

기본 위젯



기본 위젯 – 텍스트뷰의 속성

- **텍스트 뷰**

- **text** : 텍스트 뷰에 보이는 문자열을 설정할 수 있음
- **textColor** : 텍스트뷰에서 표시하는 문자열의 색상을 설정함
 - : 색상 설정은 "#AARRGGBB" 포맷을 일반적으로 사용(Alpha, Red, Green, Blue)
 - : 투명도를 나타내는 Alpha(색상만 표현할 때 - "FF", 투명 - "00", 반투명 - "88")
- **textSize** : 텍스트뷰에서 표시하는 문자열의 크기를 설정함
("dp"나 "sp" 또는 "px" 등의 단위 값을 사용함)
- **textStyle** : 텍스트뷰에서 표시하는 문자열의 스타일 속성을 설정함
("normal", "bold", "italic" 등의 값을 지정할 수 있음)
- **typeFace** : 텍스트뷰에서 표시하는 문자열의 폰트를 설정함
("normal", "sans", "serif", "monospace")
- **maxLines="1"** : 텍스트뷰에서 표시하는 문자열이 한 줄로만 표시되도록 설정함



기본 위젯 – 텍스트뷰의 속성 사용

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="vertical"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
>  
<TextView  
    android:id="@+id/TextView"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:background="#ff000055"  
    android:padding="3px"  
    android:text="사용자 이름을 입력하세요. 이름은 한 줄로 표시됩니다."  
    android:textSize="22sp"  
    android:textStyle="bold"  
    android:textColor="#88ff8888"  
    android:singleLine="true"  
    android:gravity="center"  
>  
</LinearLayout>
```

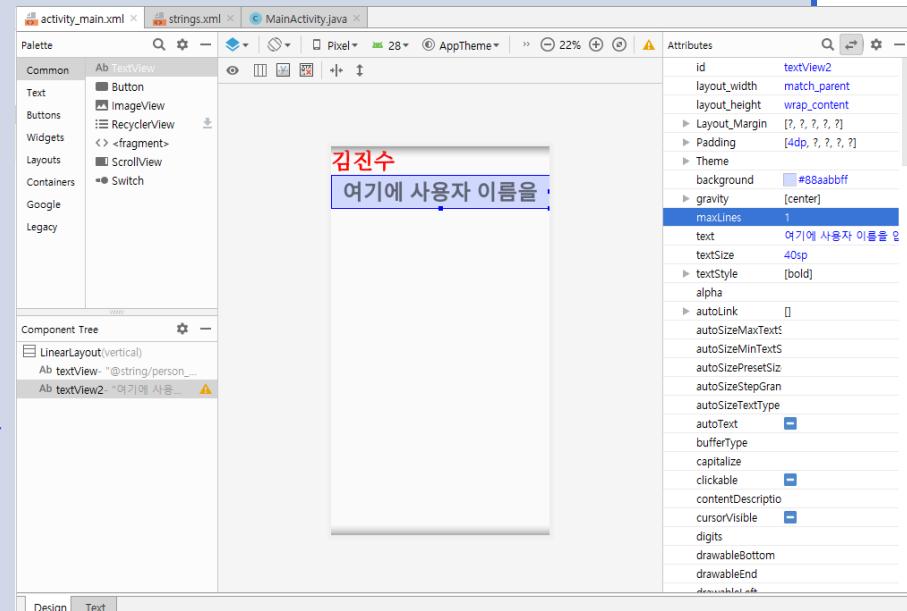
1 배경색 설정

2 크기 설정

3 스타일 설정

4 색상 설정

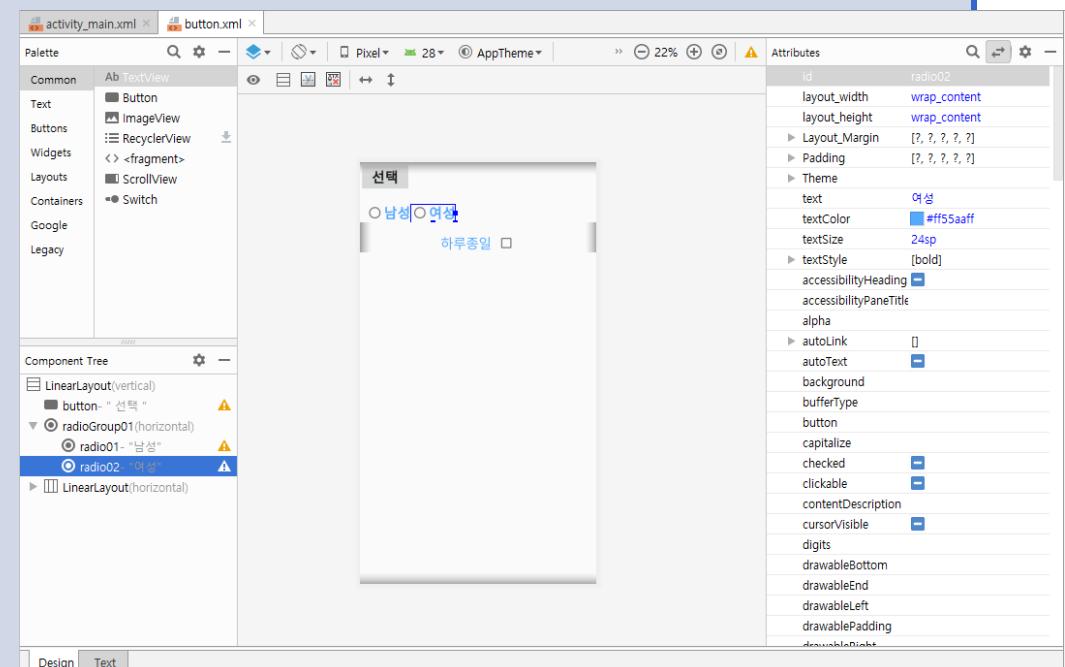
5 한 줄 설정





기본 위젯 – 버튼의 속성 사용

```
<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/btnExit"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="선택"  
    android:textSize="24dp"  
    android:textStyle="bold"  
    android:gravity="center"  
>
```



Continued..



기본 위젯 – 버튼의 속성 사용 (계속)

```
<RadioGroup  
    android:id= "@+id/radioGroup01"  
    android:layout_width= "wrap_content"  
    android:layout_height= "wrap_content"  
    android:orientation= "horizontal"  
    android:paddingLeft= "5dp"  
    android:paddingRight= "5dp"  
>  
<RadioButton  
    android:id= "@+id/radio01"  
    android:layout_width= "wrap_content"  
    android:layout_height= "wrap_content"  
    android:layout_weight= "1"  
    android:text= "남성"  
    android:textColor= "#ffaaff10"  
    android:textStyle= "bold"  
    android:textSize= "24dp"  
>
```

Continued..



기본 위젯 – 버튼의 속성 사용 (계속)

```
<RadioButton  
    android:id="@+id/radio02"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_weight="1"  
    android:text="여성"  
    android:textColor="#ffaaff10"  
    android:textStyle="bold"  
    android:textSize="24dp"  
/>  
</RadioGroup>  
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:gravity="center_vertical|center_horizontal"  
    android:paddingTop="10dp"  
>
```

Continued..



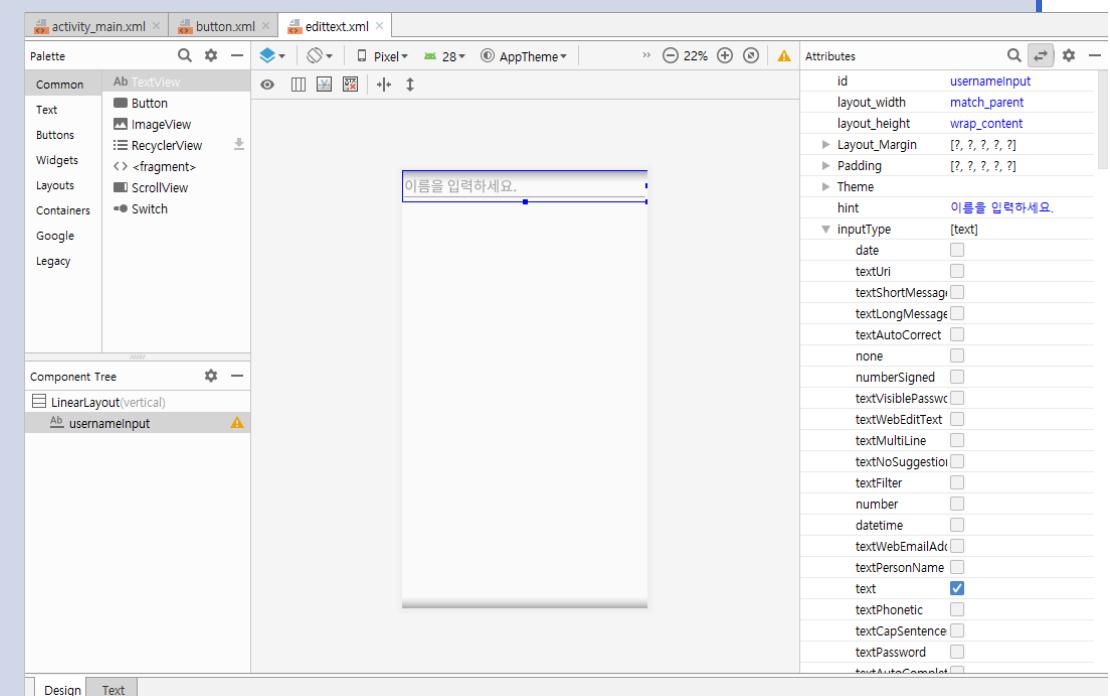
기본 위젯 – 버튼의 속성 사용 (계속)

```
<TextView  
    android:layout_width= "wrap_content"  
    android:layout_height= "wrap_content"  
    android:text= "하루종일"  
    android:textSize= "24dp"  
    android:paddingRight= "10dp"  
    android:textColor= "#ffaaff10"  
/>  
<CheckBox  
    android:id= "@+id/allDay"  
    android:layout_width= "wrap_content"  
    android:layout_height= "wrap_content"  
/>  
</LinearLayout>  
</LinearLayout>
```



기본 위젯 – 입력상자의 속성 사용

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="vertical"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
>  
  
<EditText  
    android:id="@+id/editText1"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:textSize="18sp"  
    android:autoText="true"  
    android:capitalize="words"  
    android:hint="이름을 입력하세요."  
>  
</EditText>  
</LinearLayout>
```

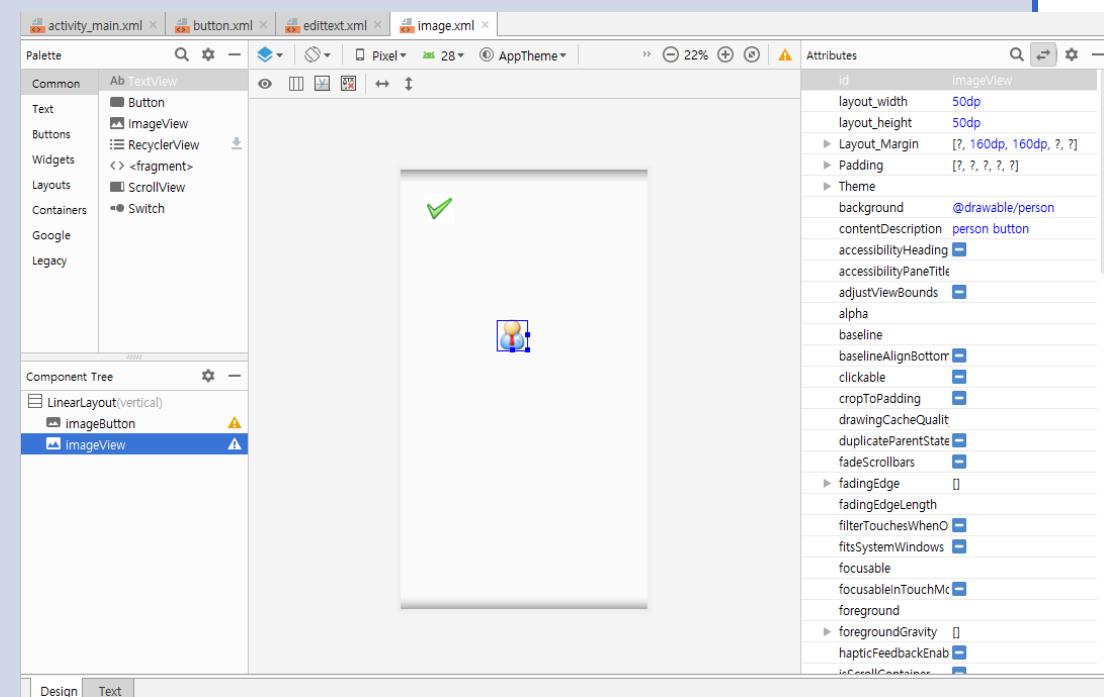




기본 위젯 – 이미지뷰의 속성 사용

```
<ImageButton
```

```
    android:id="@+id/imageButton01"
    android:background="@drawable/ok_btn"
    android:layout_width="24dp"
    android:layout_height="24dp"
    ...
/>
```



```
<ImageView
```

```
    android:id="@+id/imageView01"
    android:background="@drawable/person"
    android:layout_width="32dp"
    android:layout_height="32dp"
    ...
/>
```



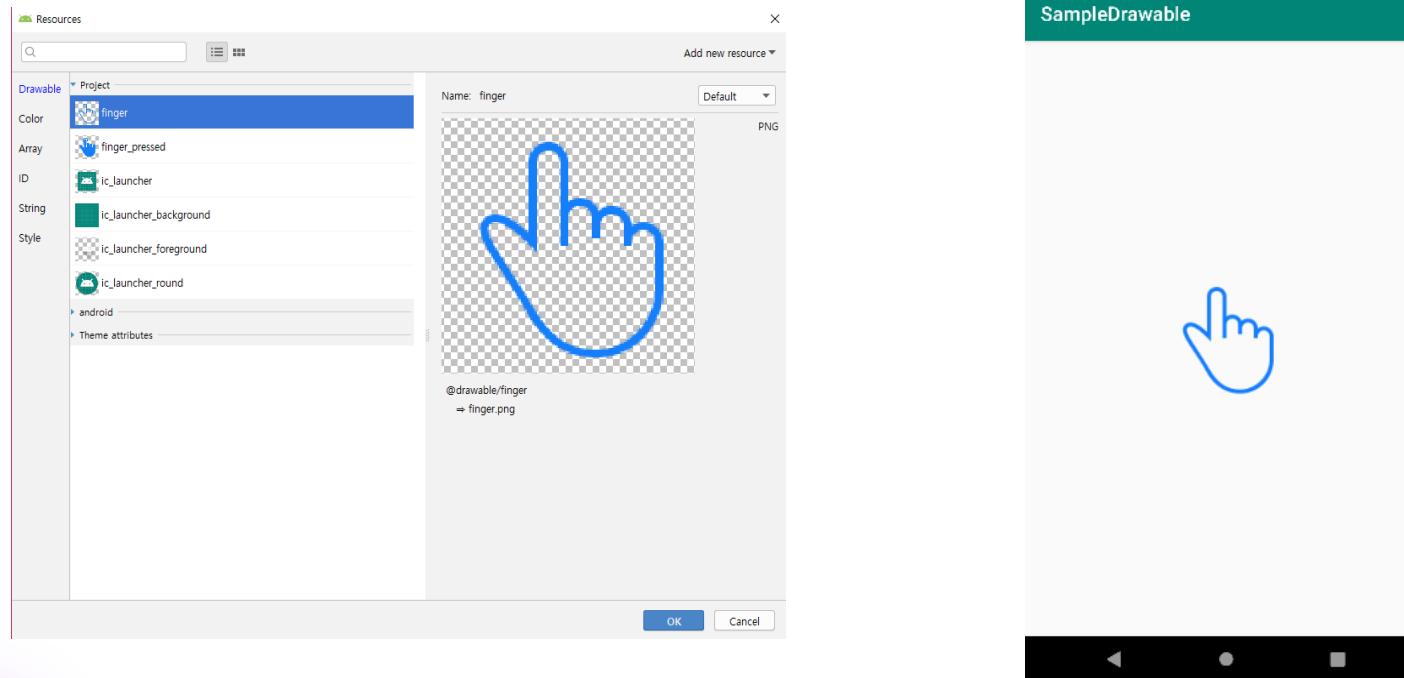
2.

드로어블 만들기



뷰의 배경으로 이미지 설정하기

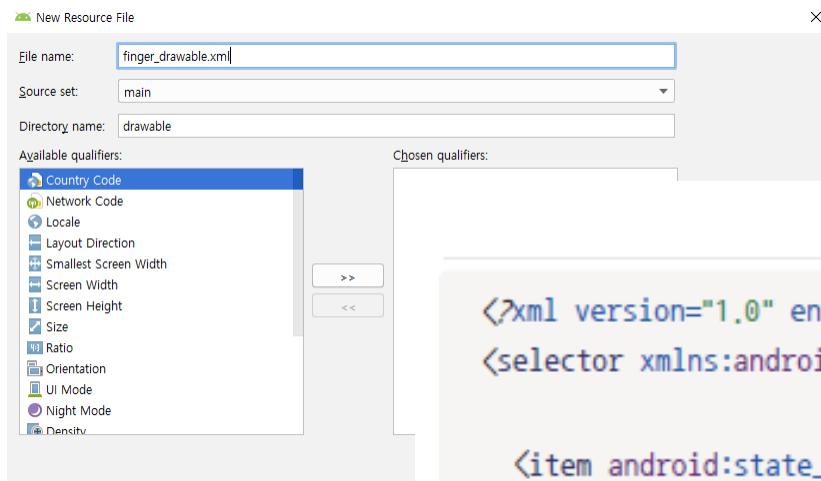
- 뷰의 `background` 속성으로 이미지 설정
- `/app/res/drawable` 폴더 안에 넣어둔 이미지를 설정할 수 있음





상태 드로어를 만들기

- 상태에 따라 다른 이미지를 보여줄 수 있음
- ./app/res/drawable 폴더 안에 xml 파일을 만들고 그 안에 정의



참조파일 SampleDrawable>/app/res/drawable/finger_drawable.xml

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

    <item android:state_pressed="true"
          android:drawable="@drawable/finger_pressed" />

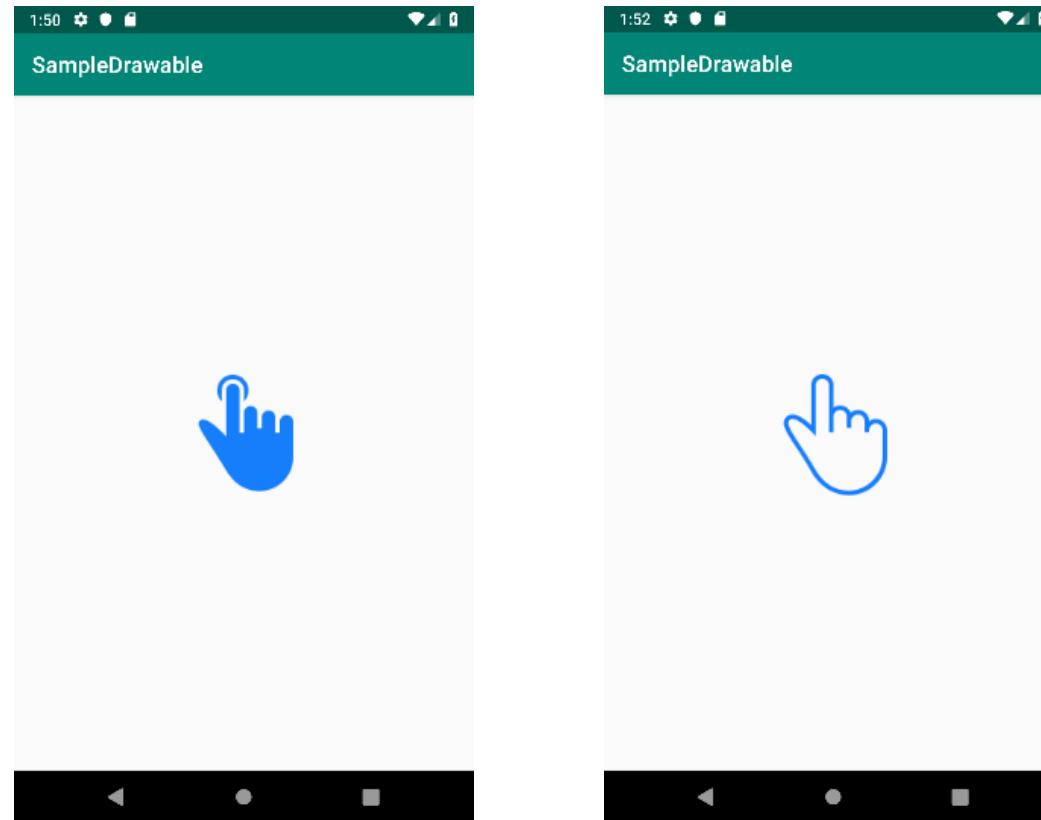
    <item android:drawable="@drawable/finger" />

</selector>
```



상태 드로어블 만들기

- 뷰의 background 속성 값으로 xml 설정
- 앱 실행 후 뷰를 누르면 눌린 상태에서 다른 이미지로 변경됨





셰이프 드로어블 만들기

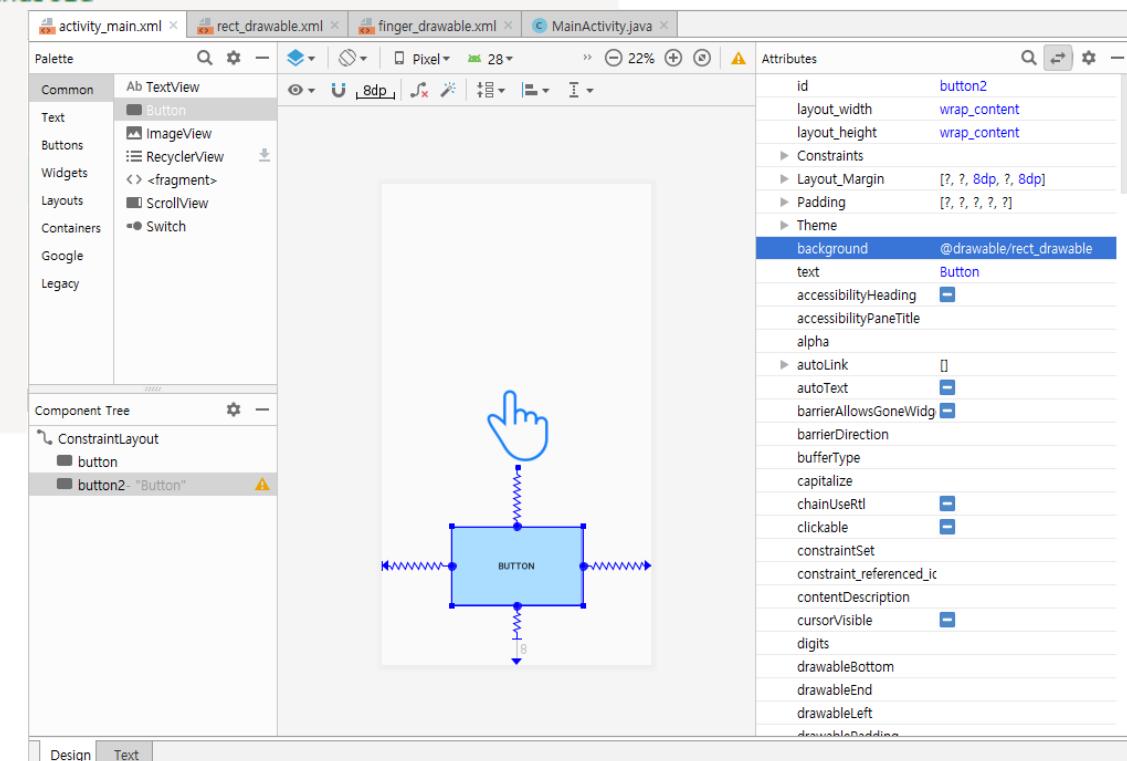
- 도형을 그릴 수 있음
- /app/res/drawable 폴더 안에 xml 파일을 만들고 그 안에 정의

참조파일 SampleDrawable>/app/res/drawable/rect_drawable.xml

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

    <size android:width="200dp" android:height="120dp"/>
    <stroke android:width="1dp" android:color="#0000ff"/>
    <solid android:color="#aaddff" />
    <padding android:bottom="1dp" />

</shape>
```





셰이프 드로어블 만들기

- 화면 전체의 배경으로 그레이데이션 적용
- 최상위 레이아웃의 background 속성 값으로 설정

참조파일 SampleDrawable>/app/res/drawable/back_drawable.xml

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

    <gradient
        android:startColor="#7288DB"
        android:centerColor="#3250B4"
        android:endColor="#254095"
        android:angle="90"
        android:centerY="0.5"
    />

    <corners android:radius="2dp" />

</shape>
```

The screenshot shows the Android Studio interface with the 'Design' tab selected. On the left, the 'Palette' sidebar lists various UI components like TextView, Button, ImageView, etc. In the center, a blue gradient shape with rounded corners and a central white button is displayed. A hand cursor is hovering over the button. On the right, the 'Attributes' panel shows the XML code for the shape, specifically the 'background' attribute set to '@drawable/back_drawable'. Other attributes like 'id', 'layout_width', and 'layout_height' are also listed.



셰이프 드로어블 만들기

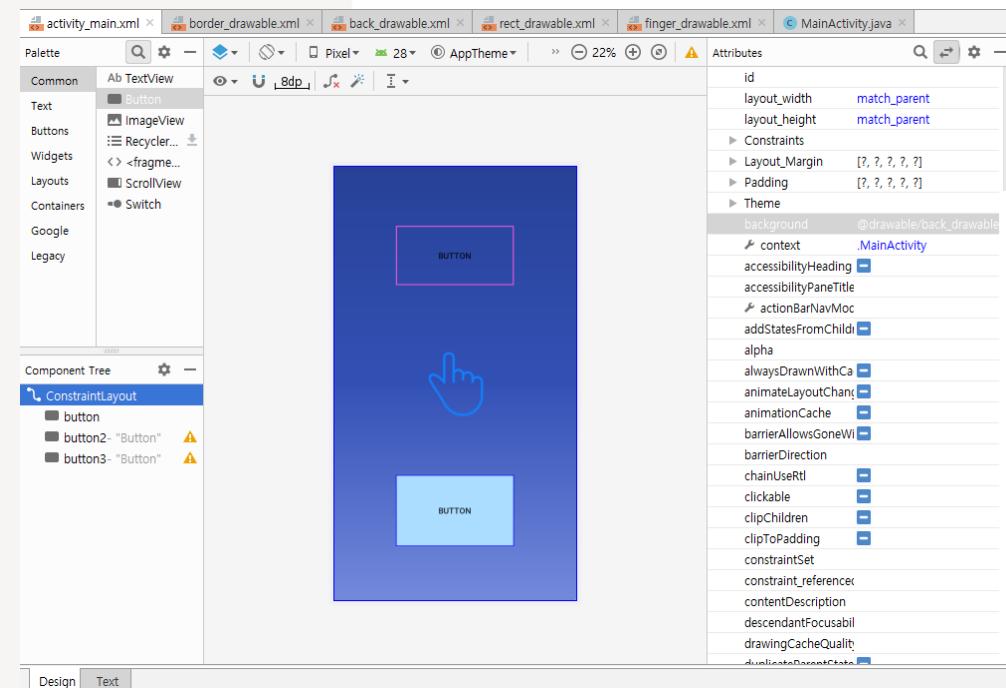
- 테두리만 있는 버튼 만들기
- 버튼의 background 속성 값으로 설정

참조파일 SampleDrawable>/app/res/drawable/border_drawable.xml

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

    <item>
        <shape android:shape="rectangle">
            <stroke android:width="1dp" android:color="#BE55DA" />
            <solid android:color="#00000000" />
            <size android:width="200dp" android:height="100dp" />
        </shape>
    </item>

    <item android:top="1dp" android:bottom="1dp"
          android:right="1dp" android:left="1dp">
        <shape android:shape="rectangle">
            <stroke android:width="1dp" android:color="#FF55DA" />
            <solid android:color="#00000000" />
        </shape>
    </item>
</layer-list>
```



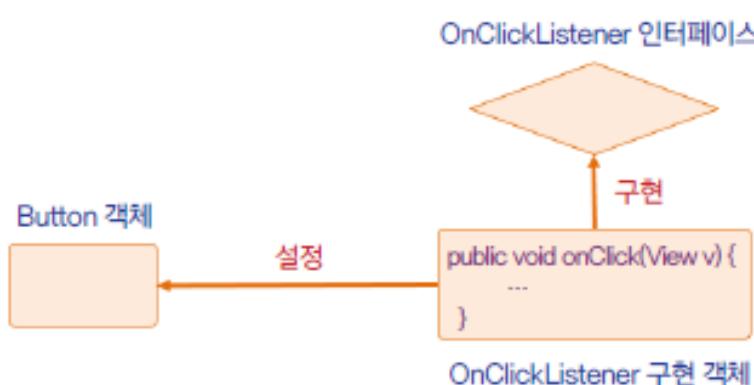


3.

이벤트 처리 이해하기



뷰의 이벤트 처리하기



[버튼에 OnClickListener를 설정할 때의 패턴]

뷰를 상속할 때 이벤트 처리를 위한 메소드 재정의

boolean onTouchEvent (MotionEvent event)

boolean onKeyDown (int keyCode, KeyEvent event)

boolean onKeyUp (int keyCode, KeyEvent event)

뷰 객체에 전달되는 이벤트를 처리하기 위한 리스너 설정

View.OnTouchListener : boolean onTouch (View v, MotionEvent event)

View.OnKeyListener : boolean onKeyDown (View v, int keyCode, KeyEvent event)

View.OnClickListener : void onClick (View v)

View.OnFocusChangeListener : void onFocusChange (View v, boolean hasFocus)



대표적인 이벤트

.터치 이벤트

- 화면을 손가락으로 누를 때 발생하는 이벤트

.키 이벤트

- 키패드나 하드웨어 버튼을 누를 때 발생하는 이벤트

.제스처 이벤트

- 터치 이벤트 중에서 일정 패턴을 만들어 내는 이벤트

.포커스

- 뷰마다 순서대로 주어지는 포커스

.화면 방향 변경

- 화면의 방향이 가로/세로로 바뀜에 따라 발생하는 이벤트



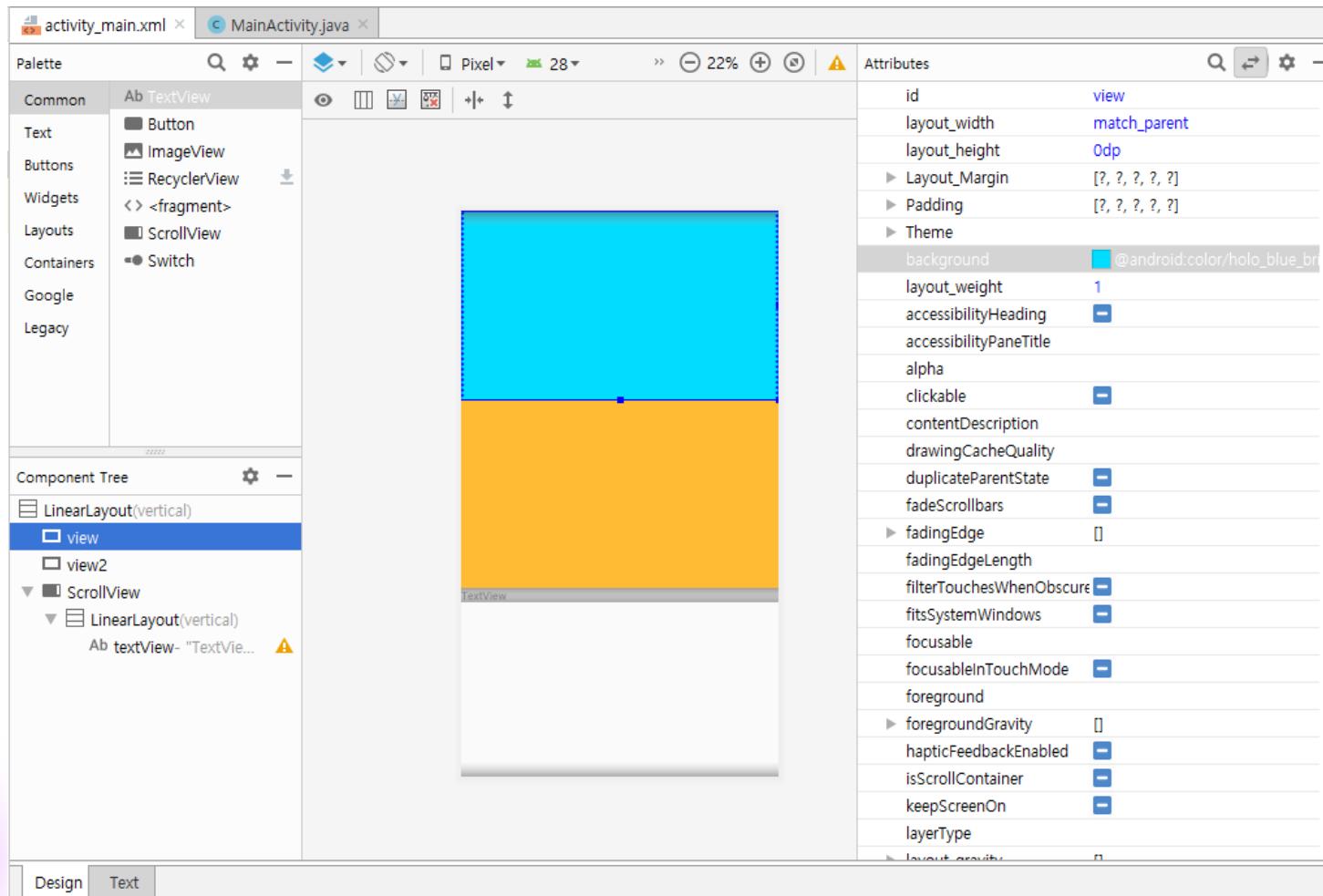
제스처를 통해 처리할 수 있는 이벤트

메소드	이벤트 유형
onDown()	- 화면이 눌렸을 경우
onShowPress()	- 화면이 눌렸다 떼어지는 경우
onSingleTapUp()	- 화면이 한 손가락으로 눌렸다 떼어지는 경우
onSingleTapConfirmed()	- 화면이 한 손가락으로 눌려지는 경우
onDoubleTap()	- 화면이 두 손가락으로 눌려지는 경우
onDoubleTapEvent()	- 화면이 두 손가락으로 눌려진 상태에서 떼거나 이동하는 등 세부적인 액션을 취하는 경우
onScroll()	- 화면이 눌린 채 일정한 속도화 방향으로 움직였다 떼는 경우
onFling()	- 화면이 눌린 채 가속도를 붙여 손가락을 움직였다 떼는 경우
onLongPress()	- 화면을 손가락으로 오래 누르는 경우



터치 이벤트 처리하기

- SampleEvent 프로젝트를 만들고 XML 레이아웃 구성





메인 액티비티 코드 만들기

```
View view = findViewById(R.id.view);
view.setOnTouchListener(new View.OnTouchListener() {
    @Override
    public boolean onTouch(View view, MotionEvent motionEvent) {
        int action = motionEvent.getAction();
        float curX = motionEvent.getX();
        float curY = motionEvent.getY();

        if (action == MotionEvent.ACTION_DOWN) {
            println("손가락 눌림 : " + curX + ", " + curY);
        } else if (action == MotionEvent.ACTION_MOVE) {
            println("손가락 움직임 : " + curX + ", " + curY);
        } else if (action == MotionEvent.ACTION_UP) {
            println("손가락 뗄 : " + curX + ", " + curY);
        }

        return true;
    }
});
```

Continued..



앱 실행 결과

- 가장 위쪽에 있는 부분을 터치했을 때 가장 아래쪽에 보이는 로그 메시지 확인





제스처 이벤트 처리하기

- **GestureDetector** 객체 만들고 터치 이벤트 발생 시 해당 객체 전달

```
public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY) {  
    println("nonFling tx = " + velocityX + "ty=" + velocityY);  
    return super.onFling(e1, e2, velocityX, velocityY);  
}
```

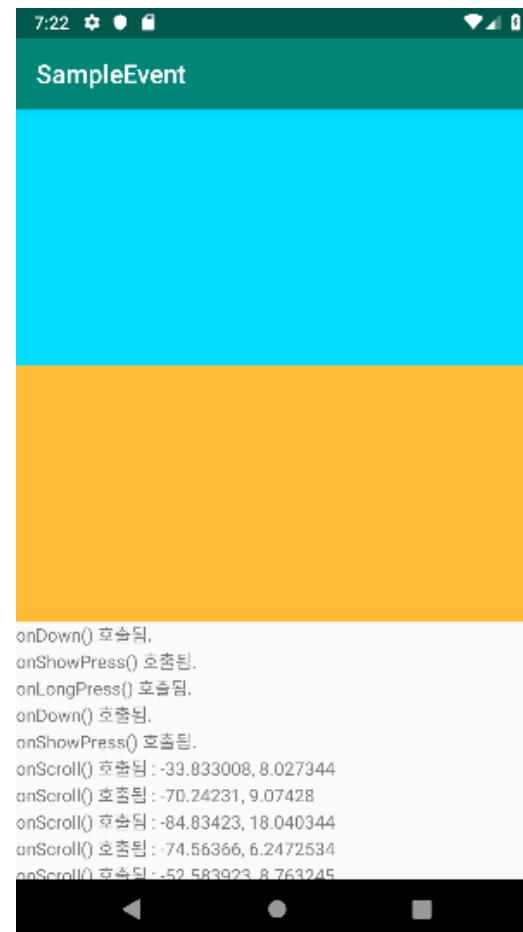
```
public boolean onScroll(MotionEvent e1, MotionEvent e2, float distanceX, float distanceY) {  
    println("nonScroll tx = " + distanceX + "ty = " + distanceY);  
    return super.onScroll(e1, e2, distanceX, distanceY);  
}
```

Continued..



앱 실행 결과

- 중간 부분을 터치했을 때 가장 아래쪽에 보이는 로그 메시지 확인





키 입력 이벤트 처리하기

[키를 눌렀을 때 전달되는 대표적인 키값]

키 코드	설 명
KEYCODE_DPAD_LEFT	- 왼쪽 화살표
KEYCODE_DPAD_RIGHT	- 오른쪽 화살표
KEYCODE_DPAD_UP	- 위쪽 화살표
KEYCODE_DPAD_DOWN	- 아래쪽 화살표
KEYCODE_DPAD_CENTER	- [중앙] 버튼
KEYCODE_CALL	- [통화] 버튼
KEYCODE_ENDCALL	- [통화 종료] 버튼
KEYCODE_HOME	- [홈] 버튼
KEYCODE_BACK	- [뒤로 가기] 버튼
KEYCODE_VOLUME_UP	- [소리 크기 증가] 버튼
KEYCODE_VOLUME_DOWN	- [소리 크기 감소] 버튼
KEYCODE_0 ~ KEYCODE_9	- 숫자 0부터 9까지의 키값
KEYCODE_A ~ KEYCODE_Z	- 알파벳 A부터 Z까지의 키값

뷰를 상속할 때 키 이벤트 처리를 위한 메소드 재정의

```
boolean onKeyDown (int keyCode, KeyEvent event)  
boolean onKey (View v, int keyCode, KeyEvent event)
```



BACK 버튼 이벤트 처리하기

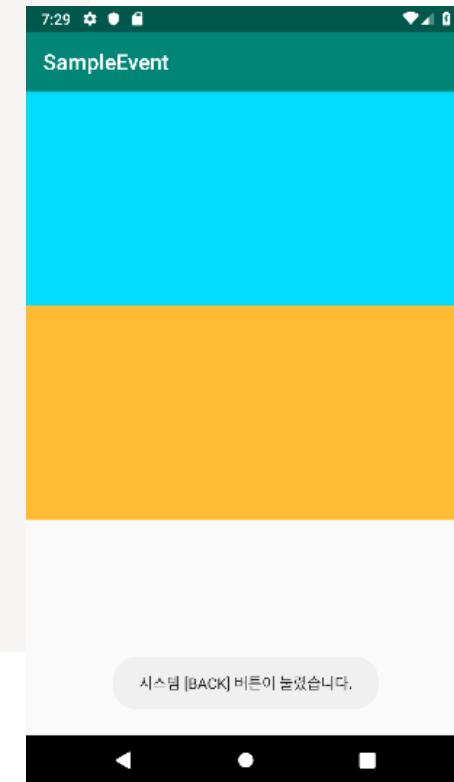
- 시스템 BACK 키를 눌렀을 때 동작할 코드 정의
- onBackPressed 메소드 재정의 또는 onKeyDown 메소드 안에서 정의

참조파일 SampleEvent>/app/java/org.techtown.sampleevent/MainActivity.java

증략...

```
@Override  
public boolean onKeyDown(int keyCode, KeyEvent event) {  
  
    if(keyCode == KeyEvent.KEYCODE_BACK) {  
        Toast.makeText(this, "시스템 [BACK] 버튼이 눌렸습니다.",  
                    Toast.LENGTH_LONG).show();  
  
        return true;  
    }  
  
    return false;  
}
```

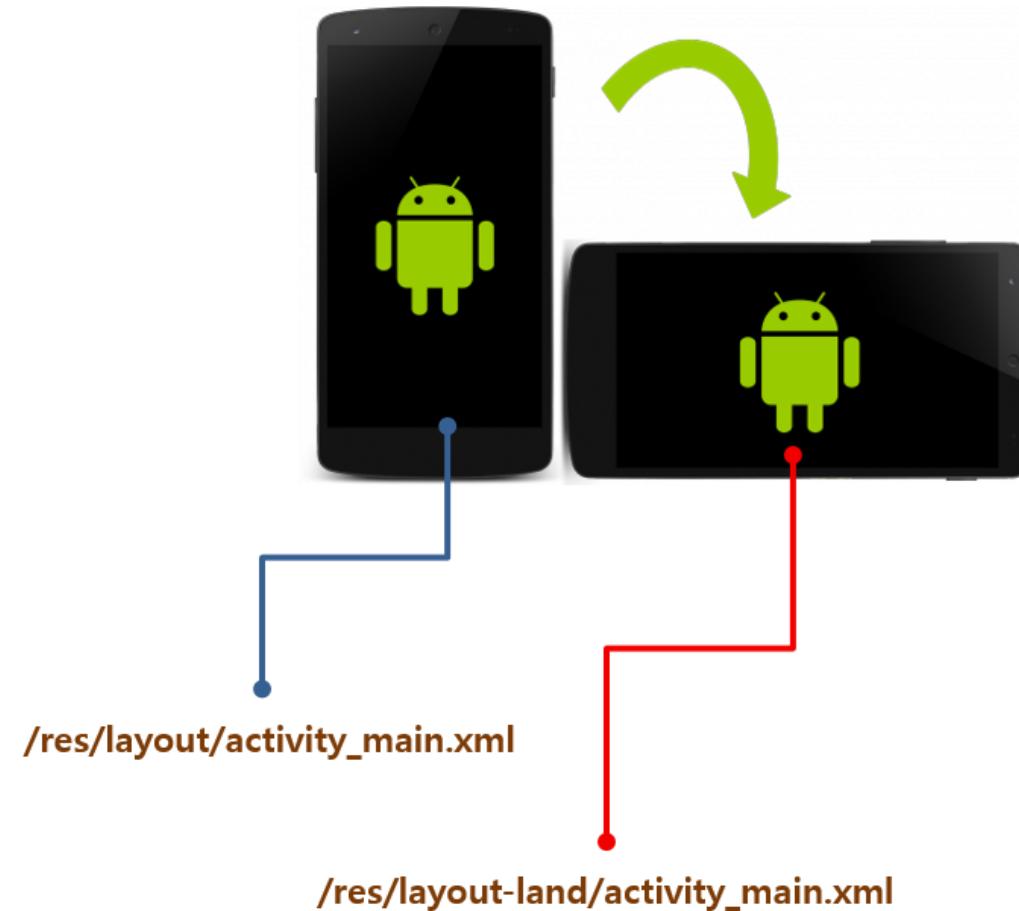
증략...





단말 방향 전환

- 병렬 리소스 로딩 방식 사용
- [res] 폴더 안에 [layout] 폴더와 [layout-land] 폴더 생성





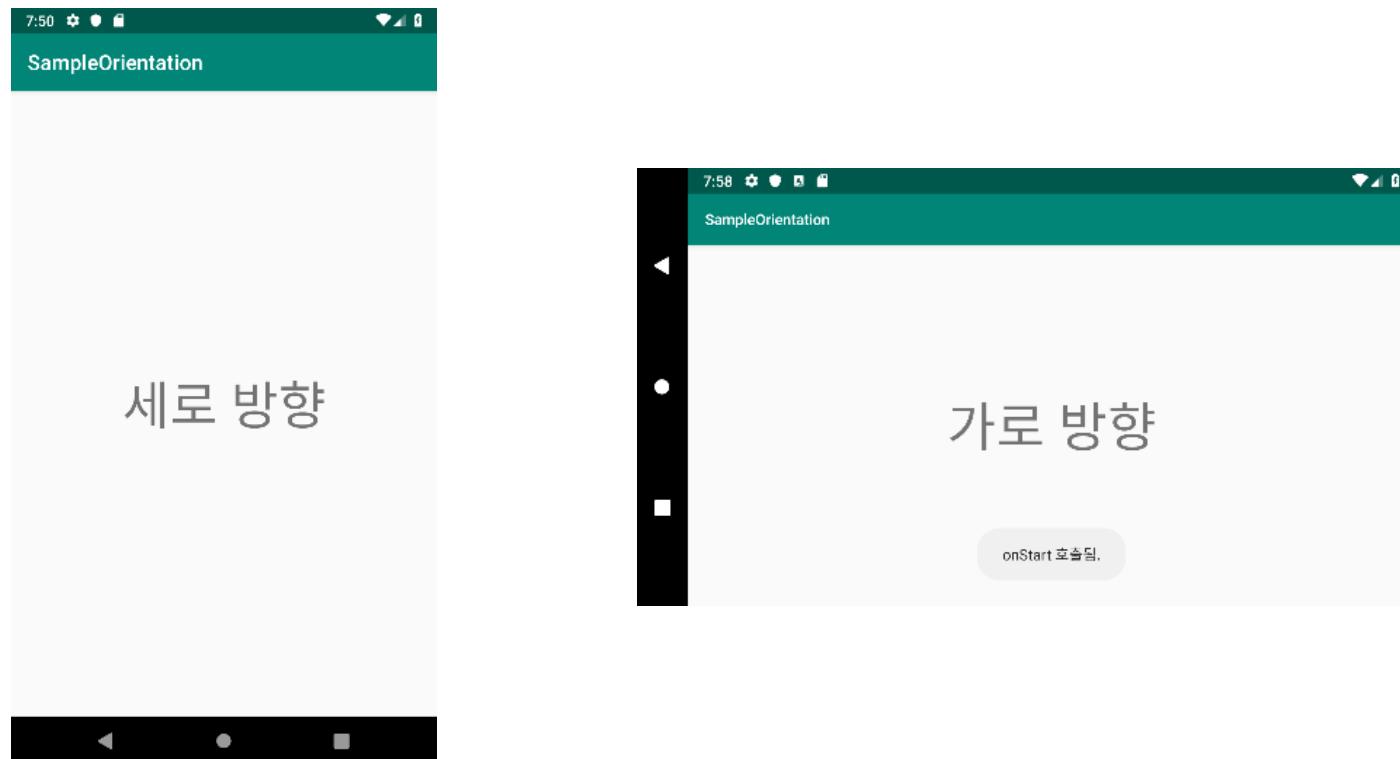
소스 코드에서 토스트 메시지 표시

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        showToast("onCreate 호출됨.");  
    }  
  
    @Override  
    protected void onStart() {  
        super.onStart();  
        showToast("onStart 호출됨.");  
    }  
  
    @Override  
    protected void onStop() {  
        super.onStop();  
        showToast("onStop 호출됨.");  
    }  
}
```



앱 실행 결과

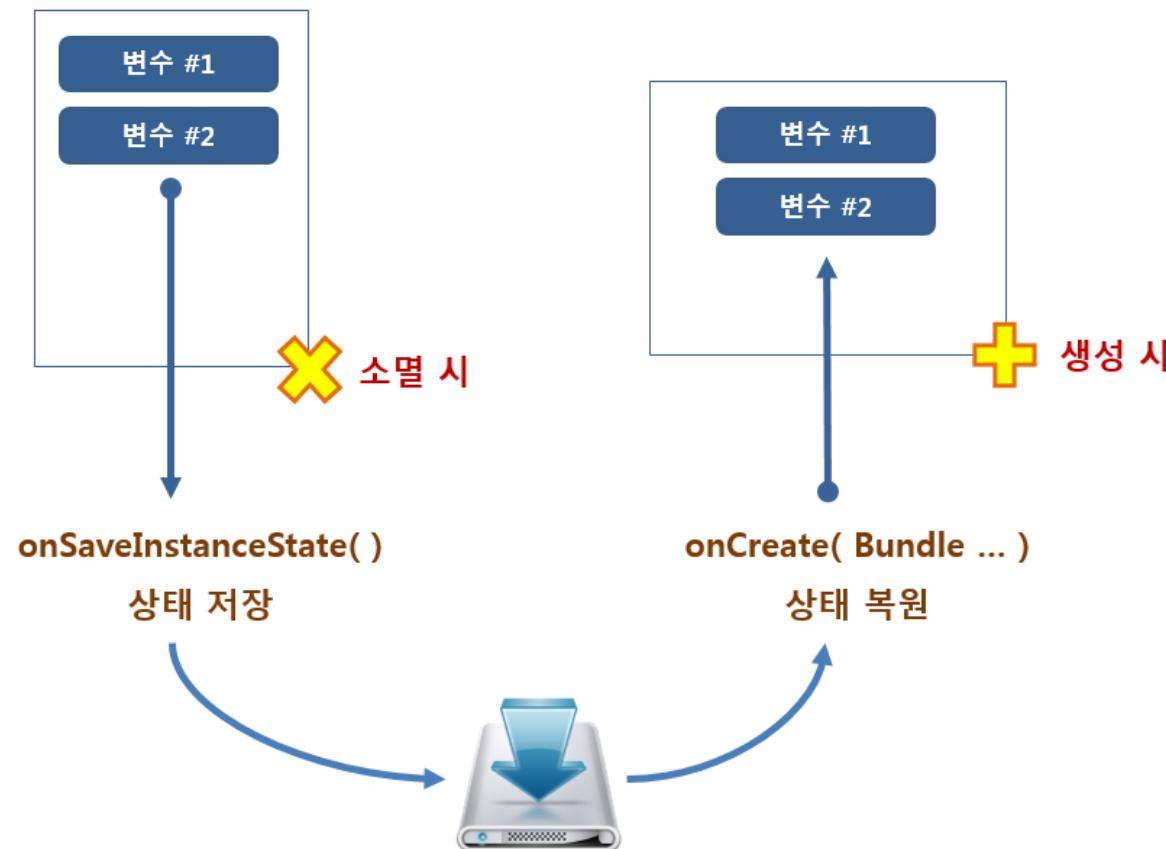
- 단말 방향 바꾸었을 때 액티비티가 새로 생성되므로 토스트 메시지 표시됨





단말 방향 전환 시 상태 저장과 복원

- onSaveInstanceState 메소드에서 상태 저장했다가 onCreate의 파라미터를 이용해 복원





단말 방향전환 상태 저장 예제

단말 방향전환 상태저장 예제

-단말 방향이 가로와 세로로 바뀌었을 때 상태 저장과 복원



세로 방향



가로 방향

매니페스트 속성 추가

-매니페스트의 액티비티 속성
추가

메인 액티비티 코드 작성

-가로와 세로 방향으로 바뀌었을
때 처리 코드 작성



XML 레이아웃 구성

- 입력상자와 버튼 추가

The screenshot shows the Android Studio interface with the XML Layout Editor open. The layout consists of a ConstraintLayout containing a TextView labeled "세로 방향", an editText with the placeholder "Plain Text", and a button labeled "확인". The editText has its inputType set to "textPersonName". The Attributes panel on the right lists various properties for the editText, such as id (editText), layout_width (wrap_content), layout_height (wrap_content), and inputType (textPersonName). The Component Tree panel shows the hierarchy of the components within the ConstraintLayout.

layout#activity_main.xml x land#activity_main.xml x MainActivity.java

Palette Attributes

Common Text Buttons Widgets Layouts Containers Google Legacy

Component Tree

- ConstraintLayout
- TextView- "세로 방향"
- editText(Plain Text)
- button- "확인"

Attributes

- id: editText
- layout_width: wrap_content
- layout_height: wrap_content
- Constraints
- Layout_Margin: [?, ?, 16dp, ?, ?]
- Padding: [?, ?, ?, ?, ?]
- Theme
- ems: 10
- inputType: [textPersonName]
- accessibilityHeading
- accessibilityPaneTitle
- alpha
- autoLink
- autoText
- background
- barrierAllowsGoneWidget:
- barrierDirection
- bufferType
- capitalize
- chainUseRtl
- clickable
- constraintSet
- constraint_referenced_ids
- contentDescription
- cursorVisible
- digits
- drawableBottom
- drawableEnd
- drawableLeft

Design Text



소스 코드에서 값의 저장과 복원

```
protected void onCreate(Bundle savedInstanceState) {  
    ...  
  
    if (savedInstanceState != null) {  
        name = savedInstanceState.getString("name");  
  
        Toast.makeText(getApplicationContext(), "값을 복원했습니다 :" + name, Toast.LENGTH_LONG).show();  
    }  
}  
  
@Override  
protected void onSaveInstanceState(Bundle outState) {  
    super.onSaveInstanceState(outState);  
  
    outState.putString("name", name);  
}  
...
```



단말 방향 전환 시 액티비티 유지

- AndroidManifest.xml 파일에 configChanges 속성 설정

```
...  
<activity android:name=".MainActivity"  
    android:configChanges="orientation/screenSize/keyboardHidden"  
    >  
    <intent-filter>  
        <action android:name="android.intent.action.MAIN" />  
        <category android:name="android.intent.category.LAUNCHER" />  
    </intent-filter>  
</activity>  
...
```



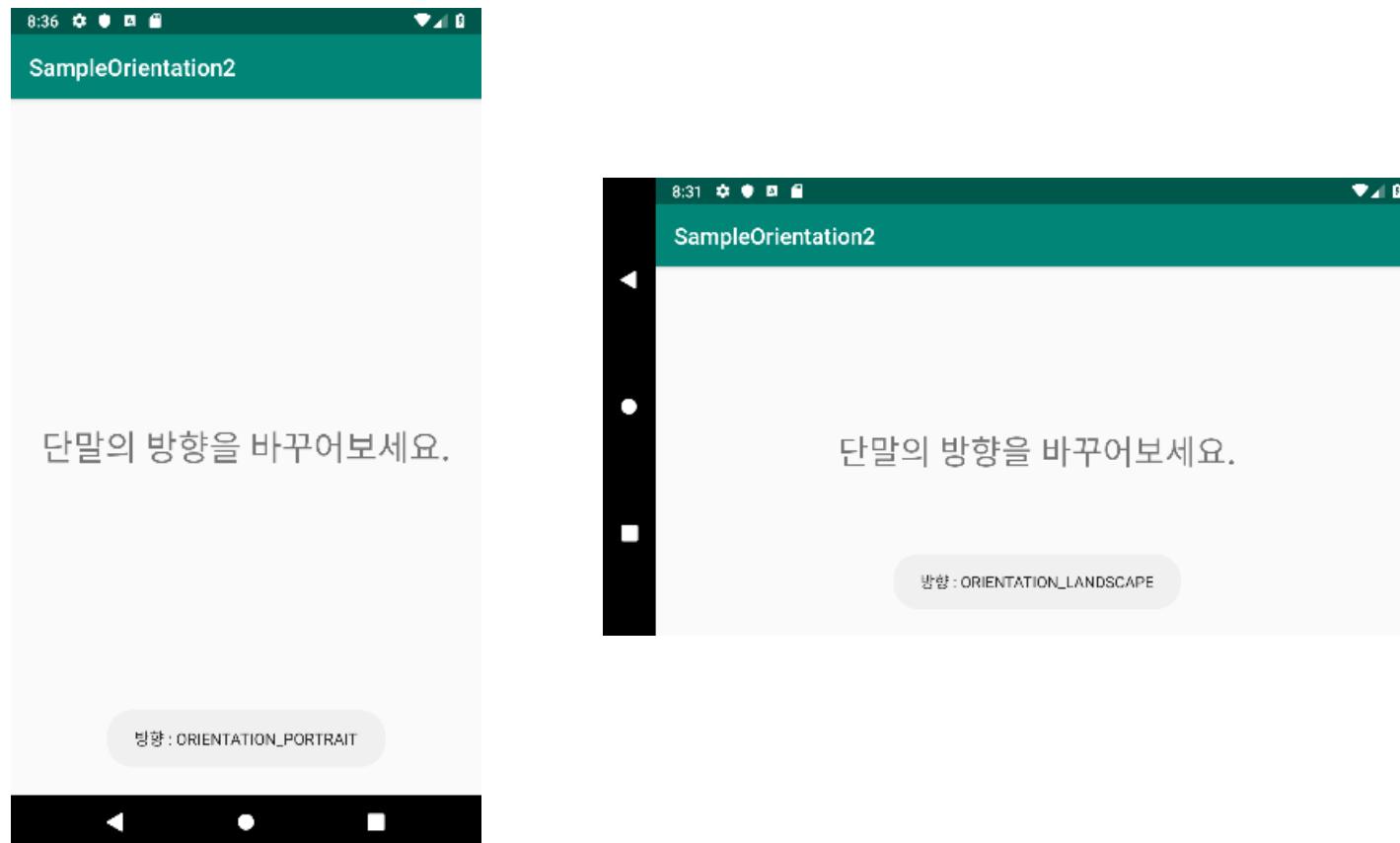
소스 코드에서 방향 전환 이벤트 전달받음

```
public class MainActivity extends AppCompatActivity {  
    ...  
    public void onConfigurationChanged(Configuration newConfig) {  
        super.onConfigurationChanged(newConfig);  
  
        if (newConfig.orientation == Configuration.ORIENTATION_LANDSCAPE) {  
            showToast("방향 : ORIENTATION_LANDSCAPE");  
        } else if (newConfig.orientation == Configuration.ORIENTATION_PORTRAIT) {  
            showToast("Orientation : ORIENTATION_PORTRAIT");  
        }  
    }  
    ...  
}
```



앱 실행 결과

- 단말 방향 바꾸었을 때 값 액티비티는 유지되고 이벤트 전달받음





4.

토스트, 스낵바 그리고 대화상자



토스트와 대화상자

- 토스트

- 간단한 메시지를 잠깐 보여주었다가 없어지는 뷰로 애플리케이션 위에 떠 있는 뷰라 할 수 있음

[Code]

```
Toast.makeText(Context context, String message, int duration)
```

[Code]

```
public void setGravity(int gravity, int xOffset, int yOffset)  
public void setMargin(float horizontalMargin, float verticalMargin)
```



토스트 만들기 예제

토스트 만들기 예제

- 토스트의 색상이나 모양을 직접 구성
- 새로운 레이아웃 정의

메인 액티비티
XML 레이아웃 정의

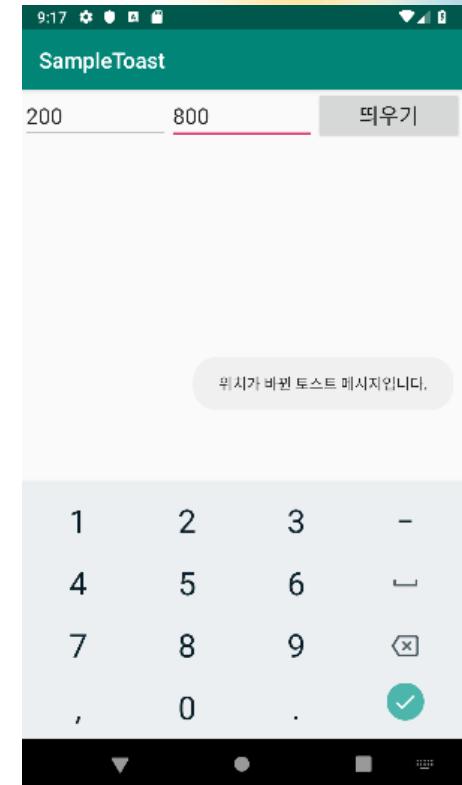
- 메인 액티비티의 레이아웃 정의

메인 액티비티 코드 작성

- 메인 액티비티에서 위치 설정

토스트를 위한
XML 레이아웃 정의

- 토스트의 모양을 XML 레이아웃으로 정의
- 메인 액티비티에서 모양 설정





위치가 바뀐 토스트

```
public void onButton1Clicked(View v) {  
    try {  
        Toast toastView = Toast.makeText(this, "위치가 바뀐 토스트 메시지입니다.",  
                                         Toast.LENGTH_LONG);  
        int xOffset = Integer.parseInt(editText.getText().toString());  
        int yOffset = Integer.parseInt(editText2.getText().toString());  
  
        toastView.setGravity(Gravity.TOP|Gravity.TOP, xOffset, yOffset);  
        toastView.show();  
    } catch (NumberFormatException e) {  
        e.printStackTrace();  
    }  
}
```



토스트 모양 바꾸기 – 메인 액티비티 코드 만들기

참조파일 SampleToast>/app/java/org.techtown.sampletoast/MainActivity.java

중략…

```
public void onButton2Clicked(View v) {
    LayoutInflater inflater = getLayoutInflater(); —→ ① 레이아웃 인플레이터 객체 참조
    View layout = inflater.inflate( —→ ② 토스트를 위한 레이아웃 인플레이션
        R.layout.toastborder,
        (ViewGroup) findViewById(R.id.toast_layout_root));
    TextView text = layout.findViewById(R.id.text);

    Toast toast = new Toast(this); —→ ③ 토스트 객체 생성
    text.setText("모양 바꾼 토스트");
    toast.setGravity(Gravity.CENTER, 0, -100);
    toast.setDuration(Toast.LENGTH_SHORT);
```



토스트 모양 바꾸기 – 토스트의 XML 레이아웃

참조파일 SampleToast>/app/res/layout/toastborder.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:id="@+id/toast_layout_root"  
    android:orientation="horizontal"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:padding="10dp"  
    >  
    <TextView  
        android:id="@+id/text"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:padding="20dp"  
        android:textSize="32sp"  
        android:background="@drawable/toast"  
    />  
</LinearLayout>
```



Shape 객체를 위한 XML 정의

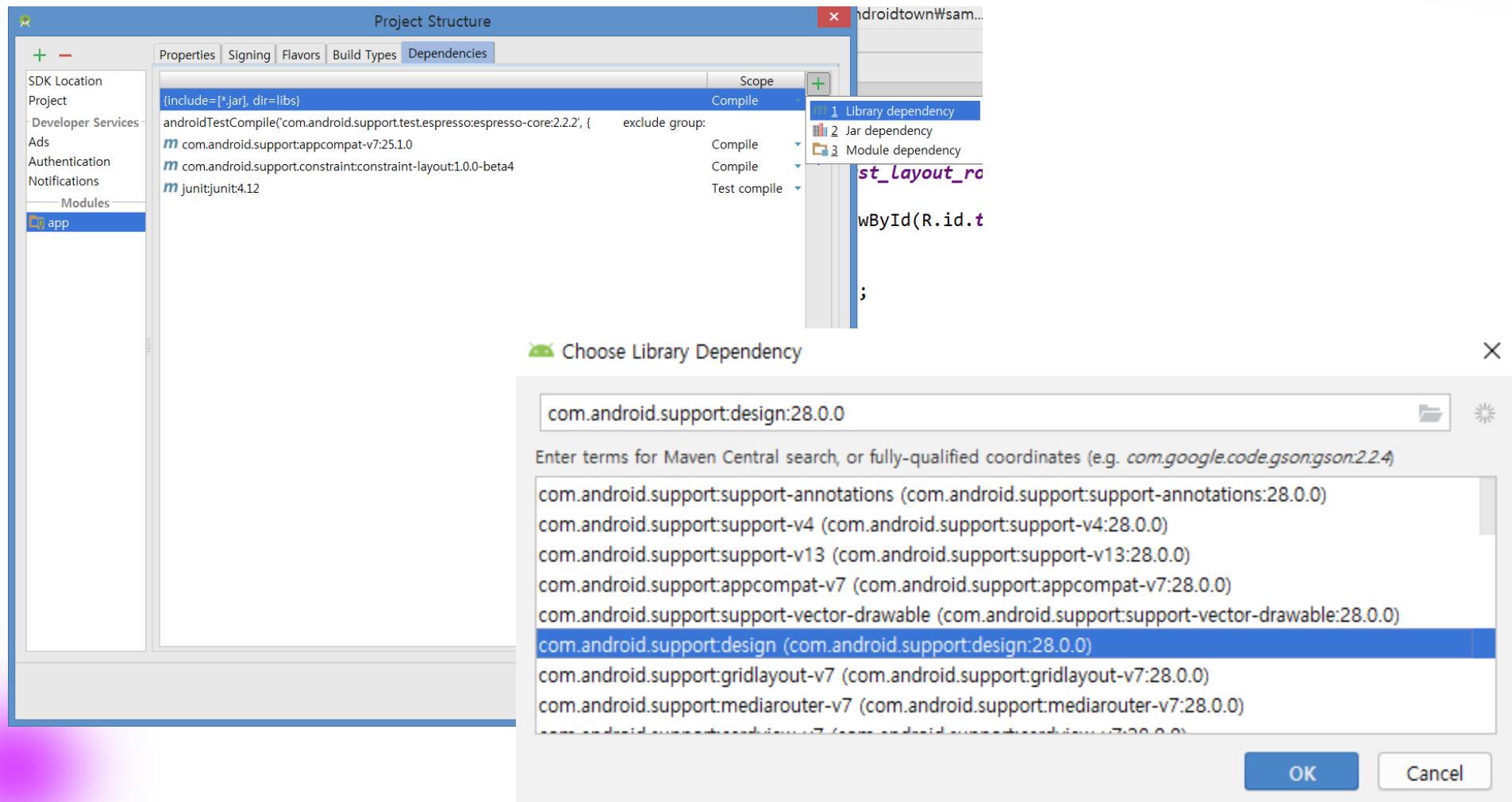
```
<?xml version="1.0" encoding="UTF-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle"
    >
    <stroke
        android:width="4dp"
        android:color="#fffff00"
    />
    <solid
        android:color="#ff883300"
    />
    <padding
        android:left="20dp"
        android:top="20dp"
        android:right="20dp"
        android:bottom="20dp"
    />
    <corners
        android:radius="15dp"
    />
</shape>
```





스낵바 보여주기

- 외부 라이브러리인 design 라이브러리 추가





스낵바 보여주기

- 사용 코드는 토스트 메시지를 띄울 때와 유사함

```
...  
public void onButton3Clicked(View v) {  
    Snackbar.make(v, "스낵바입니다.", Snackbar.LENGTH_LONG).show();  
}  
...
```





알림 대화상자 보여주기 예제

대화상자 만들기 예제

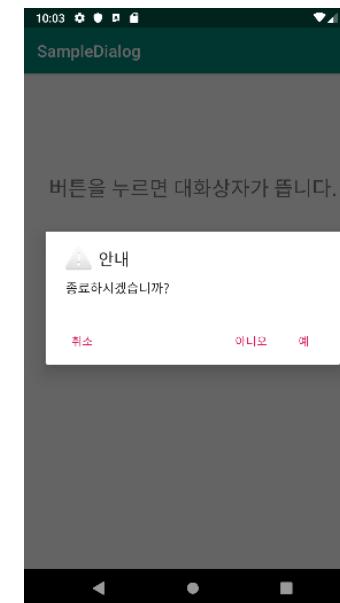
- 대화상자 보여주기
- 이벤트 처리

메인 액티비티
XML 레이아웃 정의

- 메인 액티비티의 레이아웃 정의

메인 액티비티 코드 작성

- 메인 액티비티에서 대화상자 보여주기





메인 액티비티 코드 만들기

```
private void showMessage() {
    AlertDialog.Builder builder = new AlertDialog.Builder(this); —→ ①
    builder.setTitle("안내");
    builder.setMessage("종료하시겠습니까?");
    builder.setIcon(android.R.drawable.ic_dialog_alert);

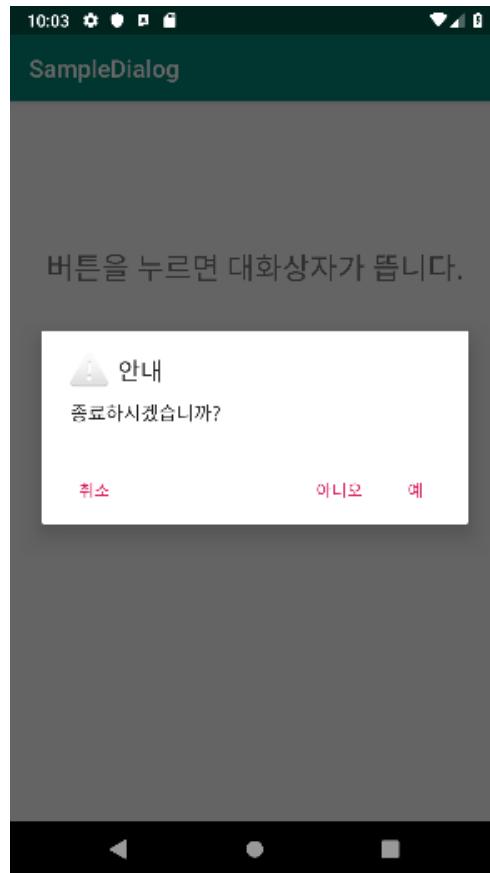
    builder.setPositiveButton("예", new DialogInterface.OnClickListener() { —→ ②
        public void onClick(DialogInterface dialog, int which) {
            String message = "예 버튼이 눌렸습니다. ";
            textView.setText(message);
        }
    });

    builder.setNeutralButton("취소", new DialogInterface.OnClickListener() { —→ ③
        public void onClick(DialogInterface dialog, int which) {
            String message = "취소 버튼이 눌렸습니다. ";
            textView.setText(message);
        }
    });

    builder.setNegativeButton("아니오", new DialogInterface.OnClickListener() { —→ ④
        public void onClick(DialogInterface dialog, int which) {
            String message = "아니오 버튼이 눌렸습니다. ";
            textView.setText(message);
        }
    });
    AlertDialog dialog = builder.create(); —→ ⑤
    dialog.show();
}
```



알림 대화상자 실행 화면





5.

프로그레스바 사용하기



프로그레스바 사용하기

- 여러 가지 화면을 구성하고 그 안에 다양한 위젯을 사용하는데 있어서 대화상자처럼 중간 중간 상태 정보를 보여주는 가장 좋은 방법 중 하나임
- 최대값을 설정하는 `max`, 현재 값을 설정하는 `progress` 속성이 중요함

• 막대 모양

- 작업의 진행 정도를 알려줄 수 있도록 막대 모양으로 표시함
- `style` 속성의 값을 "?android:attr/progressBarStyleHorizontal"로 설정함

• 원 모양

- 작업이 진행 중임을 알려줌
- 원 모양으로 된 프로그레스바가 반복적으로 표시됨



프로그래스바 사용 메소드

[Code]

```
void setProgress (int progress)  
void incrementProgressBy (int diff)
```

- 진행률이 변경되면 progress 속성으로 설정되었던 값을 바꿈

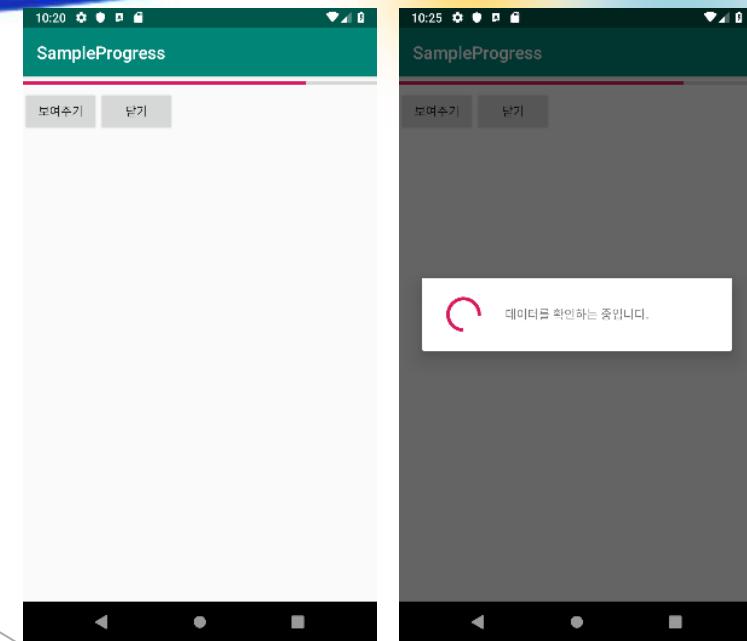
[Code]

```
requestWindowFeature(Window.FEATURE_PROGRESS);
```





프로그레스바 사용하기 예제



프로그레스바 사용하기 예제

-프로그레스바로 진행상태 보여주기

메인 액티비티의
XML 레이아웃 정의

-메인 액티비티 레이아웃 정의

메인 액티비티 코드 작성

-프로그레스바 사용 코드 넣기



레이아웃 만들기

```
<ProgressBar  
    style="?android:attr/progressBarStyleHorizontal"  
    android:id="@+id/progressBar"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:indeterminateOnly="false"  
    android:minHeight="20dp"  
    android:maxHeight="20dp"  
    android:max="100"  
    />
```

1

프로그레스바 정의



메인 액티비티 만들기

```
ProgressBar progressBar = (ProgressBar) findViewById(R.id.progressBar);
progressBar.setIndeterminate(false);
progressBar.setMax(100);
progressBar.setProgress(80);
```



메인 액티비티 만들기 (계속)

```
progressDialog = new ProgressDialog(this);  
progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);  
progressDialog.setMessage("데이터를 확인하는 중입니다.");
```



화면을 여러 개 만들고 화면 간에 전환하는 방법에 대한 이해



- 1 레이아웃 인플레이션 이해하기
- 2 여러 화면 만들고 화면 간 전환하기
- 3 인텐트 살펴보기
- 4 플래그와 부가 데이터 사용하기
- 5 태스크 관리 이해하기
- 6 액티비티의 수명주기와 SharedPreferences 이해하기



XML로 만든 레이아웃은 어떻게 화면에 보여지는 것일까요?

- 레이아웃 인플레이션 이해하기



화면을 더 추가하고 싶어요

- 화면 구성과 화면 간 전환
- 인텐트 살펴보기



다른 화면으로 데이터를 전달할 수 있나요?
태스크라는 것은 뭔가요?

- 액티비티를 위한 플래그와 부가 데이터
- 태스크 관리 이해하기



수명주기에 대해 알고 싶어요

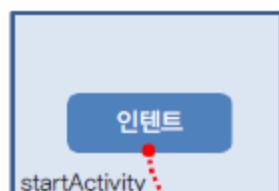
- 액티비티의 수명주기



간단한 데이터를 저장했다가 가져오려면 어떻게 하나요?

- SharedPreferences 사용하기

액티비티 A



액티비티 B





1.

레이아웃 인플레이션 이해하기



XML 레이아웃 파일과 자바 소스 파일의 매칭

- `setContentView` 메소드에서 XML 레이아웃 파일 매칭

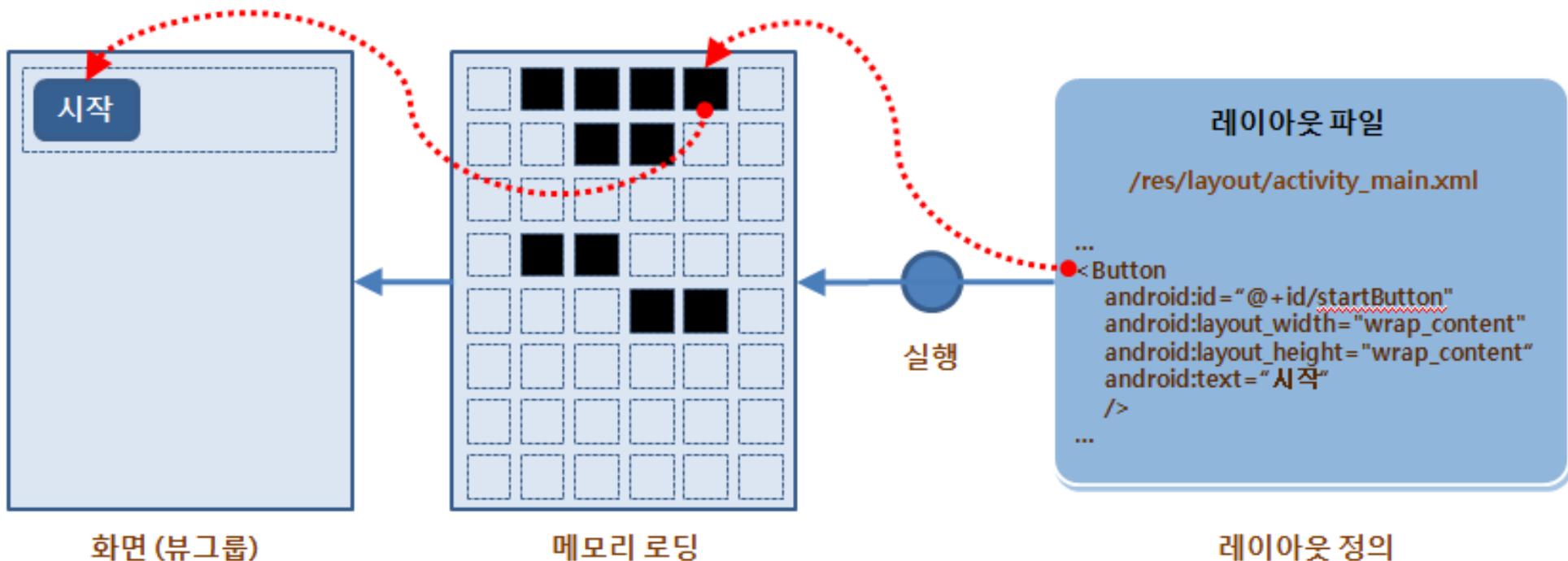


R.layout.레이아웃 파일 이름



인플레이션이란?

- 인플레이션 : XML 레이아웃에 정의된 내용이 메모리에 객체화되는 과정



[“시작” 버튼의 레이아웃 인플레이션 과정]



레이아웃 인플레이션의 이해 – 호출 순서

중략...

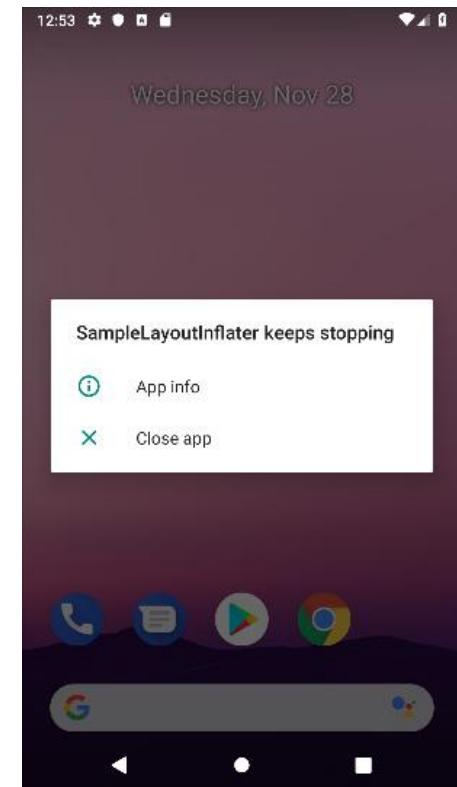
```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        Button button = findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(getApplicationContext(), "버튼이 눌렸어요.",
                    Toast.LENGTH_LONG).show();
            }
        });
    }

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

[`setContentView()` 코드와 `findViewById()` 메소드의 호출 순서를 바꾼 경우]





setContentView() 메소드의 역할

[Reference]

```
public void setContentView (int layoutResID)  
public void setContentView (View view [, ViewGroup.LayoutParams params])
```

- **setContentView() 메소드의 역할**

- 화면에 나타낼 뷰를 지정하는 역할
- XML 레이아웃의 내용을 메모리 상에 객체화하는 역할

[Reference]

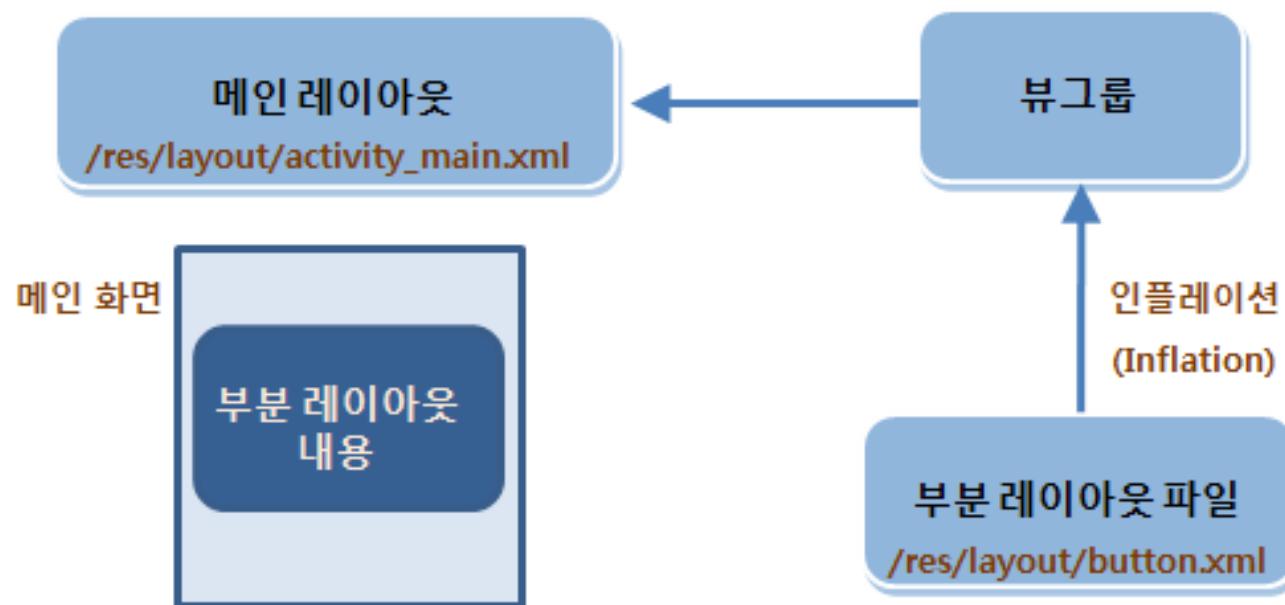
```
getSystemService(Context.LAYOUT_INFLATER_SERVICE)
```

- 전체 화면 중에서 일부분만을 차지하는 화면 구성요소들을 XML 레이아웃에서 로딩하여 보여줄 수 없을까?

- LayoutInflator라는 클래스를 제공하며, 이 클래스는 시스템 서비스로 제공됨



레이아웃 인플레이션의 개념도



[화면의 일부분을 XML 레이아웃 파일의 내용으로 적용하는 과정]



화면 전체와 화면 일부

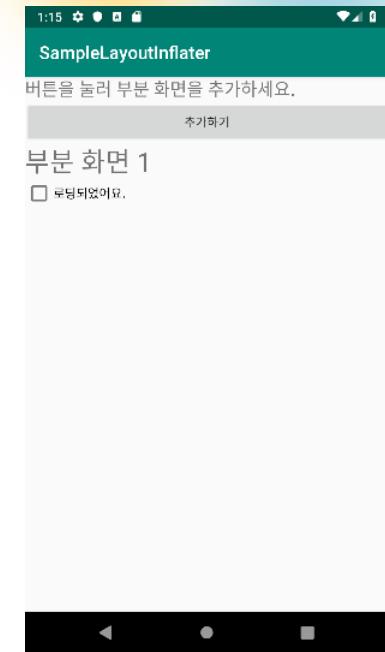
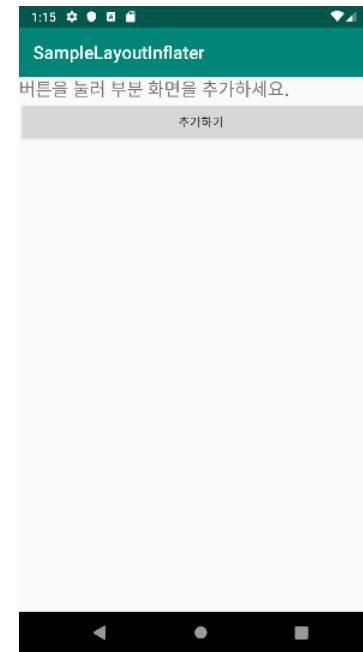
- 안드로이드에서 화면 : 소스와 화면 구성이 분리되어 있음
 - 자바 소스 1개
 - XML 레이아웃 1개
- 화면 전체 : 액티비티 → `setContentView`에서 인플레이션
 - 액티비티를 위한 자바 소스 1개 : `MainActivity.java`
 - 액티비티를 위한 XML 레이아웃 1개 : `activity_main.xml`
- 부분 화면 → 수동으로 인플레이션
 - 부분화면을 위한 자바 소스 1개 또는 뷰 (뷰가 1개의 소스 파일로 분리될 수 있음)
 - 부분화면을 위한 XML 레이아웃 1개 : `singer.xml`



레이아웃 인플레이션 예제

레이아웃 인플레이션 예제

- 화면의 일부로 추가할 뷰의 XML 레이아웃 정의
- 레이아웃 인플레이션 후 자바 코드에서 화면의 일부로 추가



메인 액티비티의
XML 레이아웃

-레이아웃 코드 작성

화면 일부의
XML 레이아웃

-레이아웃 코드 작성

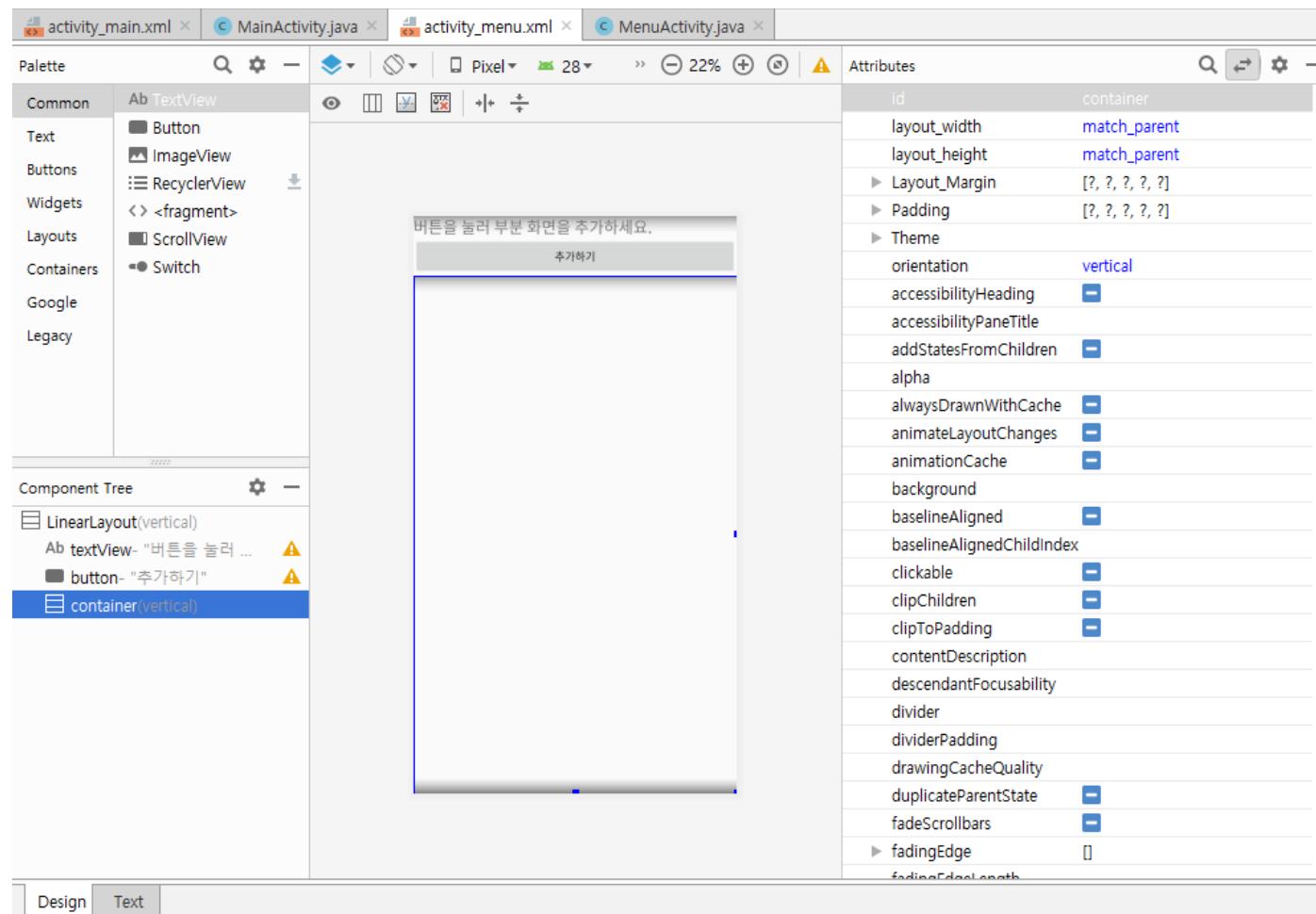
메인 액티비티 코드

-메인 액티비티 코드 작성



메인 액티비티의 레이아웃

- 위쪽에 버튼을 배치하고 아래쪽에 다른 레이아웃이 들어갈 공간을 확보





메인 액티비티의 레이아웃

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

    ...

    <Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="추가하기" />

    <LinearLayout
        android:id="@+id/container"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

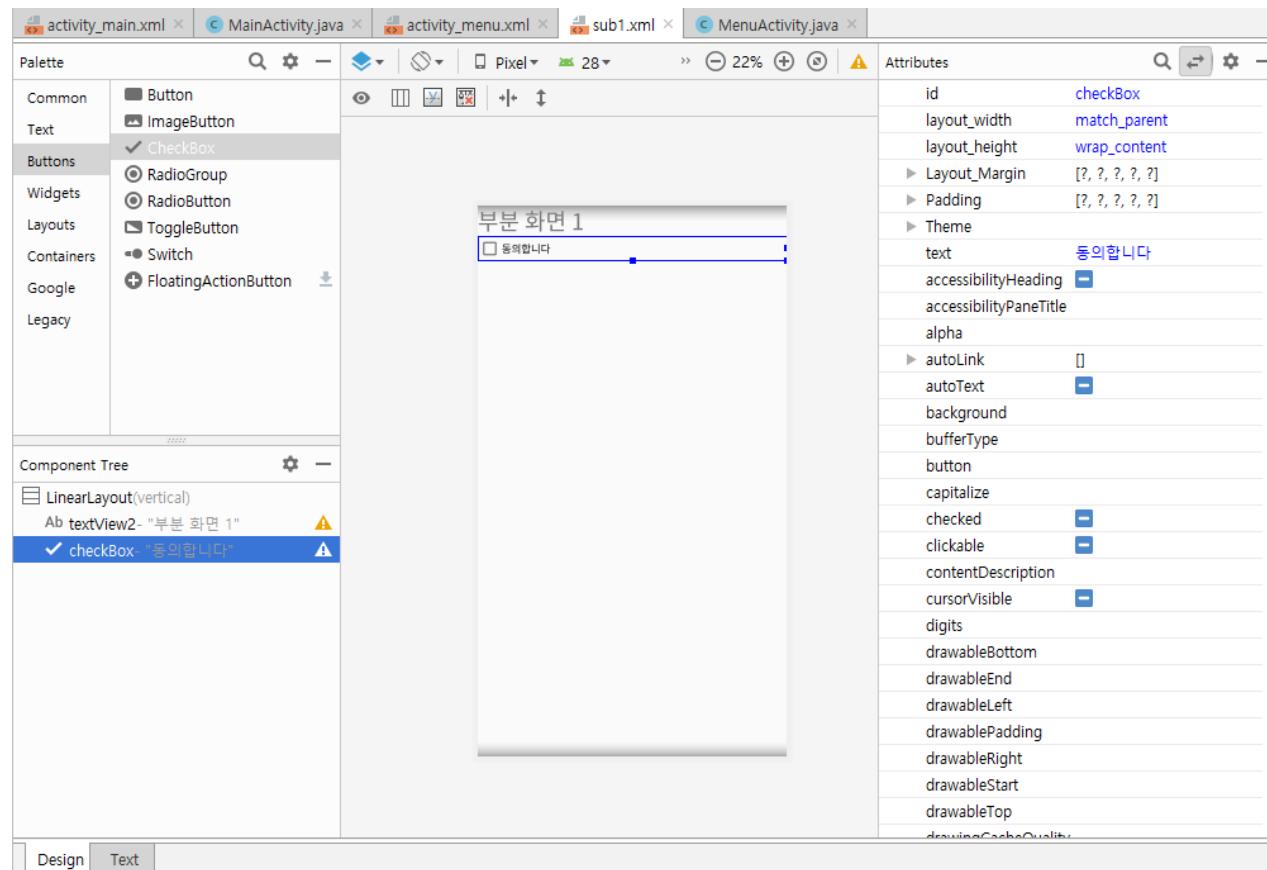
        </LinearLayout>

    </LinearLayout>
```



부분 화면을 위한 레이아웃

- sub1.xml 파일로 만들고 텍스트뷰와 체크박스 위젯 추가





자바 코드 작성

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_menu);

    container = (LinearLayout) findViewById(R.id.container);

    Button button = (Button) findViewById(R.id.button);
    button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            LayoutInflater inflater = (LayoutInflater) getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            inflater.inflate(R.layout.sub1, container, true);

            CheckBox checkBox = (CheckBox) container.findViewById(R.id.checkBox);
            checkBox.setText("로딩되었어요.");
        }
    });
}
```



[Reference]

View inflate (int resource, ViewGroup root)

[Reference]

static LayoutInflator LayoutInflater.from (Context context)

[Reference]

static View inflate (Context context, int resource, ViewGroup root)



2.

여러 화면 만들고 화면 간 전환하기



액티비티



[안드로이드 애플리케이션을 구성하는 네 가지 구성요소]



새로운 프로젝트 만들고 새 화면 추가

- SampleIntent라는 이름으로 새로운 프로젝트 만들고 MenuActivity 추가

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar shows the project structure under "app". It includes "manifests" (AndroidManifest.xml), "java" (org.techtown.sampleintent containing MainActivity and MenuActivity), "res", and "Gradle Scripts".
- Manifest File (AndroidManifest.xml):** The main editor window displays the XML code for the manifest. A tooltip at the bottom of the code area states: "App is not indexable by Google Search; consider adding at least one Activity with an ACTION-VIEW intent filter. See issue explanation for more details. more... (Ctrl+F1)".
- Code Content:**

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="org.techtown.sampleintent">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="SampleIntent"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true">

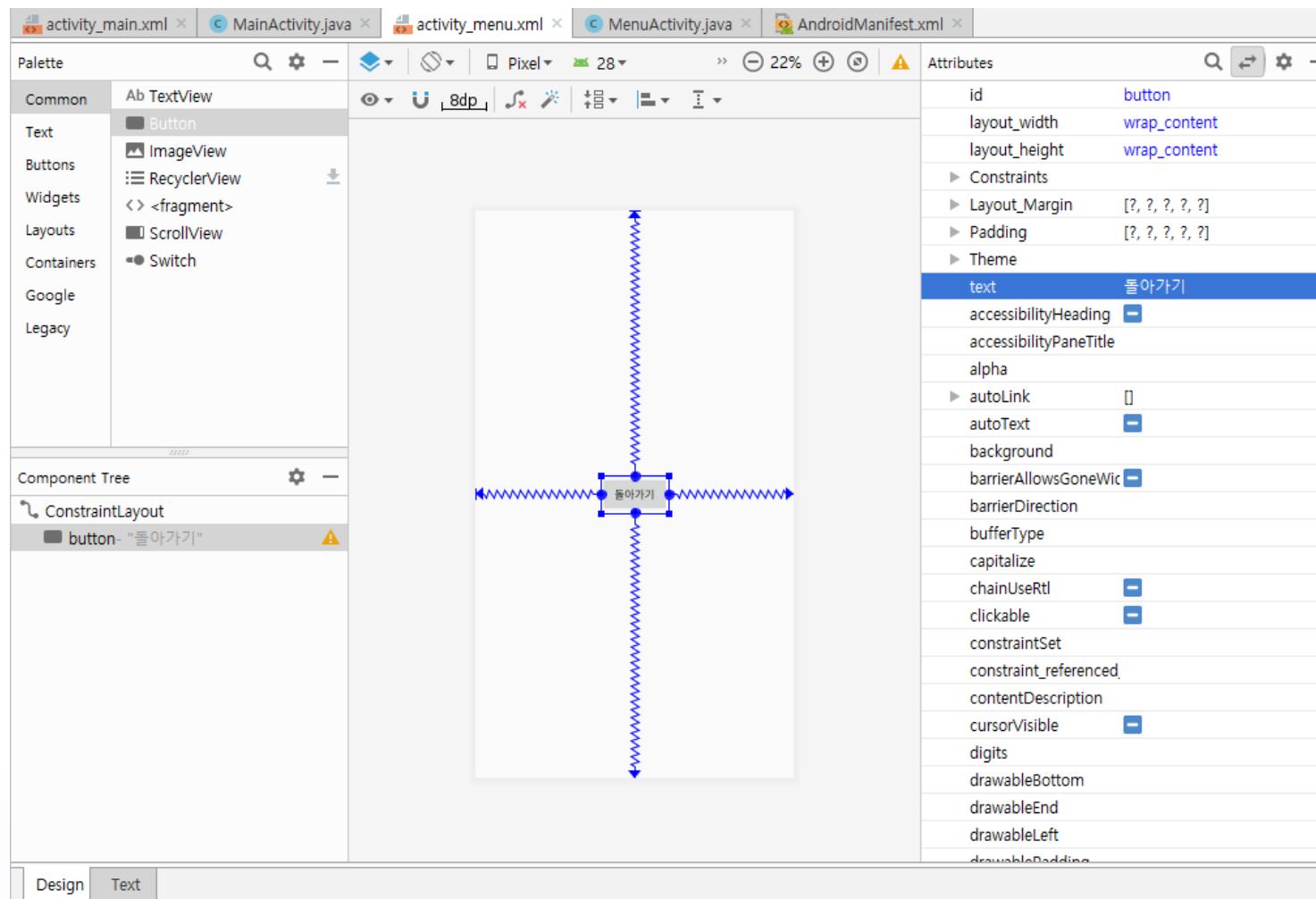
        <activity android:name=".MenuActivity"
            android:label="메뉴 액티비티"
            android:theme="@style/Theme.AppCompat.Dialog">
        </activity>

        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```
- Bottom Status Bar:** A red bar at the bottom indicates: "App is not indexable by Google Search; consider adding at least one Activity with an ACTION-VIEW intent filter. See issue explanation for more details."
- Bottom Navigation:** Includes tabs for "Text" and "Merged Manifest", along with other standard Android Studio navigation icons.



메뉴 화면에 버튼 추가

- `activity_menu.xml` 파일 열고 [돌아가기] 버튼 추가





돌아가기 버튼에 기능 추가

[Reference]

setResult(int resultCode, Intent intent)

참조파일 SampleIntent>/app/java/org.techtown.sampleintent/MenuActivity.java

중략...

```
public class MenuActivity extends AppCompatActivity {

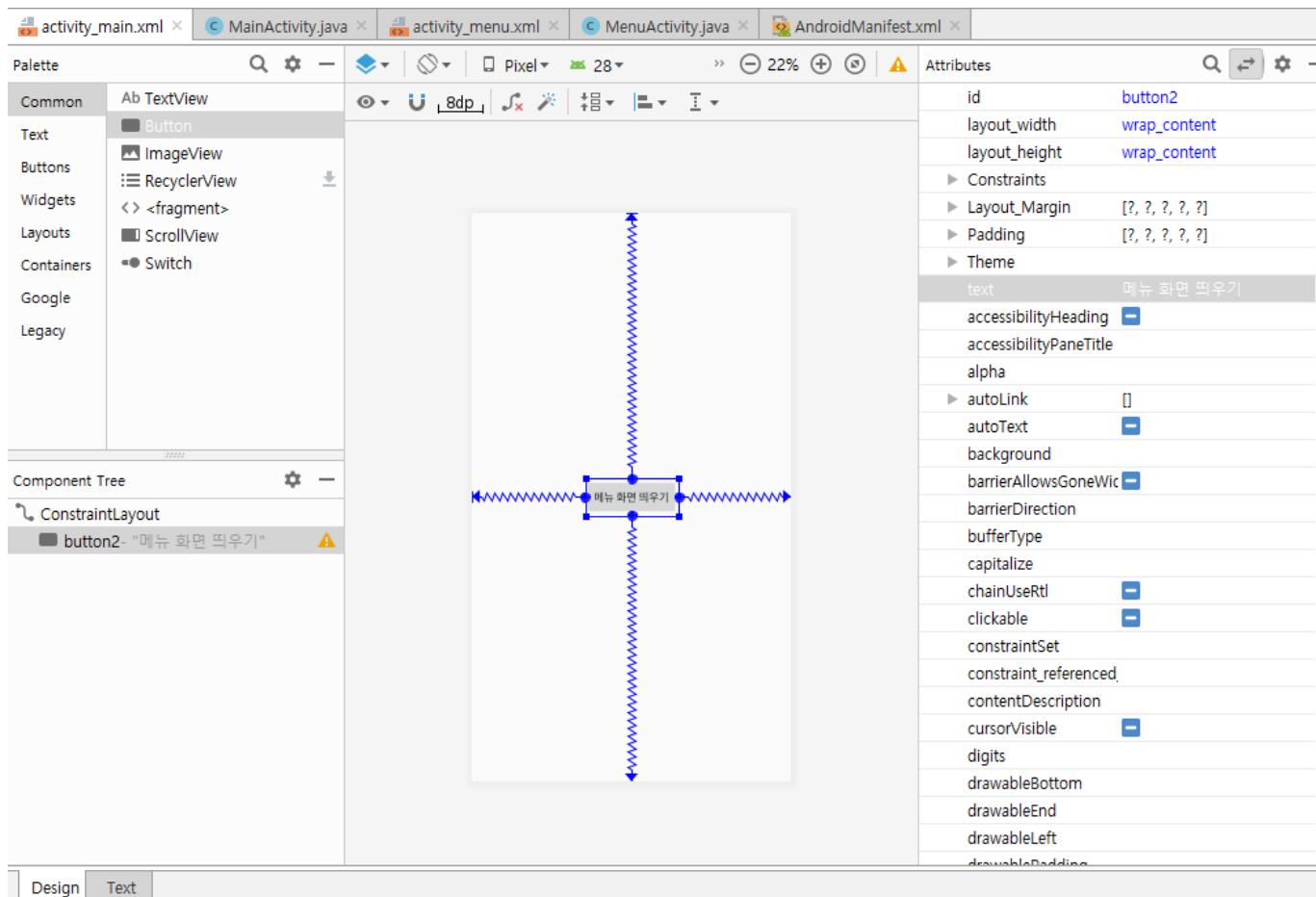
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_menu);

        Button button = findViewById(R.id.button); —→ ❶ 버튼 객체 참조
        button.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                Intent intent = new Intent(); ❷ 인텐트 객체 생성하고 name의 값을 부가 데이터로 넣기
                intent.putExtra("name", "mike");
                setResult(RESULT_OK, intent); —→ ❸ 응답 보내기
                finish(); —→ ❹ 현재 액티비티 없애기
            }
        });
    }
}
```



메인 화면에 버튼 추가

- activity_main.xml 파일 열고 [메뉴화면 띄우기] 버튼 추가





메뉴화면 띄우기 버튼에 기능 추가

[Reference]

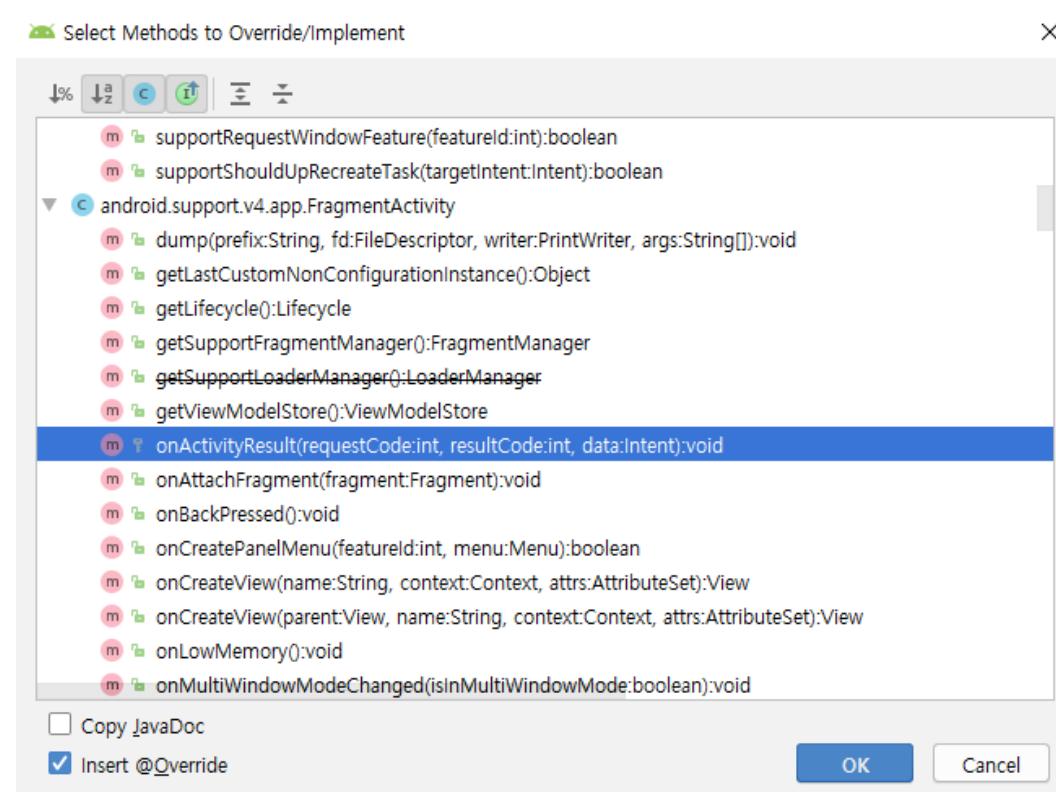
startActivityForResult(Intent intent, int requestCode)

```
public void onButton1Clicked(View v) {  
    Intent intent = new Intent(getApplicationContext(), MenuActivity.class);  
    startActivityForResult(intent, REQUEST_CODE_MENU);  
}
```



응답을 받기 위한 메소드 추가

- onActivityResult 메소드 재정의



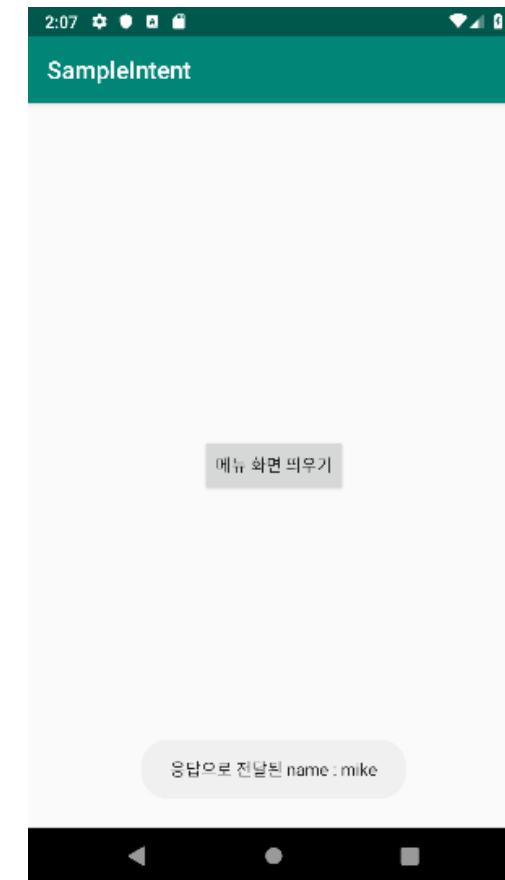
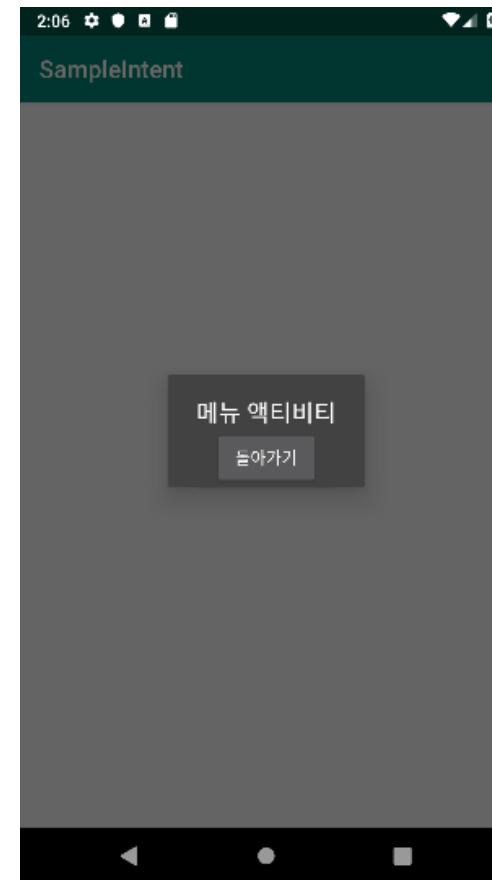
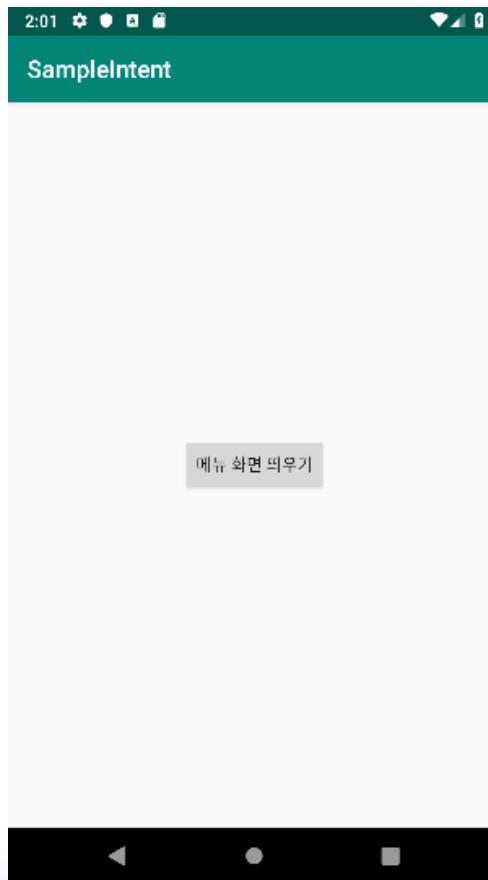


응답받은 결과를 토스트로 보여주기

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    super.onActivityResult(requestCode, resultCode, data);  
  
    if (requestCode == REQUEST_CODE_MENU) {  
        Toast.makeText(getApplicationContext(),  
            "onActivityResult 메소드 호출됨. 요청 코드 : " + requestCode +  
            ", 결과 코드 : " + resultCode, Toast.LENGTH_LONG).show();  
  
    if (resultCode == RESULT_OK) {  
        String name = data.getExtras().getString("name");  
        Toast.makeText(getApplicationContext(), "응답으로 전달된 name : " + name,  
            Toast.LENGTH_LONG).show();  
    }  
}  
}
```

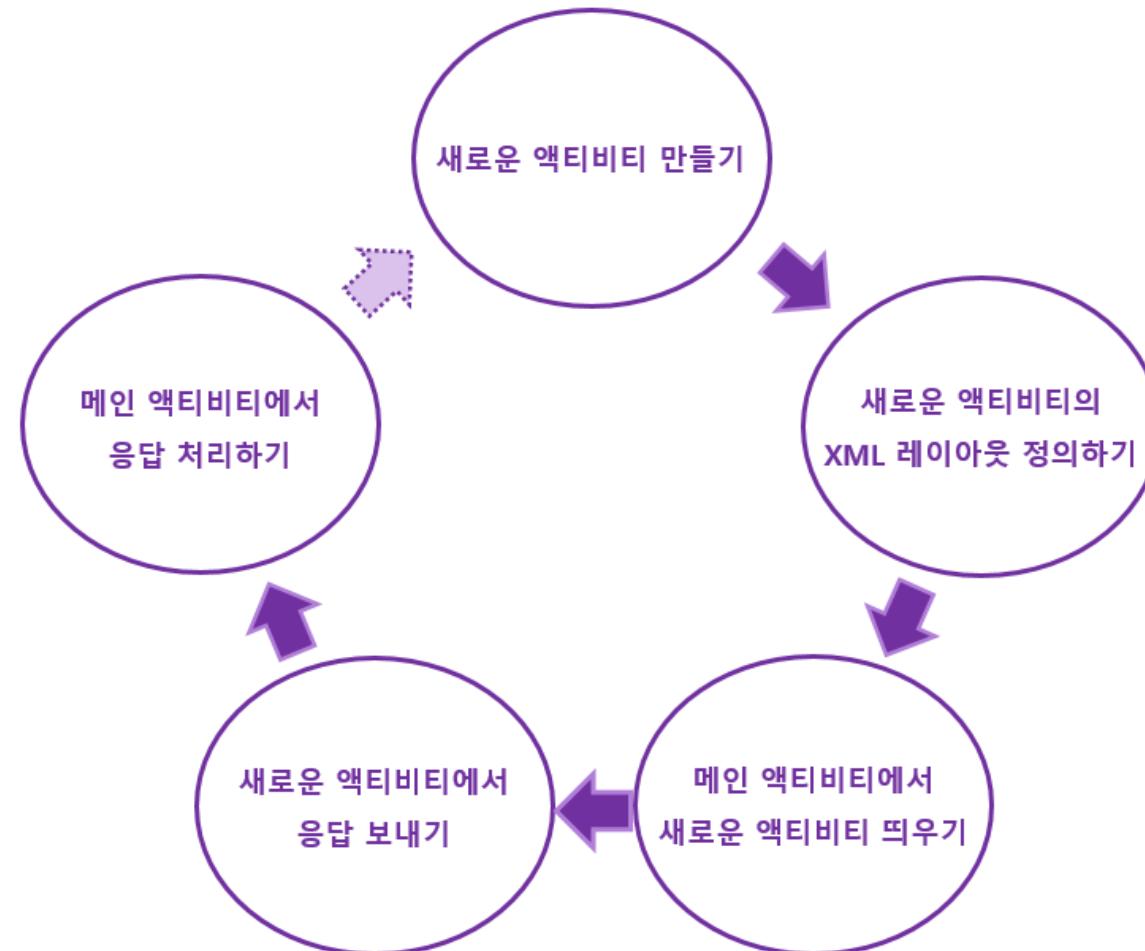


앱 실행 결과



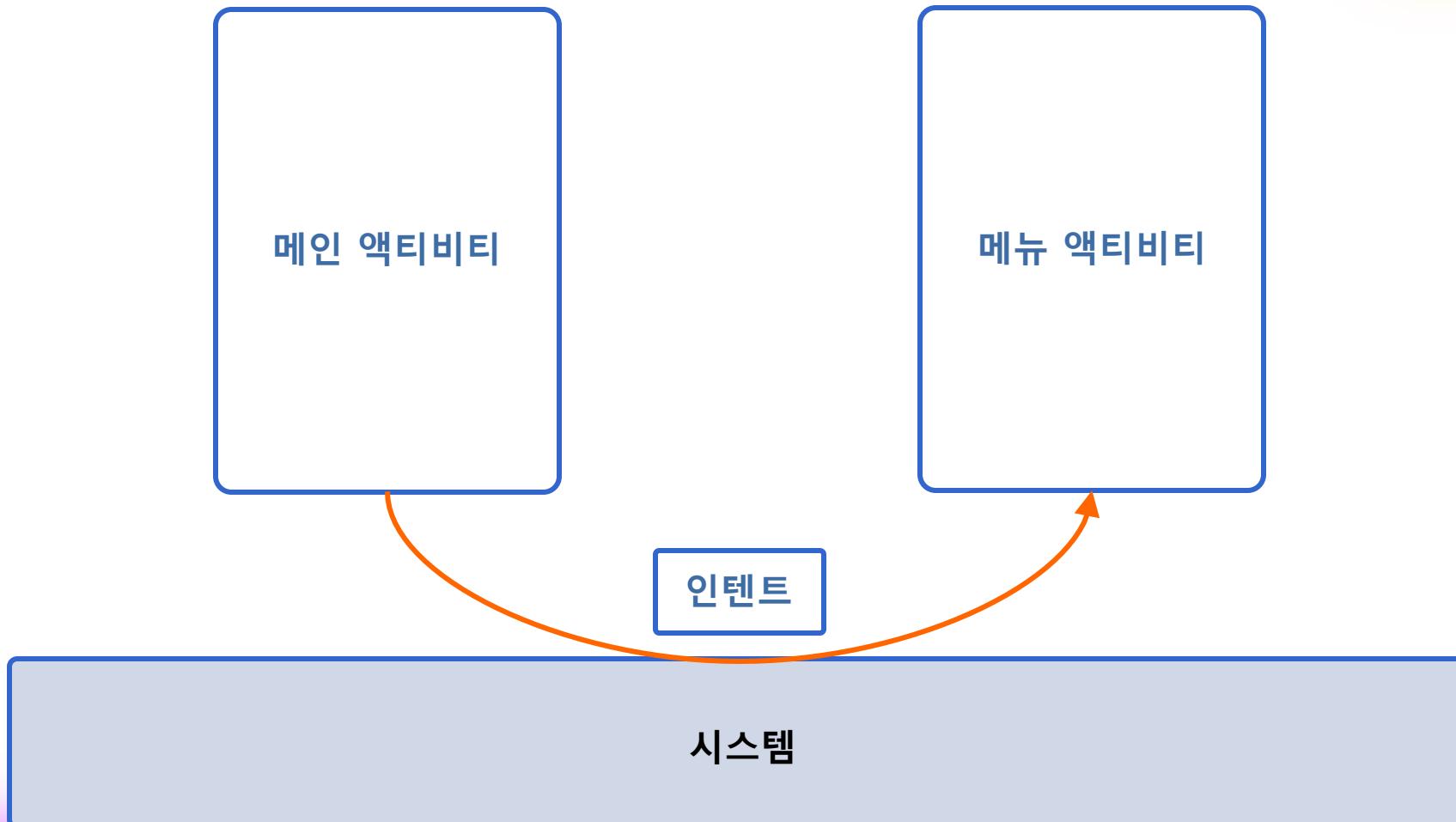


액티비티 구성 과정 정리



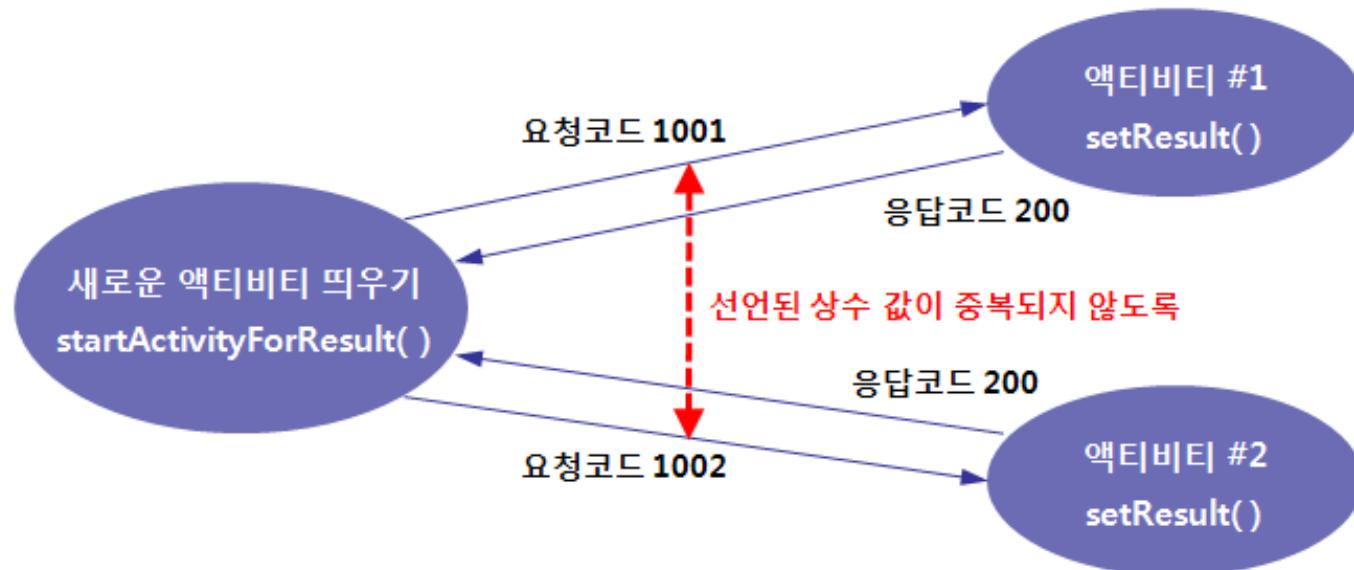


화면을 띄울 때 시스템으로 요청하기





화면 구성과 화면 간 전환 과정



[액티비티에 선언된 상수 값 중복에 주의]

[Reference]

`protected void onActivityResult(int requestCode, int resultCode, Intent Data)`

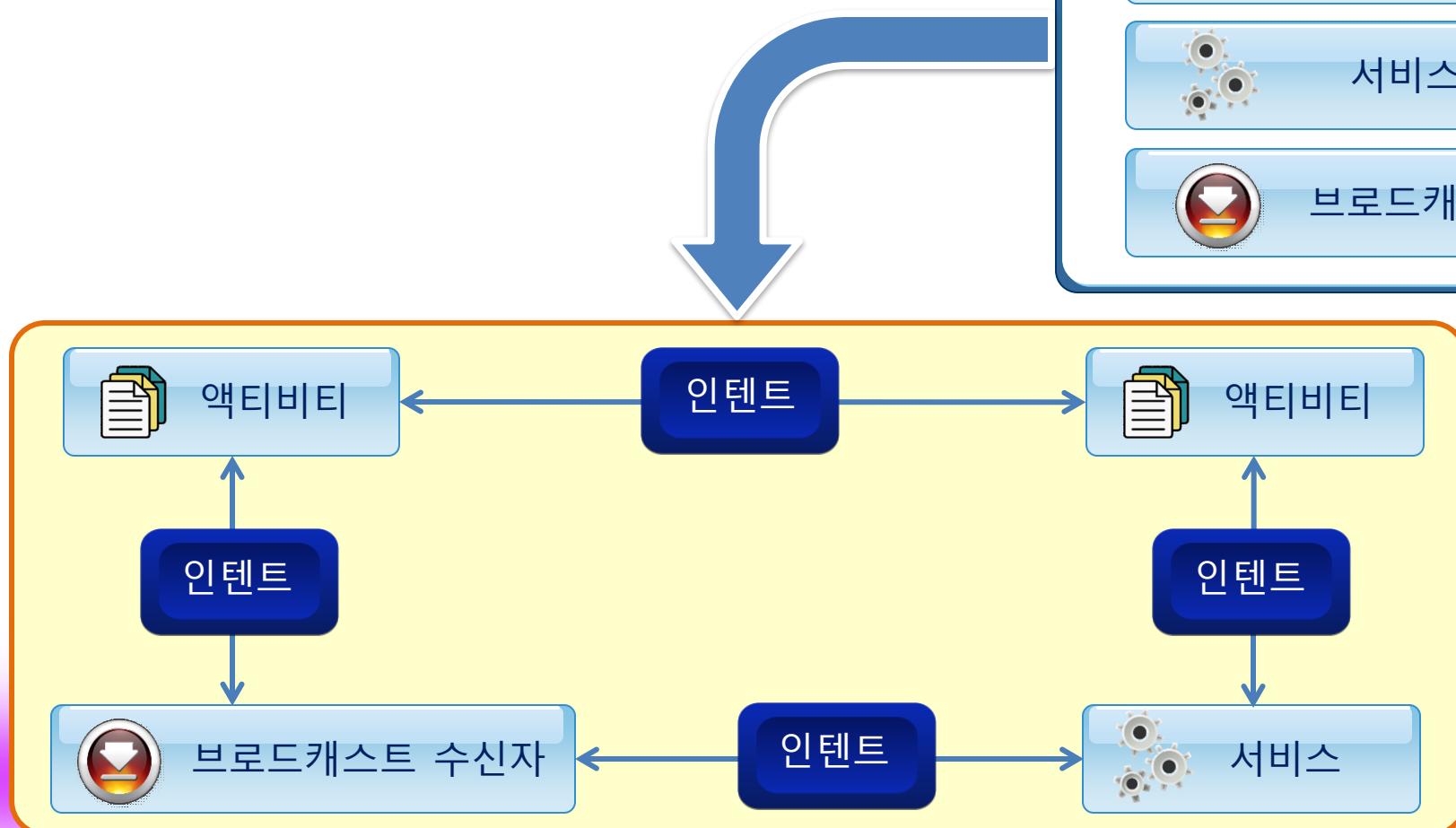


3.

인텐트 살펴보기



인텐트와 데이터 전달





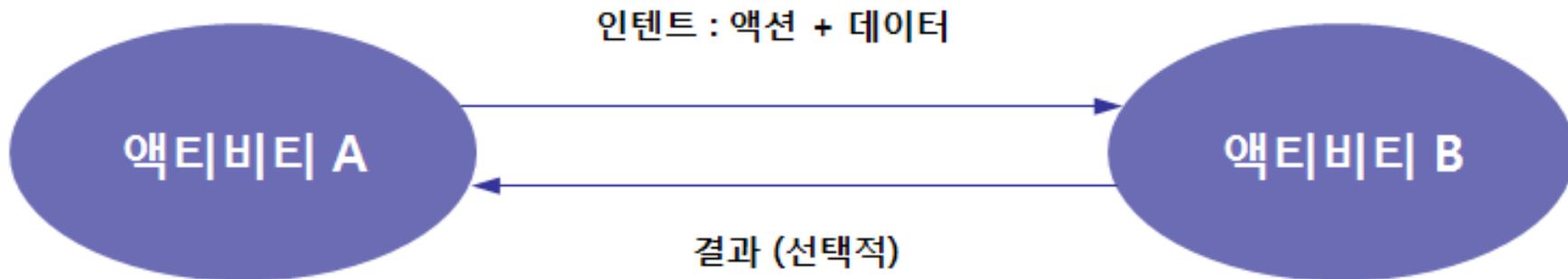
액티비티 간의 인텐트 전달

[Reference]

`startActivity()`

`startService()` 또는 `bindService()`

`broadcastIntent()`



[액티비티 간의 인텐트 전달]



액션과 데이터를 사용하는 대표적인 예

속성	설명
ACTION_DIAL tel:01077881234	주어진 전화번호를 이용해 전화걸기 화면을 보여줌
ACTION_VIEW tel:01077881234	주어진 전화번호를 이용해 전화걸기 화면을 보여줌. URI 값의 유형에 따라 VIEW 액션이 다른 기능을 수행함
ACTION_EDIT content://contacts/people/2	전화번호부 데이터베이스에 있는 정보 중에서 ID 값이 2인 정보를 편집하기 위한 화면을 보여줌
ACTION_VIEW content://contacts/people	전화번호부 데이터베이스의 내용을 보여줌



명시적 인텐트와 암시적 인텐트

[Reference]

`Intent()`

`Intent(Intent o)`

`Intent(String action [,Uri uri])`

`Intent(Context packageContext, Class<?> cls)`

`Intent(String action, Uri uri, Context packageContext, Class<?> cls)`

- 명시적 인텐트(**Explicit Intent**)
 - 인텐트에 클래스 객체나 컴포넌트 이름을 지정하여 호출할 대상을 확실히 알 수 있는 경우
- 암시적 인텐트(**Implicit Intent**)
 - 액션과 데이터를 지정하긴 했지만 호출할 대상이 달라질 수 있는 경우
 - 범주(category), 타입(Type), 컴포넌트(component), 부가 데이터(extras)



인텐트의 대표적 속성

- 범주 (Category)

- 액션이 실행되는 데 필요한 추가적인 정보를 제공

- 타입 (Type)

- 인텐트에 들어가는 데이터의 MIME 타입을 명시적으로 지정

- 컴포넌트 (Component)

- 인텐트에 사용될 컴포넌트 클래스 이름을 명시적으로 지정

- 부가 데이터 (Extra)

- 인텐트는 추가적인 정보를 넣을 수 있도록 번들(Bundle) 객체를 담고 있음
 - 이 객체를 통해 인텐트 안에 더 많은 정보를 넣어 다른 애플리케이션 구성요소에 전달할 수 있음



데이터 전달 예제

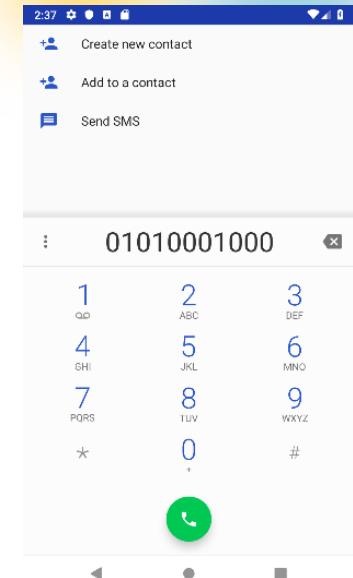
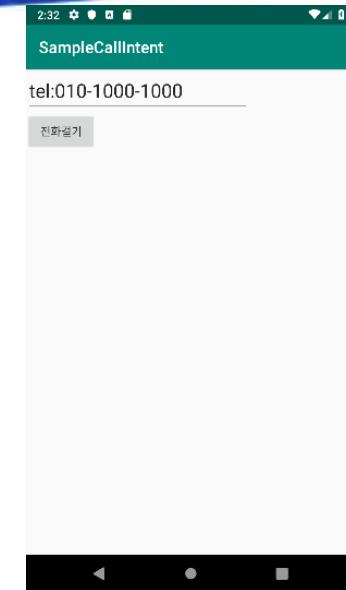
데이터 전달 예제

- 액티비티 간에 인텐트로 데이터 전달
- 전화걸기 기능 이용

XML 레이아웃 정의

메인 액티비티 코드 작성

- 전화걸기 화면을 띄우는 화면 구성
- 전화걸기 화면을 띄울 때 전화번호 전달





XML 레이아웃 만들기

참조파일 SampleCallIntent>/app/res/layout/activity_main.xml

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

```
<EditText
    android:id="@+id/editText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="tel:010-1000-1000"
    android:textSize="24sp"
    />
```

① 전화번호를 입력할 입력상자 정의

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="전화걸기"
    />
```

② 전화걸기 버튼 정의

```
</LinearLayout>
```



메인 액티비티 코드 만들기

참조파일 SampleCallIntent>/app/java/org.techtown.samplecallintent/MainActivity.java

중략...

```
public class MainActivity extends AppCompatActivity {
    EditText editText;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main); —→ ❶ 뷰 객체 참조

        editText = findViewById(R.id.editText);

        Button button = findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String data = editText.getText().toString(); —→ ❷ 입력상자에 입력된 전화번호 확인

                Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(data)); —→ ❸ 전화걸기 화
                startActivity(intent); —→ ❹ 액티비티 띄우기
            }
        });
    }
}
```

면을 보여줄
인텐트 객체
생성

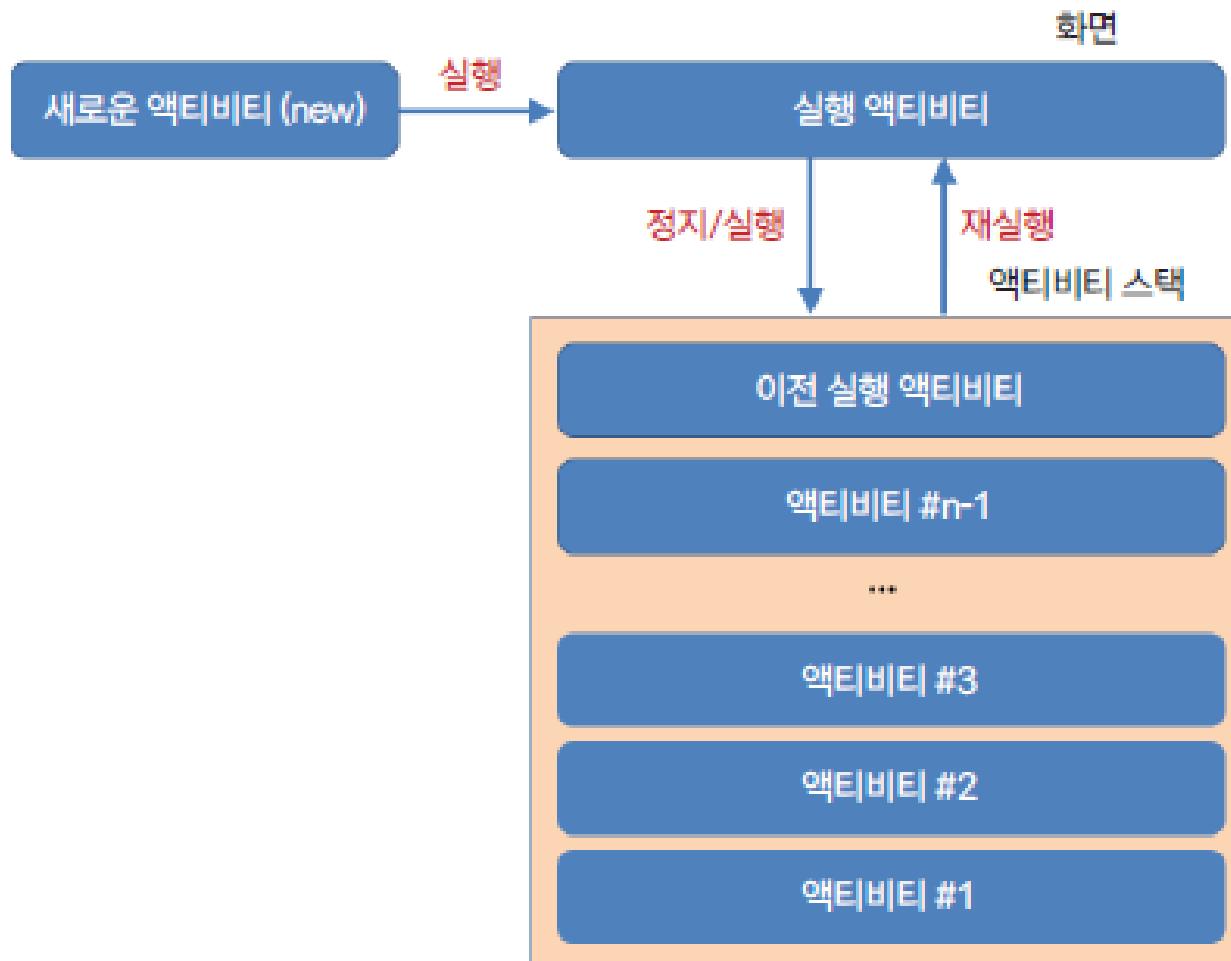


4.

플래그와 부가 데이터 사용하기



액티비티 스택





액티비티 스택과 플래그 사용

[Reference]

`FLAG_ACTIVITY_SINGLE_TOP`
`FLAG_ACTIVITY_NO_HISTORY`
`FLAG_ACTIVITY_CLEAR_TOP`

- 새로운 액티비티를 실행할 때마다 메모리에 새로운 객체를 만들고 이전 화면 위에 쌓는 방식은 비효율적일 수 있음
- 동일한 화면이 이미 만들어져 있는 경우에는 그 화면을 그대로 보여주고 싶다면 플래그를 사용하면 됨

NO_FLAG



FLAG_ACTIVITY_SINGLE_TOP



[`FLAG_ACTIVITY_SINGLE_TOP` 플래그를 사용한 경우]



액티비티 플래그 사용 예

```
Intent intent = new Intent(getApplicationContext(), AnotherActivity.class );
intent.putExtra("startCount", String.valueOf(startCount));
intent.setFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP );
startActivityForResult(intent, REQUEST_CODE_ANOTHER );
```

1 인텐트 객체 생성

2 부가 데이터 넣기

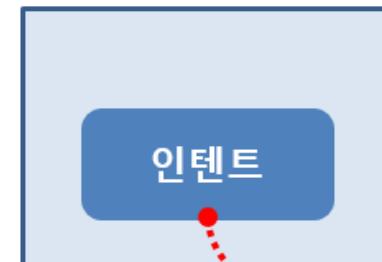
3 인텐트 플래그 설정

4 인텐트 띄우기

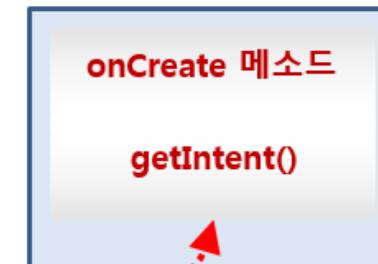


액티비티에서 인텐트를 전달받는 두 가지 경우

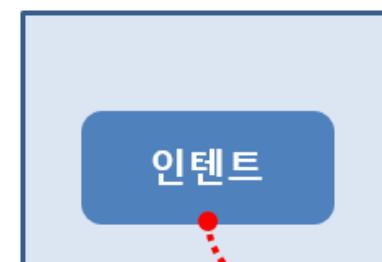
부모 액티비티



액티비티가 새로 만들어지는 경우



부모 액티비티



액티비티 재사용하는 경우





다른 액티비티 플래그의 사용

NO_FLAG



FLAG_ACTIVITY_NO_HISTORY



[`FLAG_ACTIVITY_NO_HISTORY` 플래그를 사용한 경우]

NO_FLAG



FLAG_ACTIVITY_CLEAR_TOP



[`FLAG_ACTIVITY_CLEAR_TOP` 플래그를 사용한 경우]



부가 데이터 전달하기

[Reference]

```
Intent putExtra(String name, String value)
```

```
Intent putExtra(String name, int value)
```

```
Intent putExtra(String name, boolean value)
```

```
String getStringExtra(String name)
```

```
int getIntExtra(String name, int defaultValue)
```

```
boolean getBooleanExtra(String name, boolean defaultValue)
```

[Reference]

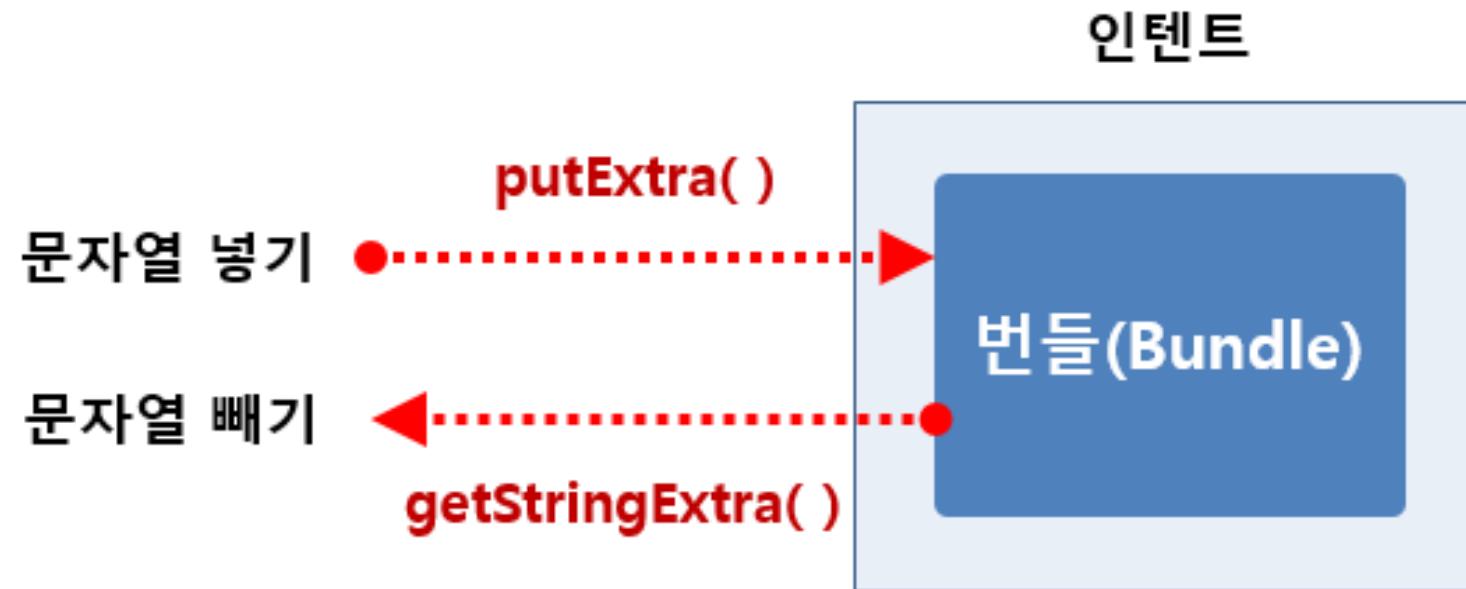
```
public abstract int describeContents()
```

```
public abstract void writeToParcel(Parcel dest, int flags)
```

- 화면과 화면 간에 데이터를 전달하고 싶다면 인텐트의 부가 데이터(Extra)로 넣어 전달하는 방법을 사용함
- 인텐트는 애플리케이션 구성 요소 간에 데이터를 전달하는 방법을 제공하는 것이므로 화면과 화면 간 뿐만 아니라 화면과 서비스 간, 또는 브로드캐스트 수신자와 화면 간 등등 애플리케이션 구성 요소 간에 부가 데이터로 넣어 데이터를 전달할 수 있음

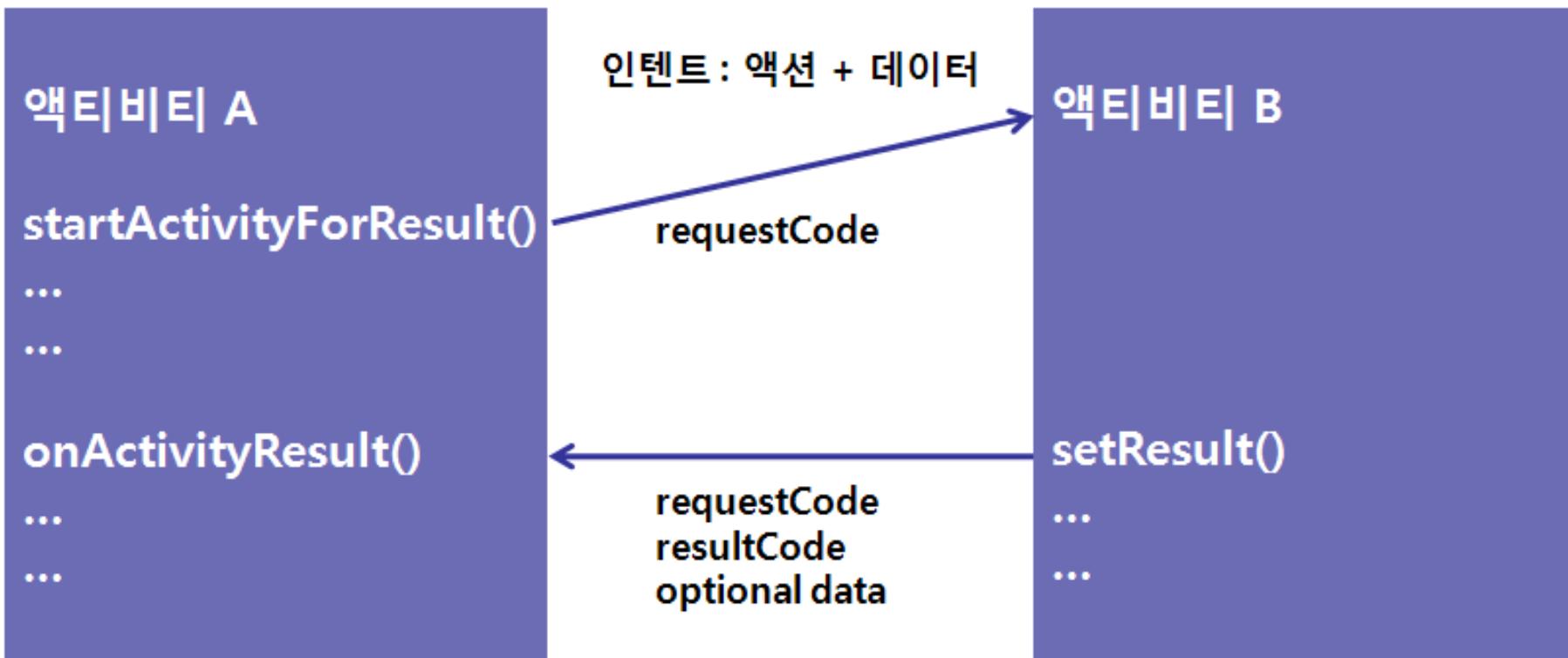


번들 객체





액티비티 간의 데이터 전달 방법 정리





Parcelable 예제

Parcelable 예제

- Parcelable 객체 만들기
- 또 다른 액티비티에 Parcelable 객체 전달하기

SimpleData 객체 정의

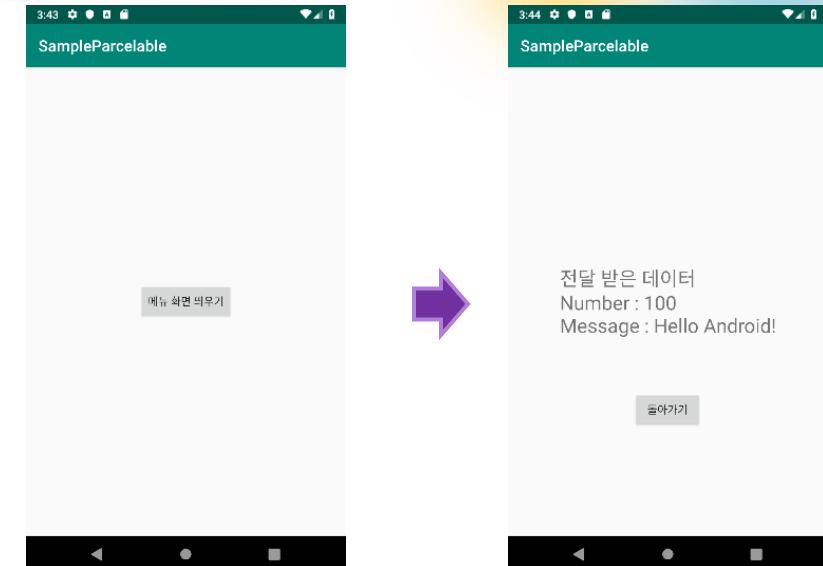
메인 액티비티 코드 작성

- Parcelable 인터페이스를 구현하는 객체 정의
- 또 다른 액티비티를 띄워줄 때 부가데이터 전달

또 다른 액티비티 코드 작성

- 전달된 객체를 이용해 데이터 표시

4. 플래그와 부가 데이터 사용하기





SimpleData 클래스 정의

```
public class SimpleData implements Parcelable {  
    int number;  
    String message;  
    public SimpleData(int num, String msg) {  
        number = num;  
        message = msg;  
    }  
  
    public SimpleData(Parcel src) {  
        number = src.readInt();  
        message = src.readString();  
    }  
  
    public static final Parcelable.Creator<CREATOR> CREATOR = new Parcelable.Creator<>() {  
        public SimpleData createFromParcel(Parcel in) {  
            return new SimpleData (in);  
        }  
        public SimpleData[] newArray(int size) {  
            return new SimpleData[size];  
        }  
    };
```

Continued..



SimpleData 클래스 정의 (계속)

```
public int describeContents() {  
    return 0;  
}  
  
public void writeToParcel(Parcel dest, int flags) {  
    dest.writeInt(number);  
    dest.writeString(message);  
}  
  
public int getNumber() {  
    return number;  
}  
  
public void setNumber(int number) {  
    this.number = number;  
}  
  
public String getMessage() {  
    return message;  
}  
  
public void setMessage(String message) {  
    this.message = message;  
}  
}
```



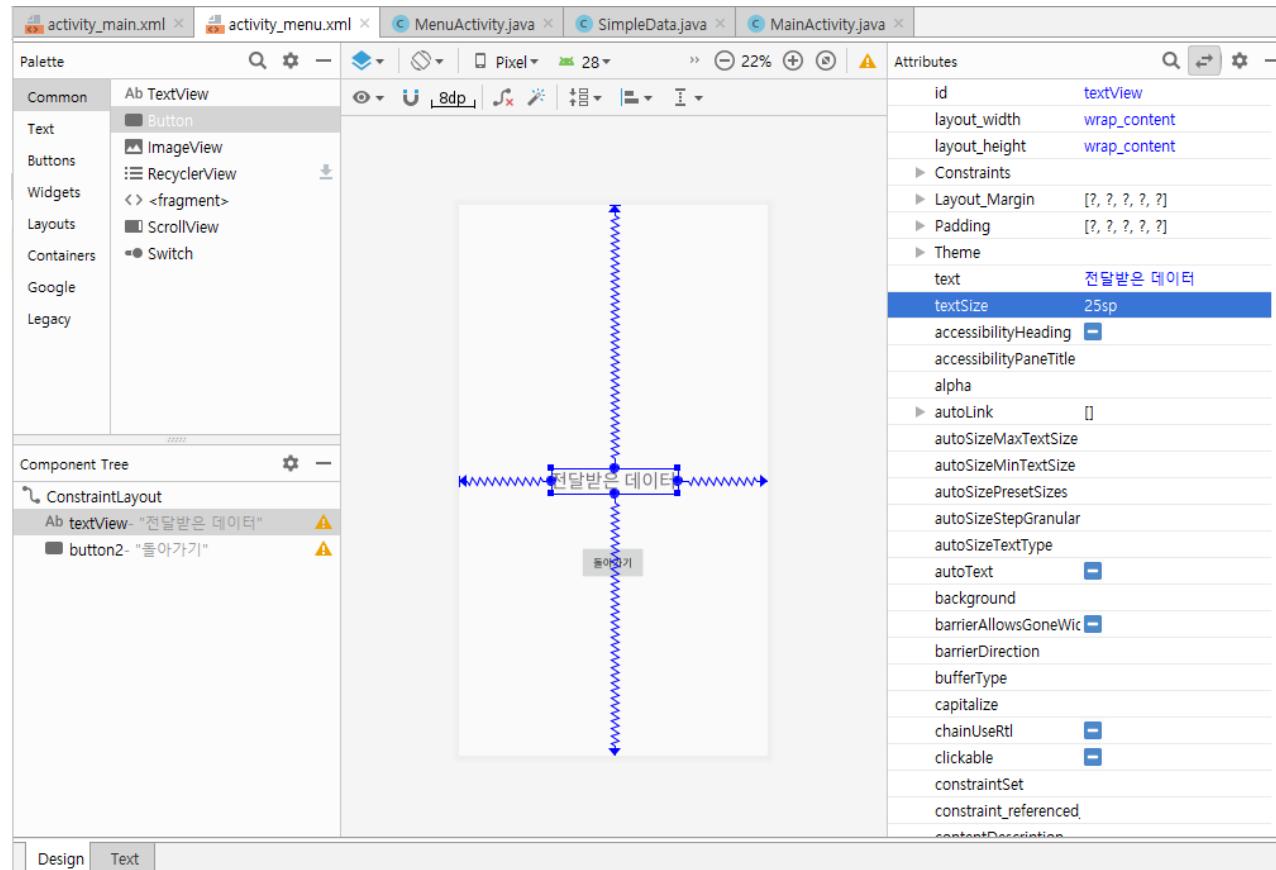
메인 액티비티 코드 만들기

```
...  
public static final String KEY_SIMPLE_DATA = "data";  
...  
  
public void onButton1Clicked(View v) {  
    Intent intent = new Intent(getApplicationContext(), AnotherActivity.class);  
    SimpleData data = new SimpleData(100, "Hello Android!");  
    intent.putExtra(KEY_SIMPLE_DATA, data );  
    startActivityForResult(intent);  
}
```

1 SimpleData 객체 생성
2 인텐트에 부가 데이터로 넣기



메뉴 액티비티의 화면 레이아웃





메뉴 액티비티 코드 만들기

```
...
Intent intent = getIntent();
processIntent(intent);
}

private void processIntent(Intent intent) {
    if (intent != null) {
        Bundle bundle = intent.getExtras();
        SimpleData data = (SimpleData) bundle.getParcelable(KEY_SIMPLE_DATA);
        textView.setText("전달 받은 데이터  
Number : " + data.getNumber()
                        + "  
Message : " + data.getMessage());
    }
}
```



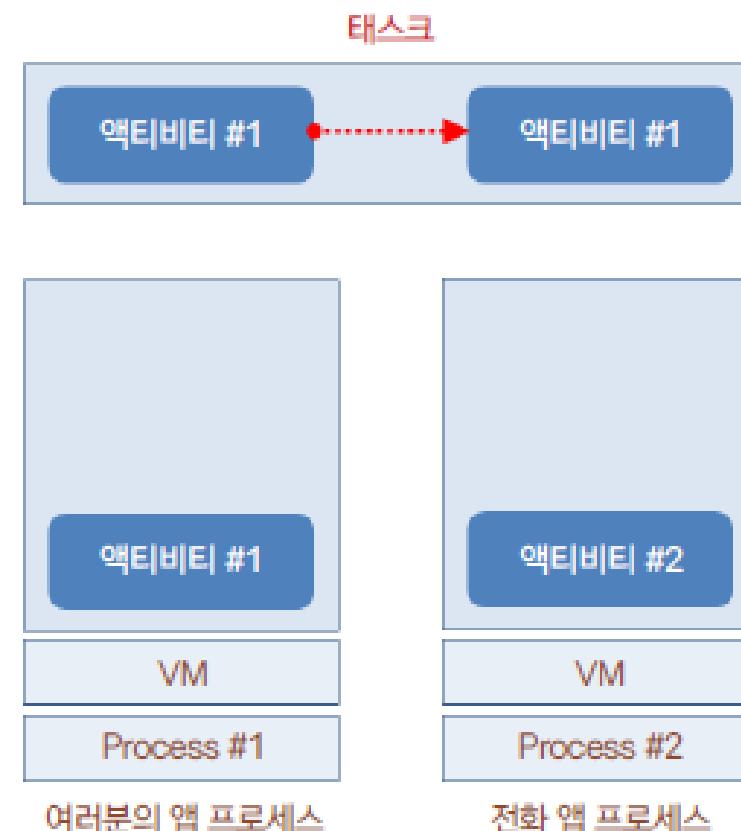
5.

태스크 관리 이해하기



태스크의 역할

- 태스크는 프로세스와 관계없이 화면의 흐름 관리



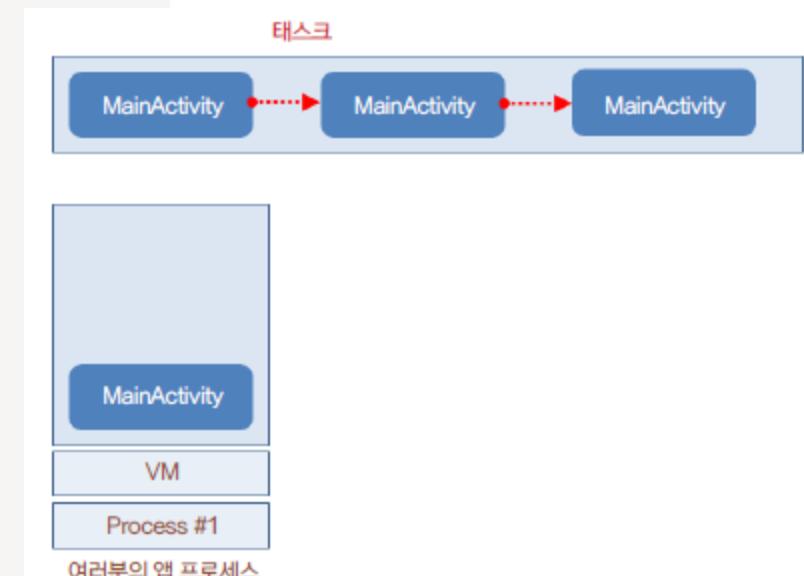


태스크의 역할

- 메인 액티비티를 여러 번 띄울 때의 태스크

참조파일 SampleTask>/app/src/org.techtown.sampletask/MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        Button button = findViewById(R.id.button);  
        button.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                Intent intent = new Intent(getApplicationContext(), MainActivity.class);  
                startActivity(intent);  
            }  
        });  
    }  
}
```



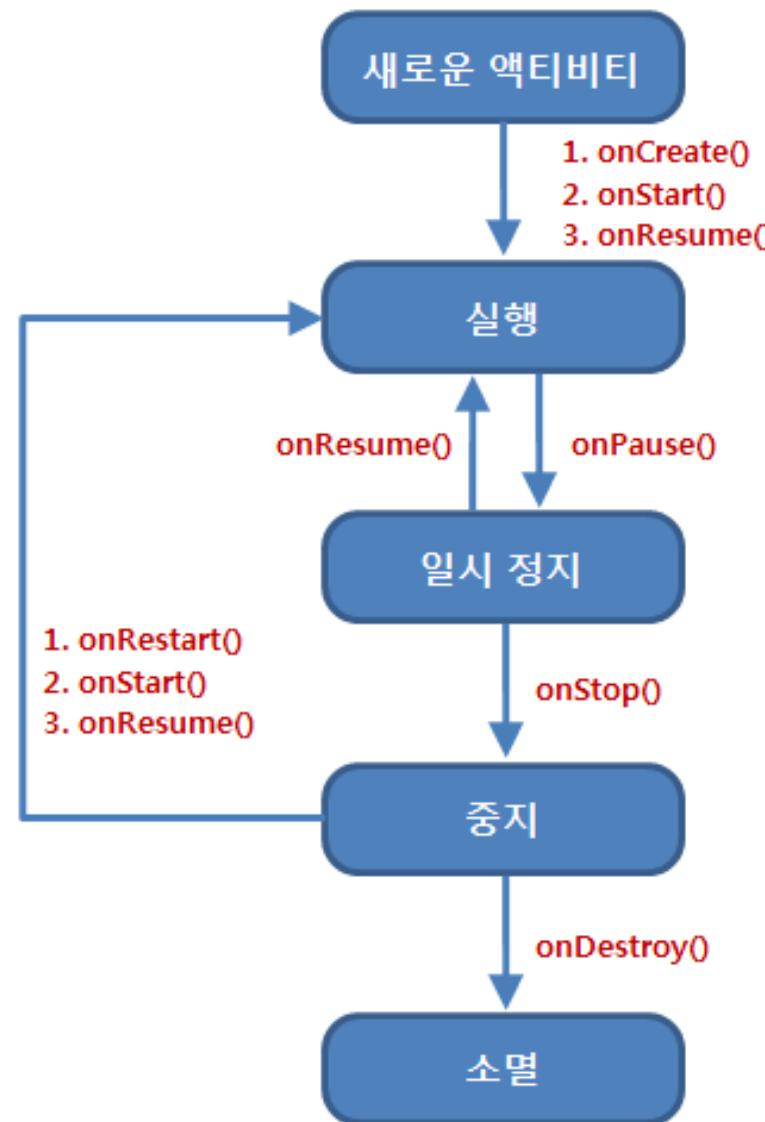


6.

액티비티의 수명주기와 SharedPreferences 이해하기



수명주기에 따른 상태 변화





액티비티의 상태 메소드

상태 메소드	설명
onCreate()	<ul style="list-style-type: none">- 액티비티가 처음에 만들어졌을 때 호출됨- 화면에 보이는 뷰들의 일반적인 상태를 설정하는 부분- 이전 상태가 저장되어 있는 경우에는 번들 객체를 참조하여 이전 상태 복원 가능- 이 메소드 다음에는 항상 onStart() 메소드가 호출됨
onStart()	<ul style="list-style-type: none">- 액티비티가 화면에 보이기 바로 전에 호출됨- 액티비티가 화면 상에 보이면 이 메소드 다음에 onResume() 메소드가 호출됨- 액티비티가 화면에서 가려지게 되면 이 메소드 다음에 onStop() 메소드가 호출됨
onResume()	<ul style="list-style-type: none">- 액티비티가 사용자와 상호작용하기 바로 전에 호출됨
onRestart()	<ul style="list-style-type: none">- 액티비티가 중지된 이후에 호출되는 메소드로 다시 시작되기 바로 전에 호출됨- 이 메소드 다음에는 항상 onStart() 메소드가 호출됨
onPause()	<ul style="list-style-type: none">- 또 다른 액티비티를 시작하려고 할 때 호출됨- 저장되지 않은 데이터를 저장소에 저장하거나 애니메이션 중인 작업을 중지하는 등의 기능을 수행하는 메소드임- 이 메소드가 리턴하기 전에는 다음 액티비티가 시작될 수 없으므로 이 작업은 매우 빨리 수행된 후 리턴되어야 함- 액티비티가 이 상태에 들어가면 시스템은 액티비티를 강제 종료할 수 있음
onStop()	<ul style="list-style-type: none">- 액티비티가 사용자에게 더 이상 보이지 않을 때 호출됨- 액티비티가 소멸되거나 또 다른 액티비티가 화면을 가릴 때 호출됨- 액티비티가 이 상태에 들어가면 시스템은 액티비티를 강제 종료할 수 있음
onDestroy()	<ul style="list-style-type: none">- 액티비티가 소멸되어 없어지기 전에 호출됨- 이 메소드는 액티비티가 받는 마지막 호출이 됨- 액티비티가 애플리케이션에 의해 종료되거나(finish() 메소드 호출) 시스템이 강제로 종료시키는 경우에 호출될 수 있음- 위의 두 가지 경우를 구분할 때 isFinishing() 메소드를 이용함- 액티비티가 이 상태에 들어가면 시스템은 액티비티를 강제 종료할 수 있음



수명주기 확인하기 예제



수명주기 확인하기 예제

- 액티비티 상태에 따른 수명주기 확인하기
- 상태 메소드 별로 토스트 메시지 추가

XML 레이아웃 정의

메인 액티비티 코드 작성

- 입력상자와 버튼이 있는 레이아웃
- 상태 메소드 별로 토스트 메시지 코드 추가



메인 액티비티 코드 만들기

```
@Override  
protected void onDestroy() {  
    super.onDestroy();  
    Toast.makeText(getApplicationContext(), "onDestroy ...", Toast.LENGTH_LONG).show();  
}  
  
@Override  
protected void onPause() {  
    super.onPause();  
    saveState();  
    Toast.makeText(getApplicationContext(), "onPause ...", Toast.LENGTH_LONG).show();  
}  
  
@Override  
protected void onRestart() {  
    super.onRestart();  
    Toast.makeText(getApplicationContext(), "onRestart ...", Toast.LENGTH_LONG).show();  
}
```

현재 상태 저장

Continued..



메인 액티비티 코드 만들기 (계속)

```
@Override  
protected void onResume() {  
    super.onResume();  
    restoreState();  
    Toast.makeText(getApplicationContext(), "onResume...", Toast.LENGTH_LONG).show();  
}  
  
@Override  
protected void onStart() {  
    super.onStart();  
    Toast.makeText(getApplicationContext(), "onStart ...", Toast.LENGTH_LONG).show();  
}  
  
@Override  
protected void onStop() {  
    super.onStop();  
    Toast.makeText(getApplicationContext(), "onStop ...", Toast.LENGTH_LONG).show();  
}
```

현재 상태 복원

Continued..



메인 액티비티 코드 만들기 (계속)

```
protected void saveState() {
    SharedPreferences pref = getSharedPreferences("pref", Activity.MODE_PRIVATE);
    SharedPreferences.Editor editor = pref.edit();
    editor.putString( "name", nameInput.getText().toString() );
    editor.commit();
}

protected void clearState() {
    SharedPreferences pref = getSharedPreferences("pref", Activity.MODE_PRIVATE);
    SharedPreferences.Editor editor = pref.edit();
    editor.clear();
    editor.commit();
}
```



프래그먼트를 사용하는 방법에 대한 이해



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



-  **프로그먼트가 뭔가요?**
 - 프래그먼트란?
-  **프래그먼트로 화면을 만들어보고 싶어요**
 - 프래그먼트로 화면 만들기
-  **상단의 타이틀 부분을 바꾸고 텁도 만들고 싶어요**
 - 액션바와 텁 사용하기
-  **손가락으로 좌우 스크롤해서 화면을 넘길 수 있나요?**
 - 뷰페이저 만들기
-  **바로가기 메뉴는 어떻게 만드나요?**
 - 바로가기 메뉴 만들기

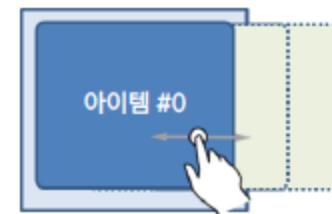
탭



바로가기 메뉴



뷰페이저





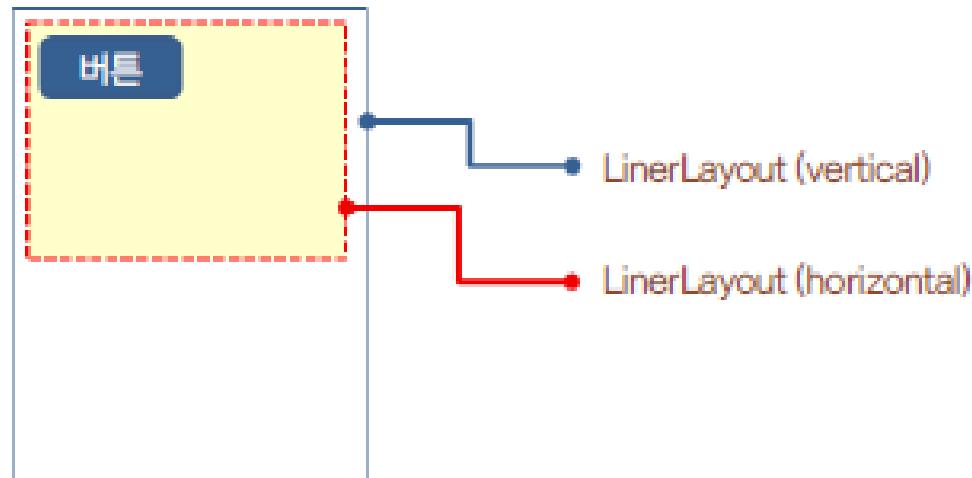
1.

프로그먼트란?



레이아웃 안에 레이아웃을 넣는 방식

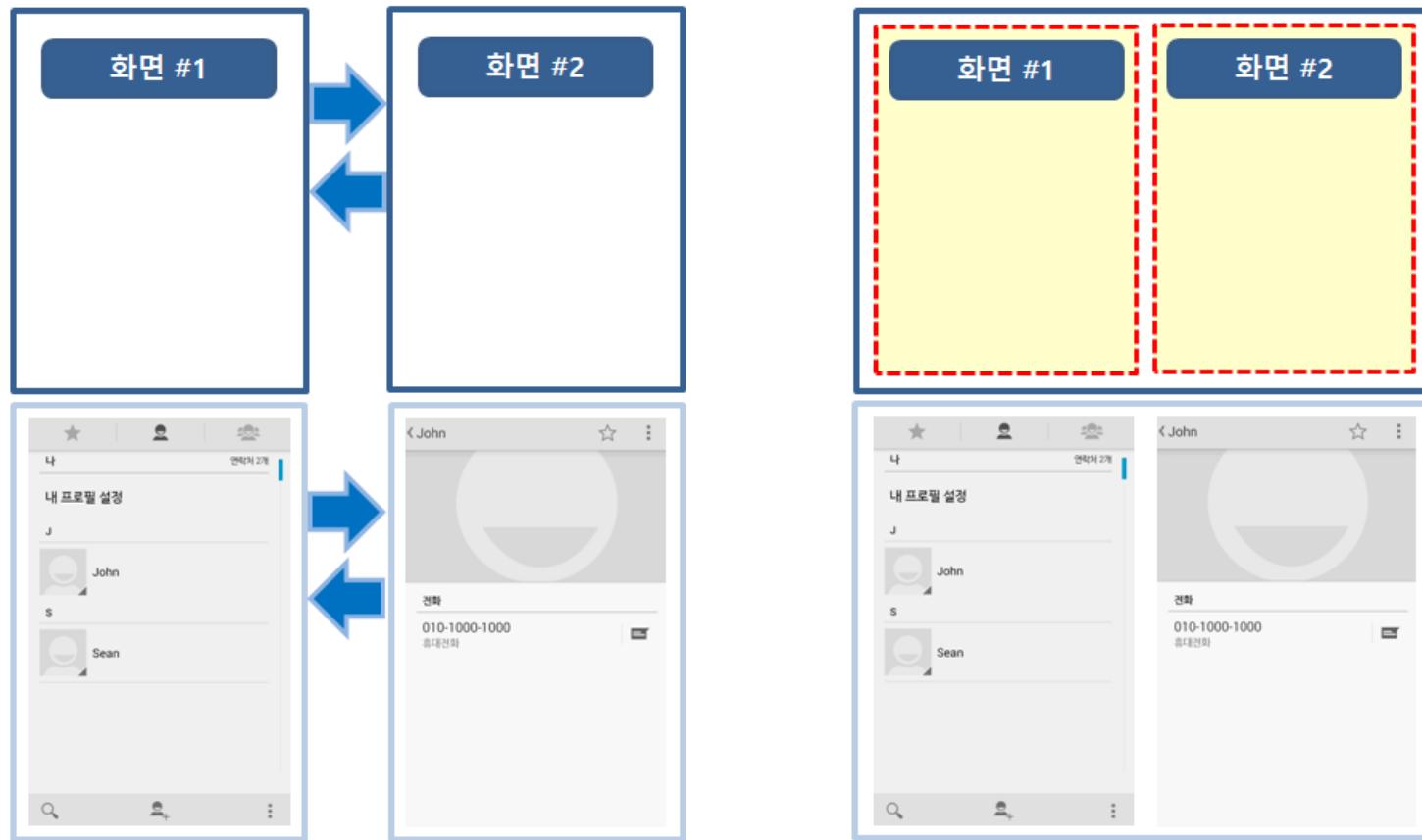
- 레이아웃 안에 부분 화면으로 레이아웃을 넣을 수 있음





한 화면에 여러 부분화면을 넣는 방식

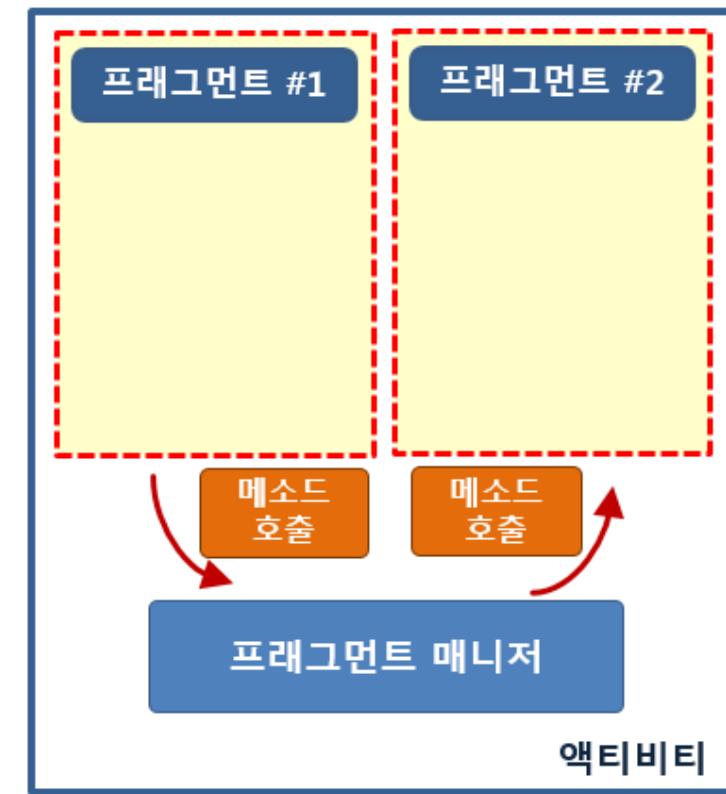
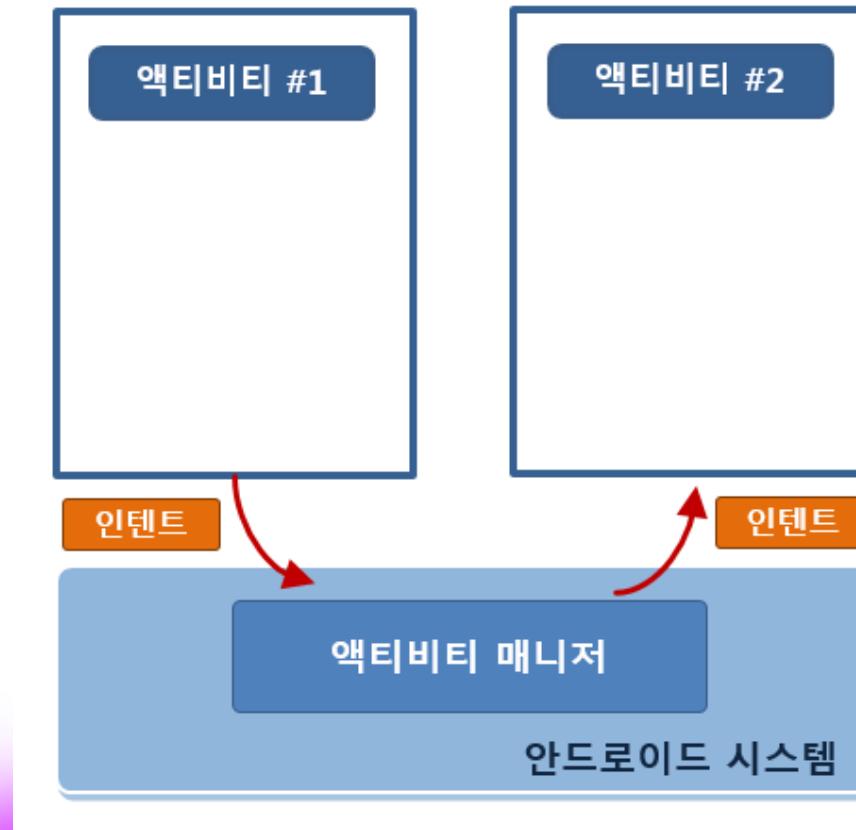
- 태블릿과 같이 큰 화면에서는 한 화면에 여러 부분화면을 넣는 것이 효율적임





부분화면을 독립적으로 만들기

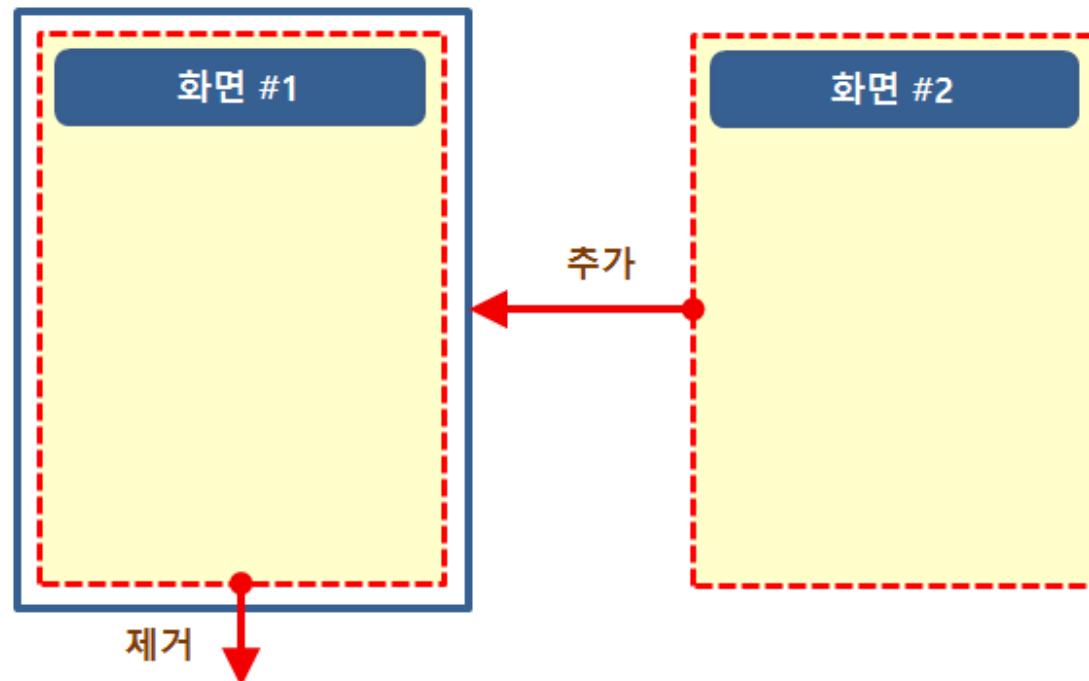
- 프래그먼트는 부분화면을 독립적으로 만들어주며 액티비티를 그대로 본떠 만든 것임





프로그먼트를 이용해 화면 전환하기

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

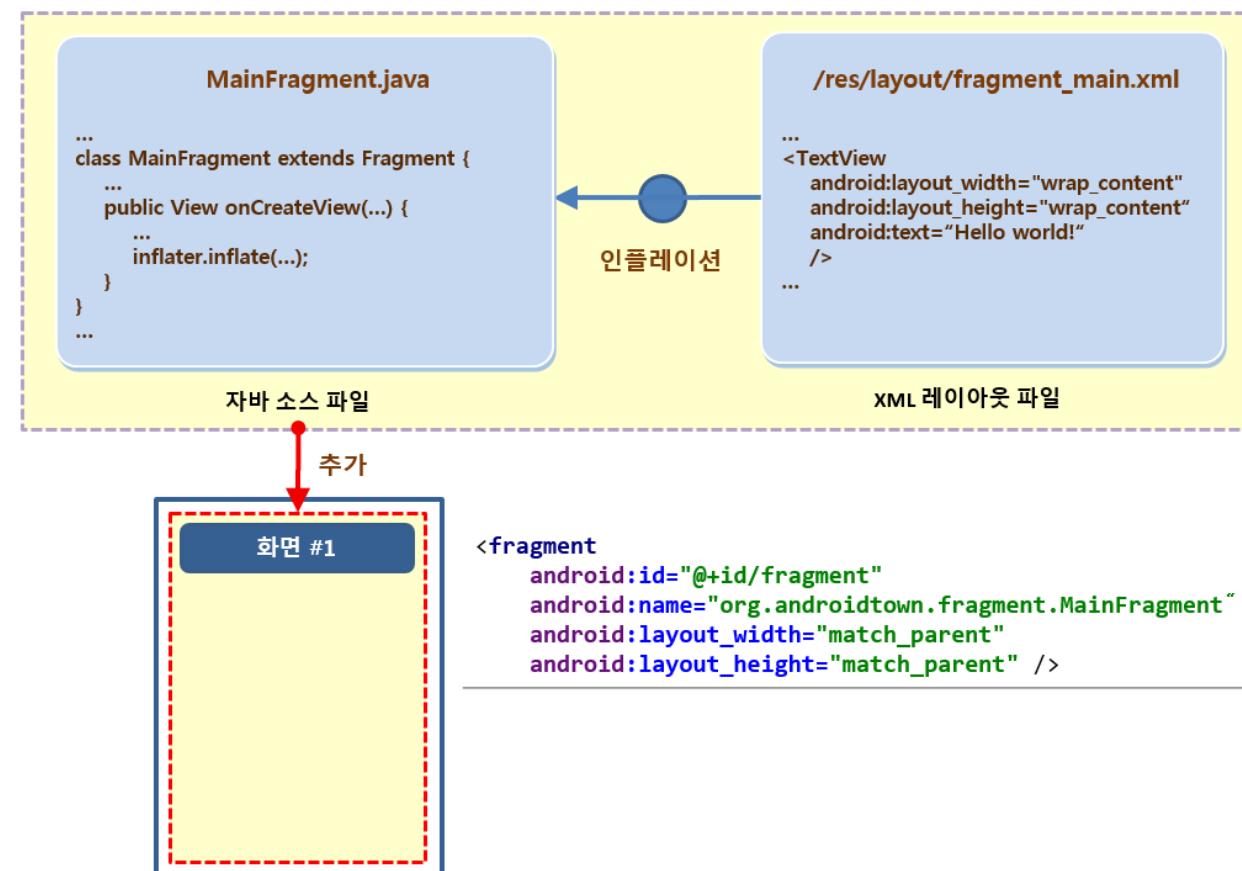




프래그먼트를 화면에 추가하기

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

하나의 프래그먼트





프래그먼트 만들어 화면에 추가하기

- 프래그먼트 추가

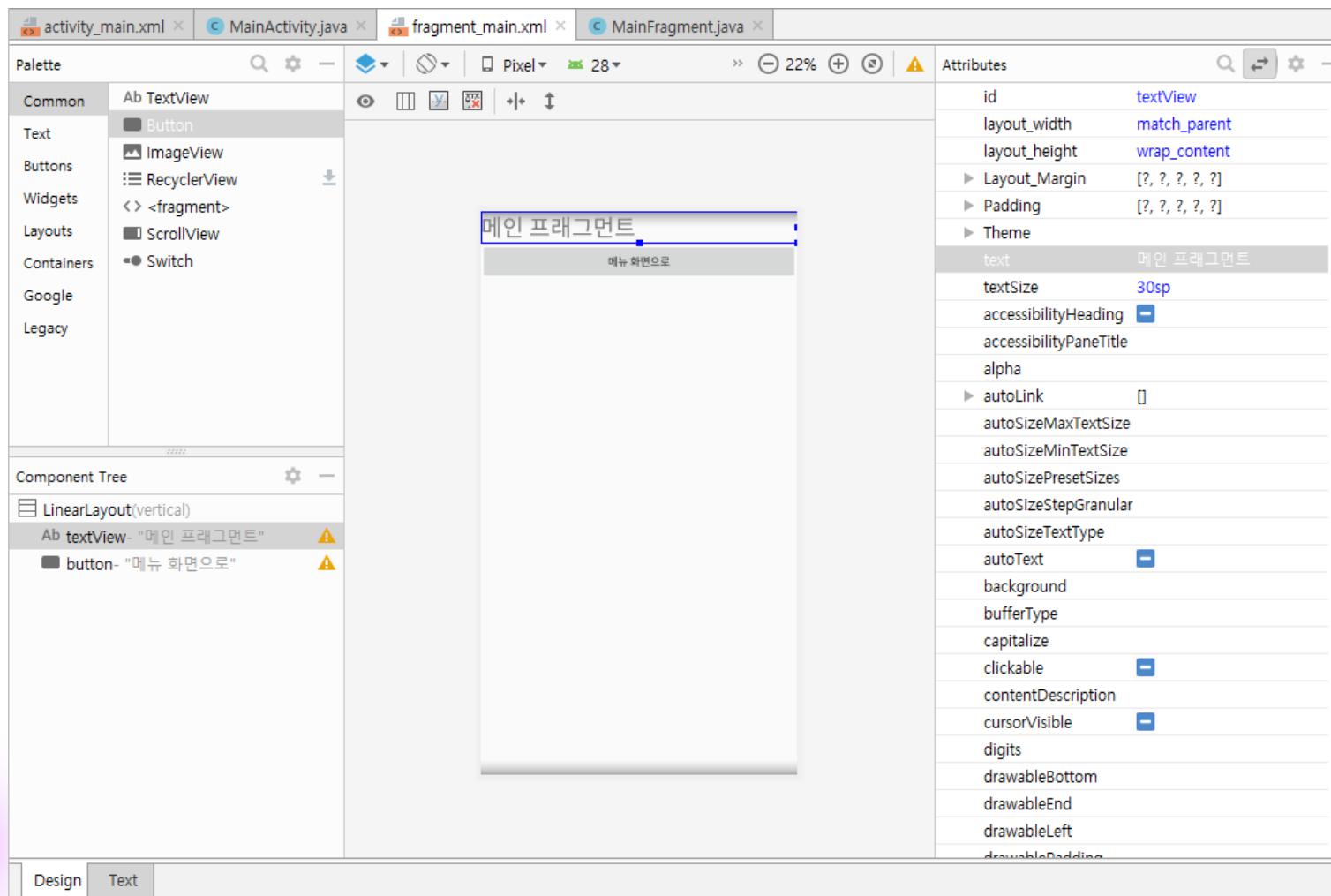
The screenshot shows the Android Studio interface with the following details:

- Project View:** Shows the project structure under "app".
- Toolbar:** Includes icons for back, forward, search, and other common operations.
- Editor Area:** Displays two tabs: "activity_main.xml" and "MainActivity.java".
- Context Menu:** Opened from the "app" folder, showing options like "New", "Link C++ Project with Gradle", and "Fragment".
- New Component Dialog:** A modal window titled "Configure Component" with the sub-titel "Creates a blank fragment that is compatible back to API level 4." It contains the following fields:
 - Fragment Name: `MainFragment`
 - Create layout XML? (checked)
 - Fragment Layout Name: `fragment_main`
 - Include fragment factory methods? (checked)
 - Include interface callbacks? (checked)
 - Source Language: Java
 - Target Source Set: main
- Bottom Buttons:** "Previous", "Next", "Cancel", and "Finish" buttons.



프래그먼트 만들어 화면에 추가하기

- 프래그먼트를 위한 화면 구성





프래그먼트 만들어 화면에 추가하기

- MainFragment.java 파일 만들고 프래그먼트 상속

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

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



프래그먼트 만들어 화면에 추가하기

- 메인 액티비티의 XML 레이아웃에 추가

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

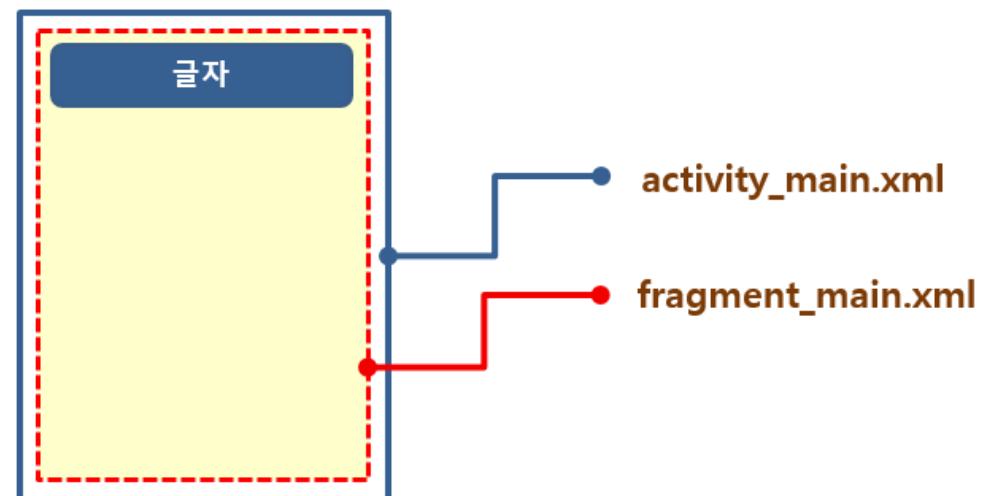
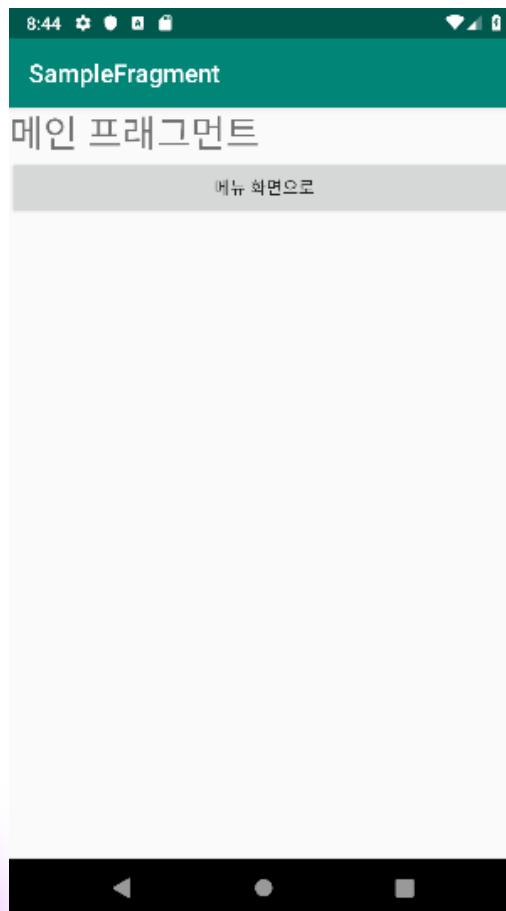
    <fragment
        android:id="@+id/mainFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:name="org.techtown.fragment.MainFragment" />

</RelativeLayout>
```



앱 실행 결과

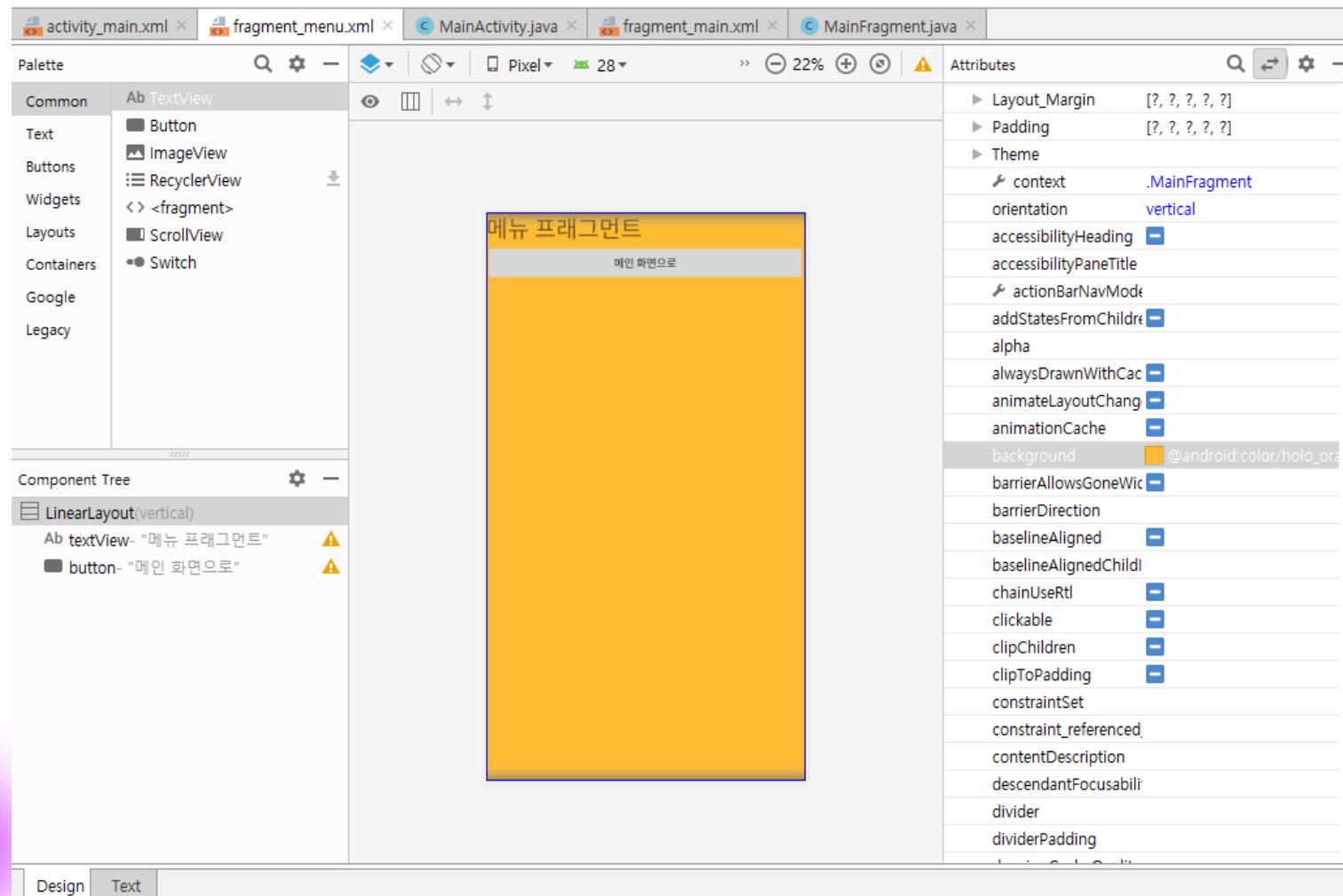
- 프래그먼트가 부분화면으로 추가됨





코드에서 프래그먼트 추가하기

- 메뉴 프래그먼트를 위한 XML 레이아웃 파일 만들고 화면 구성





코드에서 프래그먼트 추가하기

- 프래그먼트 소스 코드 입력

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

```
public class MainActivity extends Fragment {

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

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

        return rootView;
    }
}
```



코드에서 프래그먼트 추가하기

- 메인 액티비티 코드에서 프래그먼트 매니저 사용

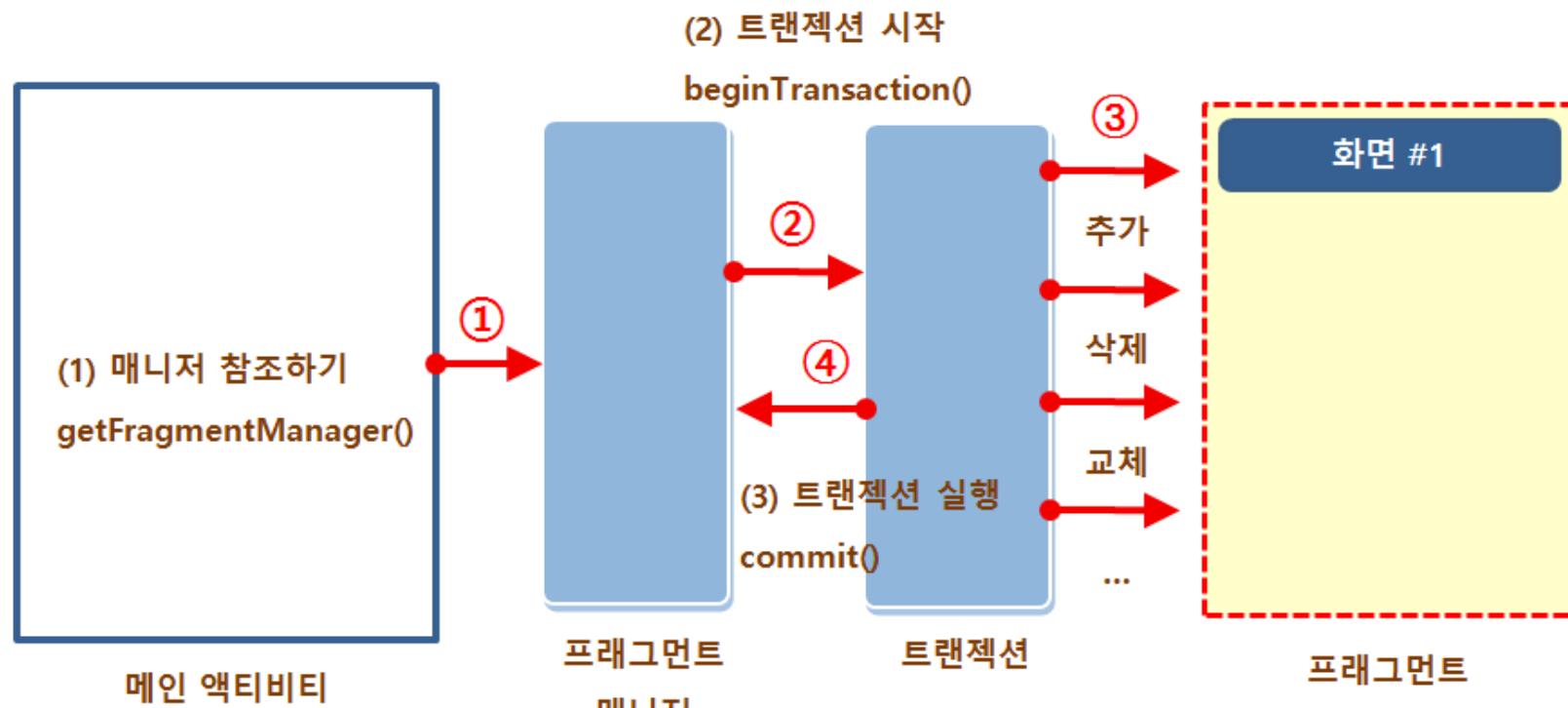
참조파일 SampleFragment>/app/java/org.techtown.fragment/MainActivity.java

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



코드에서 프래그먼트 추가하기

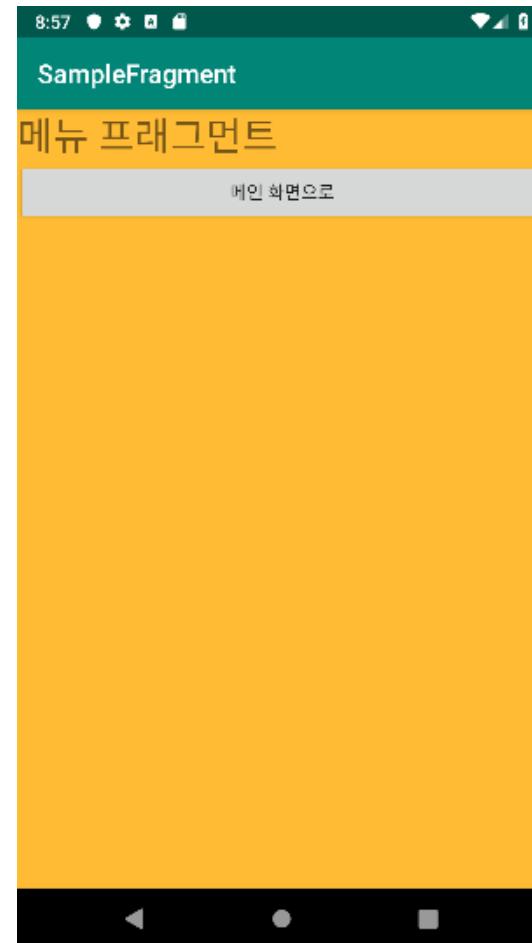
- 프래그먼트가 액티비티의 프래그먼트 매니저를 통해 의사소통하는 방식





앱 실행 결과

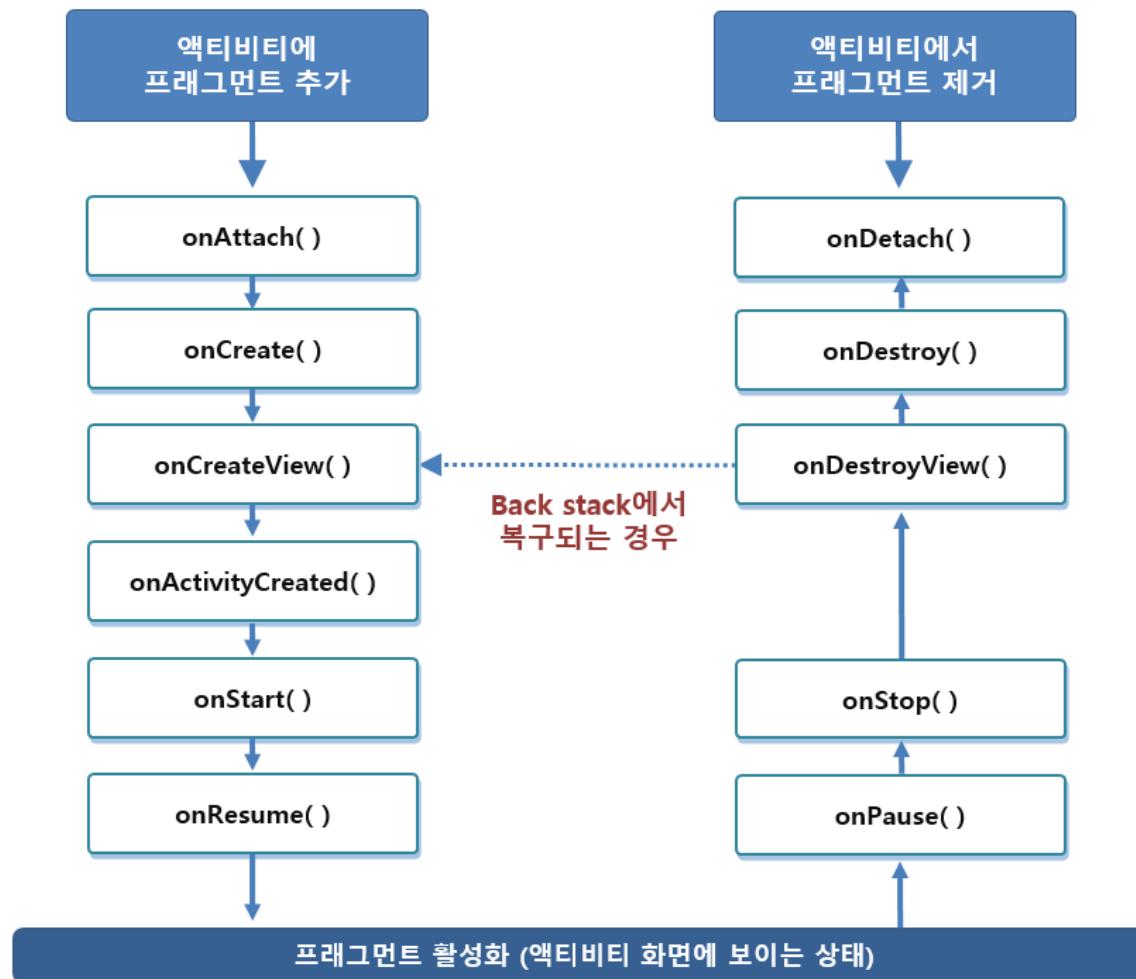
- 프래그먼트 안의 버튼 클릭 시 액티비티를 통해 프래그먼트 전환





프래그먼트의 수명주기

- 독립적인 상태 유지를 위해 수명주기 메소드를 제공함





2.

프래그먼트로 화면 만들기



두 개의 프래그먼트로 구성된 이미지 뷰어

- 이미지 복사한 후 프래그먼트 구성

The screenshot shows the Android Studio interface with the following details:

- Project Tab:** Shows the project structure with files like `activity_main.xml`, `MainActivity.java`, and `fragment_list.xml`.
- Toolbars:** Includes tabs for `activity_main.xml`, `MainActivity.java`, and `fragment_list.xml`. Other toolbars include `Palette`, `Search`, `Pixel`, `Attributes`, and `Component Tree`.
- Palette:** The `Common` tab is selected, showing categories like `Text`, `Buttons`, `Widgets`, `Layouts`, `Containers`, `Google`, and `Legacy`. A specific `Button` is selected.
- Component Tree:** Shows a `LinearLayout` containing three `button` elements with the following labels:
 - button- "첫 번째 이미지"
 - button2- "두 번째 이미지"
 - button3- "세 번째 이미지"
- Design View:** Displays a preview of the fragment layout with three buttons stacked vertically, each labeled with its respective text.
- Attributes Panel:** Shows the properties for the selected `button3` element. Key settings include:
 - `id`: `button3`
 - `layout_width`: `match_parent`
 - `layout_height`: `wrap_content`
 - `Layout_Margin`: `[?, ?, ?, ?, ?]`
 - `Padding`: `[?, ?, ?, ?, ?]`
 - `Theme`: `text` is set to `세 번째 이미지`.
- Bottom Navigation:** Shows tabs for `Design` and `Text`.



두 개의 프래그먼트로 구성된 이미지 뷰어

- 첫번째 프래그먼트에 코드 추가

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

```
public class ListFragment extends Fragment {

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

    public ImageSelectionCallback callback;

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

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

    @Nullable
    @Override
```



두 개의 프래그먼트로 구성된 이미지 뷰어

- 첫번째 프래그먼트에 코드 추가

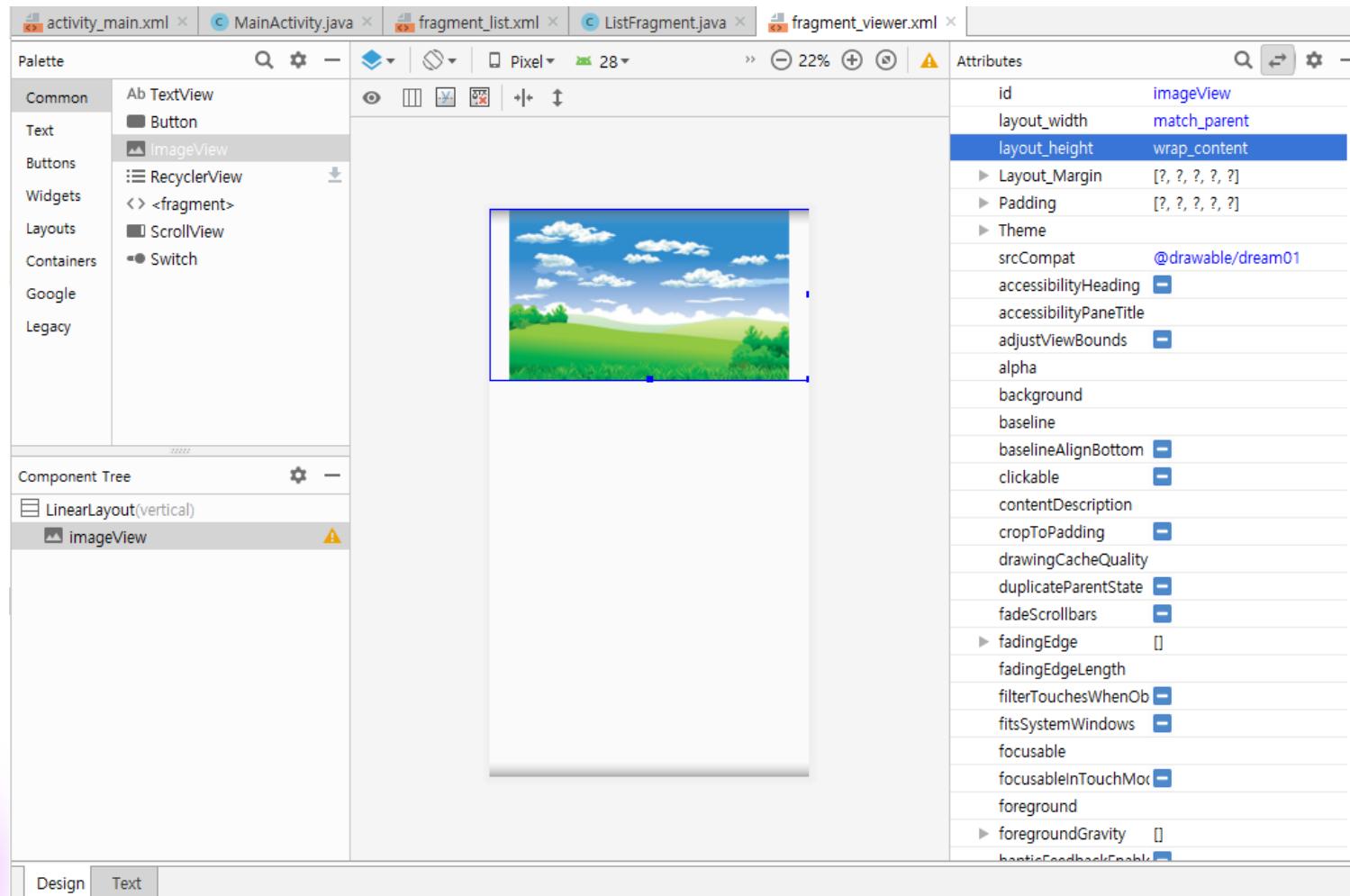
```
public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,
                        @Nullable Bundle savedInstanceState) {
    ViewGroup rootView = (ViewGroup) inflater.inflate(R.layout.fragment_list, container, false);
    Button button = rootView.findViewById(R.id.button);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

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



두 개의 프래그먼트로 구성된 이미지 뷰어

- 두번째 프래그먼트에 이미지뷰 추가





두 개의 프래그먼트로 구성된 이미지 뷰어

- 두번째 프래그먼트에 이미지뷰 추가

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

```
public class ViewerFragment extends Fragment {
    ImageView imageView;

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

        imageView = rootView.findViewById(R.id.imageView);
        return rootView;
    }

    public void setImage(int resId) {
        imageView.setImageResource(resId);
    }
}
```



두 개의 프래그먼트로 구성된 이미지 뷰어

- 메인 액티비티의 XML 레이아웃에 두 개의 프래그먼트 추가

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

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



두 개의 프래그먼트로 구성된 이미지 뷰어

- 메인 액티비티 코드에서 버튼 클릭 시 이미지 설정

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

```
public class MainActivity extends AppCompatActivity
    implements ListFragment.ImageSelectionCallback {
    ListFragment listFragment;
    ViewerFragment viewerFragment;

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

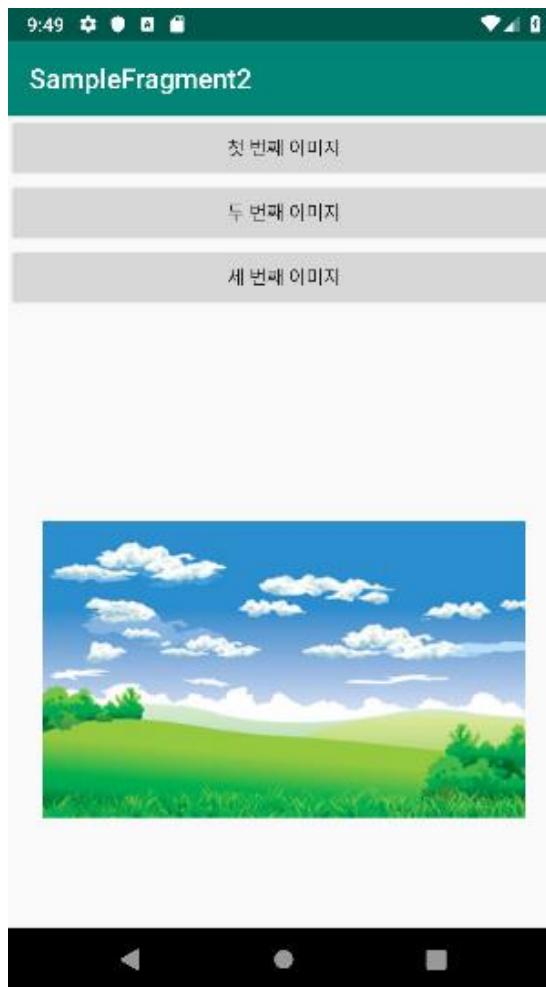
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

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

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



앱 실행 결과





3.

액션바 사용하기



안드로이드의 메뉴

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

옵션 메뉴

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

컨텍스트 메뉴

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



메뉴 사용 방식의 기본

- `onCreateOptionsMenu()`와 `onCreateContextMenu()` 메소드 재정의
- 메뉴 선택 시의 이벤트 처리
 - ✓ 컨텍스트 메뉴의 아이템 선택 : `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));  
}
```

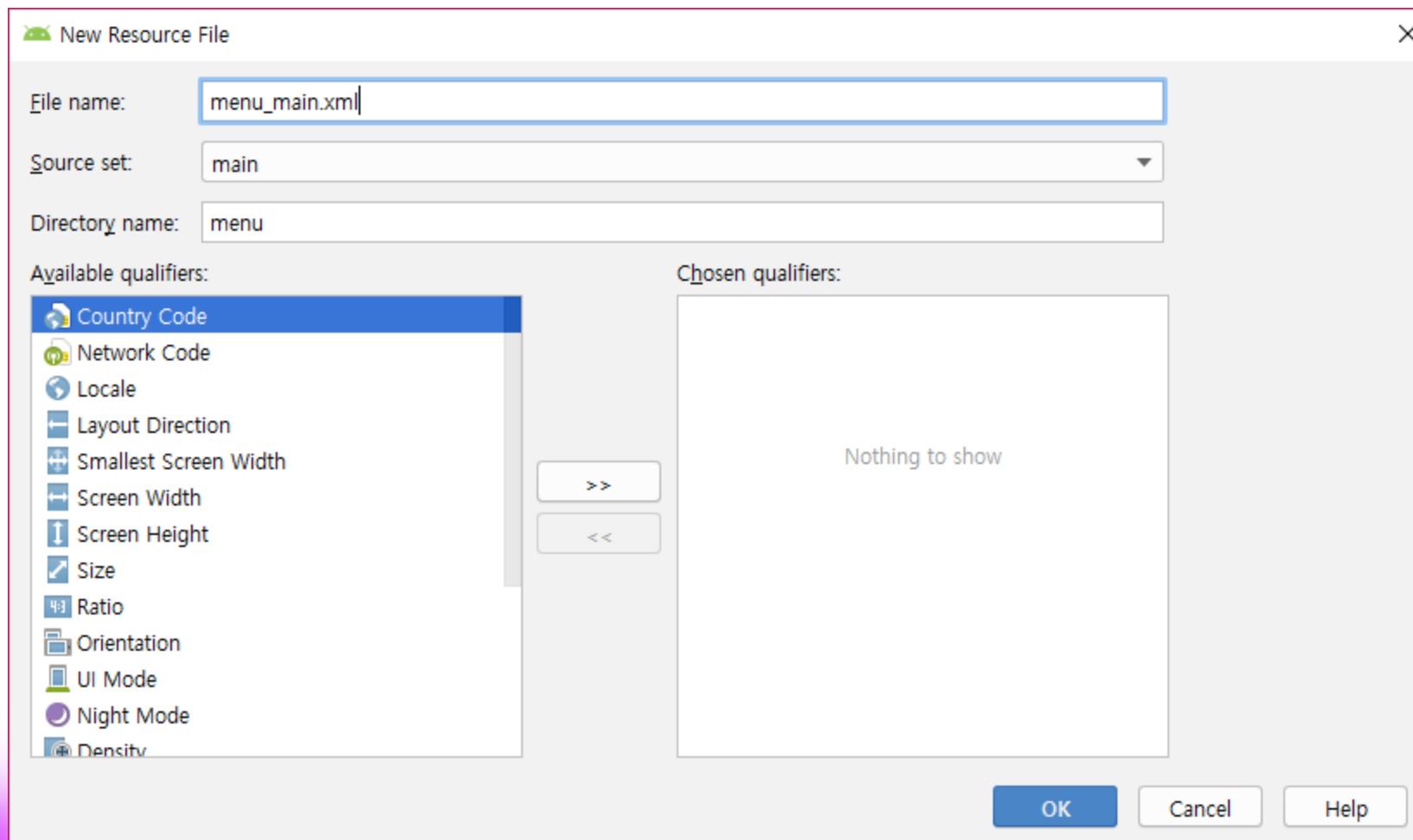
```
@Override  
public boolean onContextItemSelected(MenuItem item) {  
    return (applyMenuChoice(item) || super.onContextItemSelected(item));  
}
```

[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

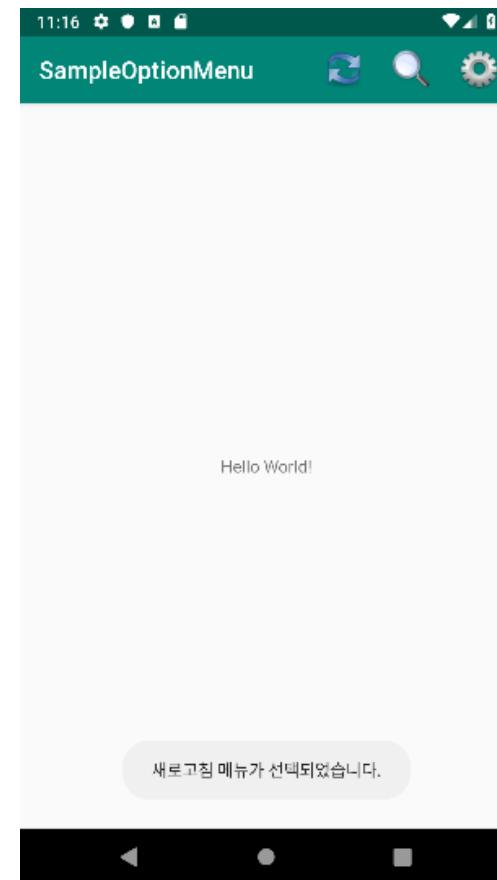
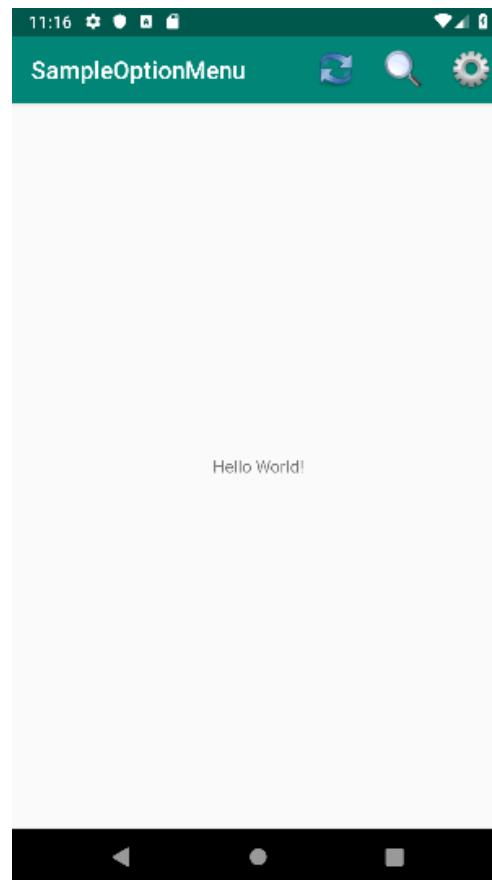
```
public boolean onCreateOptionsMenu(Menu menu) {  
    getMenuInflater().inflate(R.menu.menu_main, menu);  
    return true;  
}
```

@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





액션바에 입력상자 넣기

참조파일 SampleActionBar2>/app/res/layout/search_layout.xml

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

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

① 문자열 표시를 위한 텍스트뷰

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

② 검색어를 입력하기 위한 입력상자

```
</LinearLayout>
```



액션바에 입력상자 넣기

참조파일 SampleActionBar2>/app/res/menu/menu_main.xml

중략...

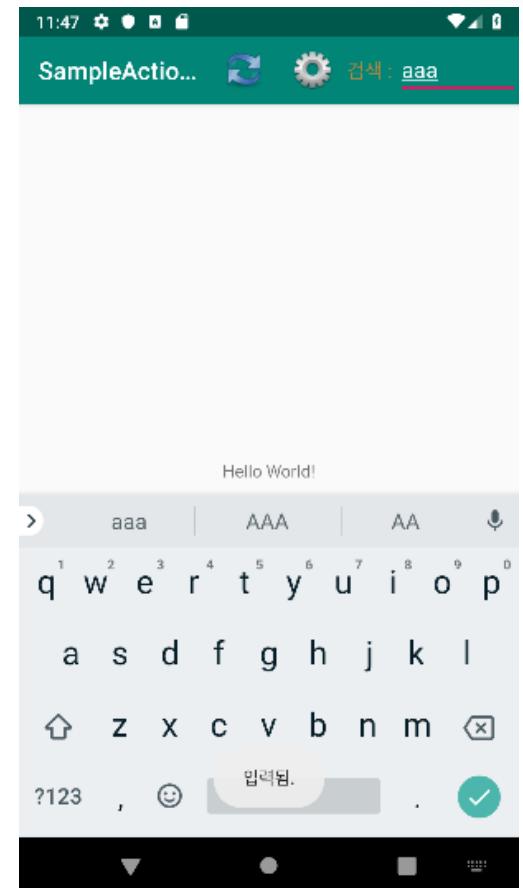
```
<item android:id="@+id/menu_search"
    android:title="검색"
    android:orderInCategory="102"
    app:showAsAction="always/withText"
    app:actionLayout="@layout/search_layout" /> → 메뉴가 화면에 보이는 방식을 정의한 레이아웃 설정
```

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

중략...

```
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_main, menu); → ❶ XML로 정의된 메뉴 정보를 인플레이션하여 메모리에 로딩

    View v = menu.findItem(R.id.menu_search).getActionView(); → ❷ 메뉴 아이템 중에서 검색을 위해 정의한 아이템을 뷰 객체로 참조
    if (v != null) {
        EditText editText = v.findViewById(R.id.editText); → ❸ 검색을 위한 메뉴 아이템 안에 정의한 입력상자 객체 참조
        if (editText != null) {
            editText.setOnEditorActionListener(new TextView.OnEditorActionListener() {→ ❹ 입력상자 객체에 리
                @Override
                public boolean onEditorAction(TextView v, int actionId, KeyEvent event) { 스너 설정
                    Toast.makeText(getApplicationContext(), "입력됨.", Toast.LENGTH_LONG).show();
                    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.android.support:design:28.0.0'
```

```
}
```



메인 액티비티의 XML 레이아웃 작성

참조파일 SampleTab>/app/res/layout/activity_main.xml

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

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

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

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

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



메인 액티비티의 XML 레이아웃 작성

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

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

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

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

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

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



각 템을 위한 프래그먼트 만들기

The screenshot shows the Android Studio interface with the following details:

- Top Bar:** activity_main.xml, fragment1.xml, Fragment1.java, MainActivity.java
- Palette:** Common (selected), Text, Buttons, Widgets, Layouts, Containers, Google, Legacy.
- Component Tree:** LinearLayout(vertical) containing a button labeled "첫 번째".
- Design View:** Shows a blue rectangular fragment with a white header bar containing the text "첫 번째".
- Attributes Panel:** Lists various button properties:
 - id: button
 - layout_width: match_parent
 - layout_height: wrap_content
 - Layout_Margin: [?, ?, ?, ?, ?]
 - Padding: [?, ?, ?, ?, ?]
 - Theme
 - text: 첫 번째
 - accessibilityHeading
 - accessibilityPaneTitle
 - alpha
 - autoLink
 - autoText
 - background: (selected)
 - bufferType
 - capitalize
 - clickable
 - contentDescription
 - cursorVisible
 - digits
 - drawableBottom
 - drawableEnd
 - drawableLeft
 - drawablePadding
 - drawableRight
 - drawableStart
 - drawableTop
 - drawingCacheQuality
 - duplicateParentState
 - editable
- Bottom Navigation:** Design (selected), Text



프로그먼트의 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





하단 탭 보여주기

- 하단 탭 구성을 위한 XML 생성

참조파일 SampleTab2>/app/res/menu/menu_bottom.xml

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

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

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

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

</menu>
```



하단 탭 보여주기

- BottomNavigationView 위젯으로 구성

참조파일 SampleTab2>/app/res/layout/activity_main.xml

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

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

    <android.support.design.widget.BottomNavigationView
        android:id="@+id/bottom_navigation"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginEnd="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" />

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



하단 탭 보여주기

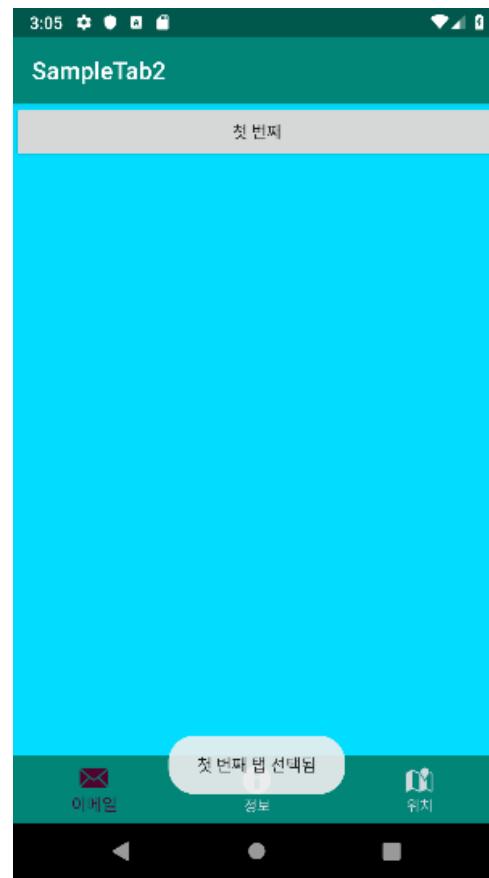
- 코드 구성

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

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



하단 탭 보여주기





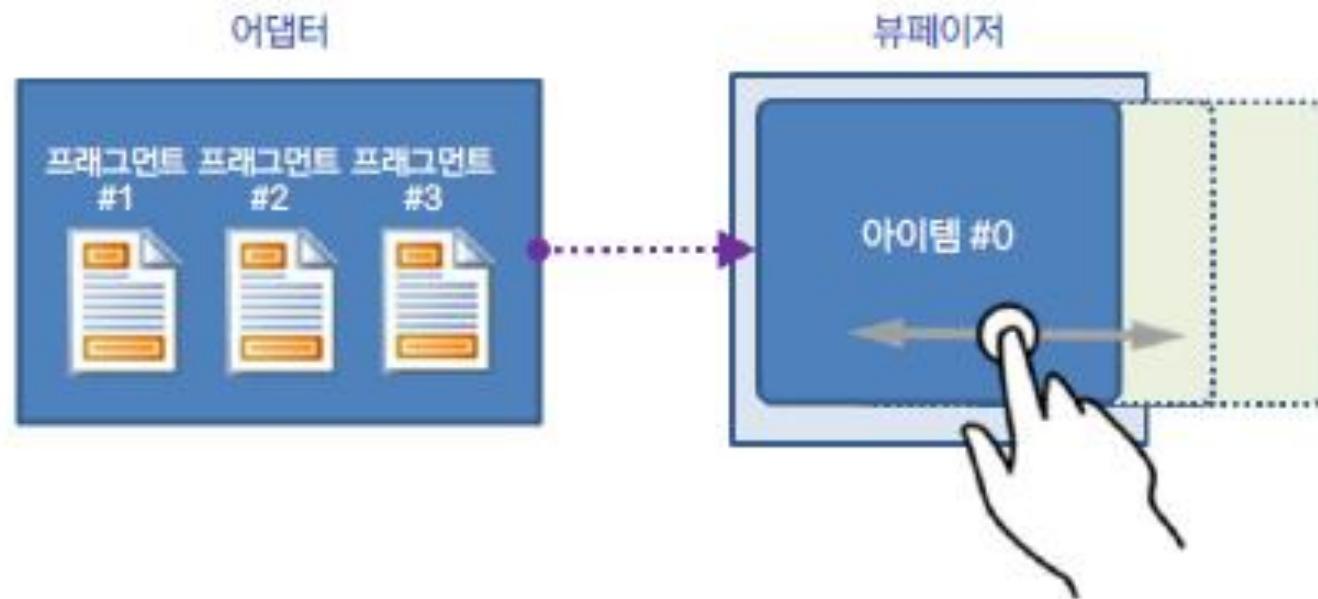
5.

뷰페이지 만들기



뷰페이지

- 손가락으로 좌우 스크롤하여 프래그먼트 화면 넘김





레이아웃에 뷰페이저 위젯 추가

The screenshot shows the Android Studio Layout Editor interface. At the top, there are tabs for `activity_main.xml` and `MainActivity.java`. The main area consists of several panels:

- Palette:** On the left, under the **Layouts** category, the **ViewPager** widget is selected.
- Component Tree:** In the center-left, it shows a `LinearLayout` (vertical) containing a `button- "Button"` and a `pager`.
- Design View:** The central workspace displays a `BUTTON` and a `pager` view.
- Attributes:** A large panel on the right lists numerous attributes for the `pager` view, such as `id`, `layout_width`, `layout_height`, `Layout_Margin`, `Padding`, `Theme`, and many others.



뷰페이지를 위한 어댑터 생성

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

```
public class MainActivity extends AppCompatActivity {

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

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

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



어댑터에 프래그먼트 추가

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

```
public class MainActivity extends AppCompatActivity {
    ViewPager pager;

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

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

        MyPagerAdapter adapter = new MyPagerAdapter(getSupportFragmentManager());

        Fragment1 fragment1 = new Fragment1();
        adapter.addItem(fragment1);

        Fragment2 fragment2 = new Fragment2();
        adapter.addItem(fragment2);

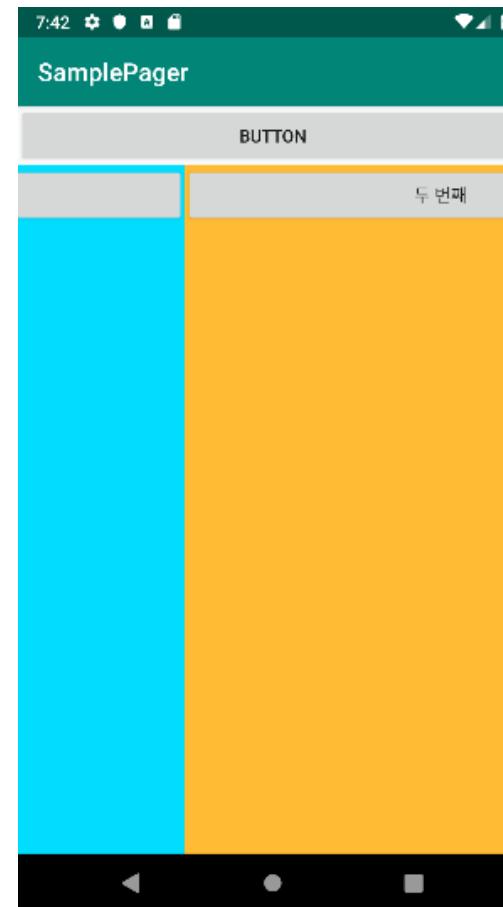
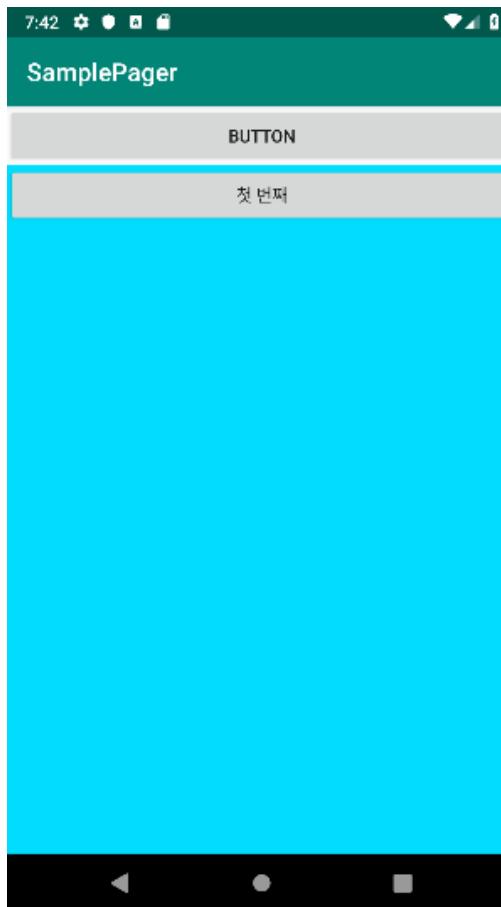
        Fragment3 fragment3 = new Fragment3();
        adapter.addItem(fragment3);

        pager.setAdapter(adapter);
    }
}
```

중략...



뷰페이지





특정 프래그먼트 화면을 보여주기

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

중략...

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    Button button = findViewById(R.id.button);  
    button.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View view) {  
            pager.setCurrentItem(1);  
        }  
    });  
  
    pager = findViewById(R.id.pager);  
    pager.setOffscreenPageLimit(3);
```

중략...



타이틀 스트립 보여주기

참조파일 SamplePager>/app/res/layout/activity_main.xml

중략...

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

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

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

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

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

중략...



타이틀 스트립 보여주기

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

중략...

```
class MyPagerAdapter extends FragmentStatePagerAdapter {
```

중략...

```
@Nullable
```

```
@Override
```

```
public CharSequence getPageTitle(int position) {
```

```
    return "페이지" + position;
```

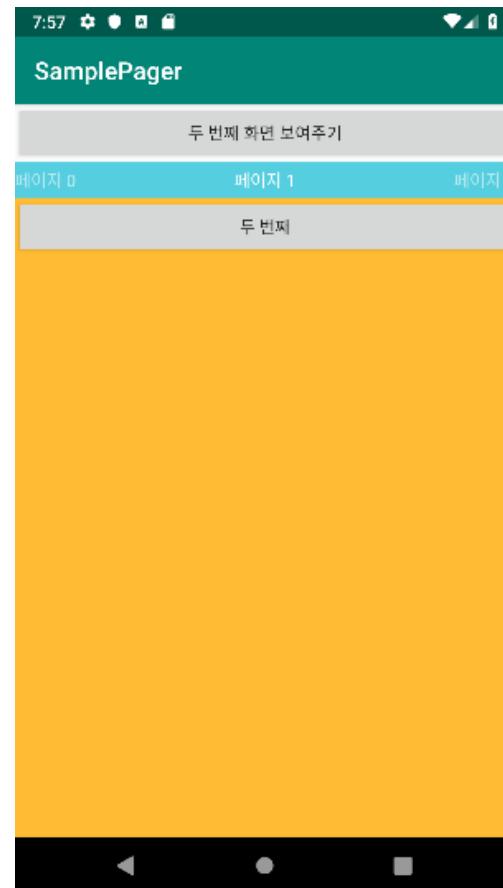
```
}
```

```
}
```

중략...



타이틀 스트립 보여주기





6.

바로가기 메뉴 만들기



바로가기 메뉴

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



Create New Project

Choose your project

Phone and Tablet Wear OS TV Android Auto Android Things

Add No Activity Basic Activity Empty Activity Bottom Navigation Activity

Fullscreen Activity Master/Detail Flow Navigation Drawer Activity Google Maps Activity

Navigation Drawer Activity

Creates a new Activity with a Navigation Drawer.

Previous Next Cancel Finish

The screenshot shows the 'Create New Project' dialog in Android Studio. At the top, there are tabs for 'Phone and Tablet', 'Wear OS', 'TV', 'Android Auto', and 'Android Things'. Below the tabs, there are four main activity types: 'Add No Activity', 'Basic Activity', 'Empty Activity', and 'Bottom Navigation Activity'. Under each type, there is a preview image and a detailed description. The 'Navigation Drawer Activity' section is highlighted with a blue border. It includes a preview image of a smartphone with a navigation drawer open, a detailed description 'Creates a new Activity with a Navigation Drawer.', and four buttons at the bottom: 'Previous', 'Next', 'Cancel', and 'Finish'.



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 스튜디오에서 숨김 또는 대체했기 때문입니다. 회색 영역을 클릭하면 원문 코드가 나타납니다.

중략...

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

중략...

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

중략...



바로가기 메뉴

• 상단 액션바 설정

참조파일 SampleDrawer>/app/res/layout/activity_main.xml

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

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

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

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

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



바로가기 메뉴

- 상단 액션바 설정

```
<FrameLayout  
    android:id="@+id/container"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    app:layout_behavior="@string/appbar_scrolling_view_behavior">  
  
</FrameLayout>  
  
</android.support.design.widget.CoordinatorLayout>  
  
<android.support.design.widget.NavigationView  
    android:id="@+id/nav_view"  
    android:layout_width="wrap_content"  
    android:layout_height="match_parent"  
    android:layout_gravity="start"  
    android:fitsSystemWindows="true"  
    app:headerLayout="@layout/nav_header_main"  
    app:menu="@menu/activity_main_drawer" />  
  
</android.support.v4.widget.DrawerLayout>
```



바로가기 메뉴

- 코드에서 NavigationDrawer 설정

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    toolbar = findViewById(R.id.toolbar);  
    setSupportActionBar(toolbar);  
  
    DrawerLayout drawer = findViewById(R.id.drawer_layout);  
    ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(  
        this, drawer, toolbar, R.string.navigation_drawer_open, R.string.navigation_drawer_close);  
    drawer.addDrawerListener(toggle);  
    toggle.syncState();  
  
    NavigationView navigationView = findViewById(R.id.nav_view);  
    navigationView.setNavigationItemSelectedListener(this);  
  
    fragment1 = new Fragment1();  
    fragment2 = new Fragment2();  
    fragment3 = new Fragment3();  
  
    getSupportFragmentManager().beginTransaction().add(R.id.container, fragment1).commit();  
}
```



바로가기 메뉴

- 바로가기 메뉴의 메뉴아이템 클릭 시 실행될 코드 추가

```
@Override  
public boolean onNavigationItemSelected(MenuItem item) {  
    int id = item.getItemId();  
  
    if (id == R.id.menu1) {  
        Toast.makeText(this, "첫 번째 메뉴 선택됨.", Toast.LENGTH_LONG).show();  
        onFragmentSelected(0, null);  
    } else if (id == R.id.menu2) {  
        Toast.makeText(this, "두 번째 메뉴 선택됨.", Toast.LENGTH_LONG).show();  
        onFragmentSelected(1, null);  
    } else if (id == R.id.menu3) {  
        Toast.makeText(this, "세 번째 메뉴 선택됨.", Toast.LENGTH_LONG).show();  
        onFragmentSelected(2, null);  
    }  
  
    DrawerLayout drawer = findViewById(R.id.drawer_layout);  
    drawer.closeDrawer(GravityCompat.START);  
    return true;  
}
```



바로가기 메뉴

- 바로가기 메뉴의 메뉴 XML 구성

참조파일 SampleDrawer>/app/res/menu/activity_main_drawer.xml

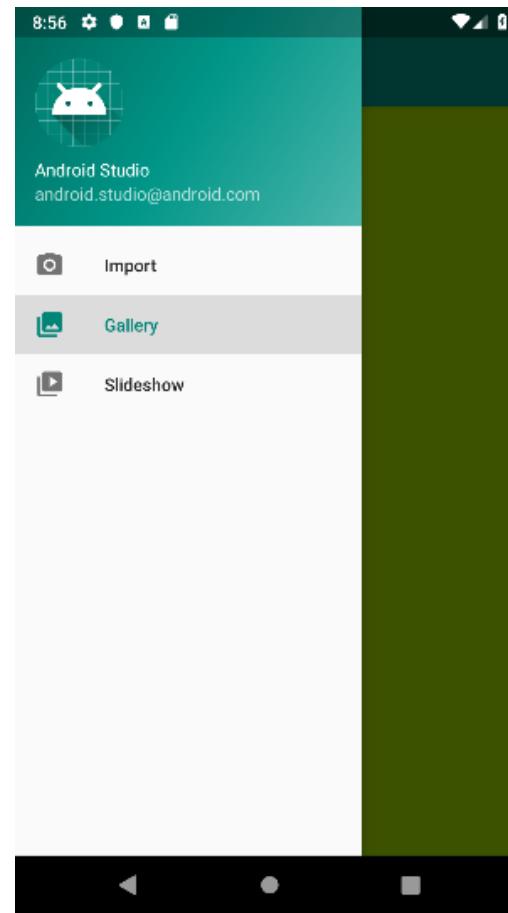
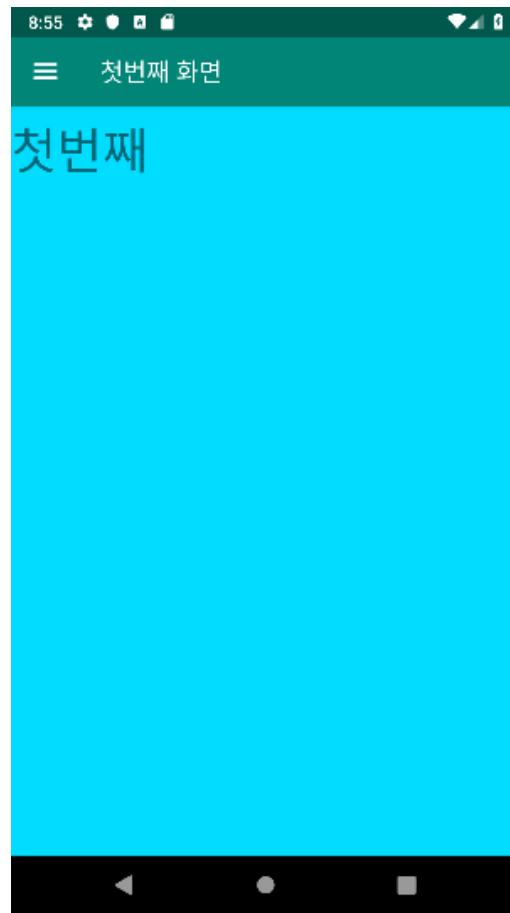
```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    tools:showIn="navigation_view">

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

</menu>
```



바로가기 메뉴





서비스와 브로드캐스트 수신자에 대한 이해



- 1 서비스
- 2 브로드캐스트 수신자 이해하기
- 3 위험권한 부여하기
- 4 리소스와 매니페스트 이해하기
- 5 그래들 이해하기



- ① 서비스는 어떻게 사용하나요?

 - 서비스 살펴보기
- ② 수신자는 어떻게 사용하나요?

 - 수신자 살펴보기
- ③ 위험권한을 위한 코드는 어떻게 추가하나요?

 - 위험권한 부여하기
- ④ 리소스에 대해 좀 더 알고 싶어요

 - 리소스와 매니페스트 이해하기
- ⑤ 그레들이 무엇인가요?

 - 그레들 이해하기





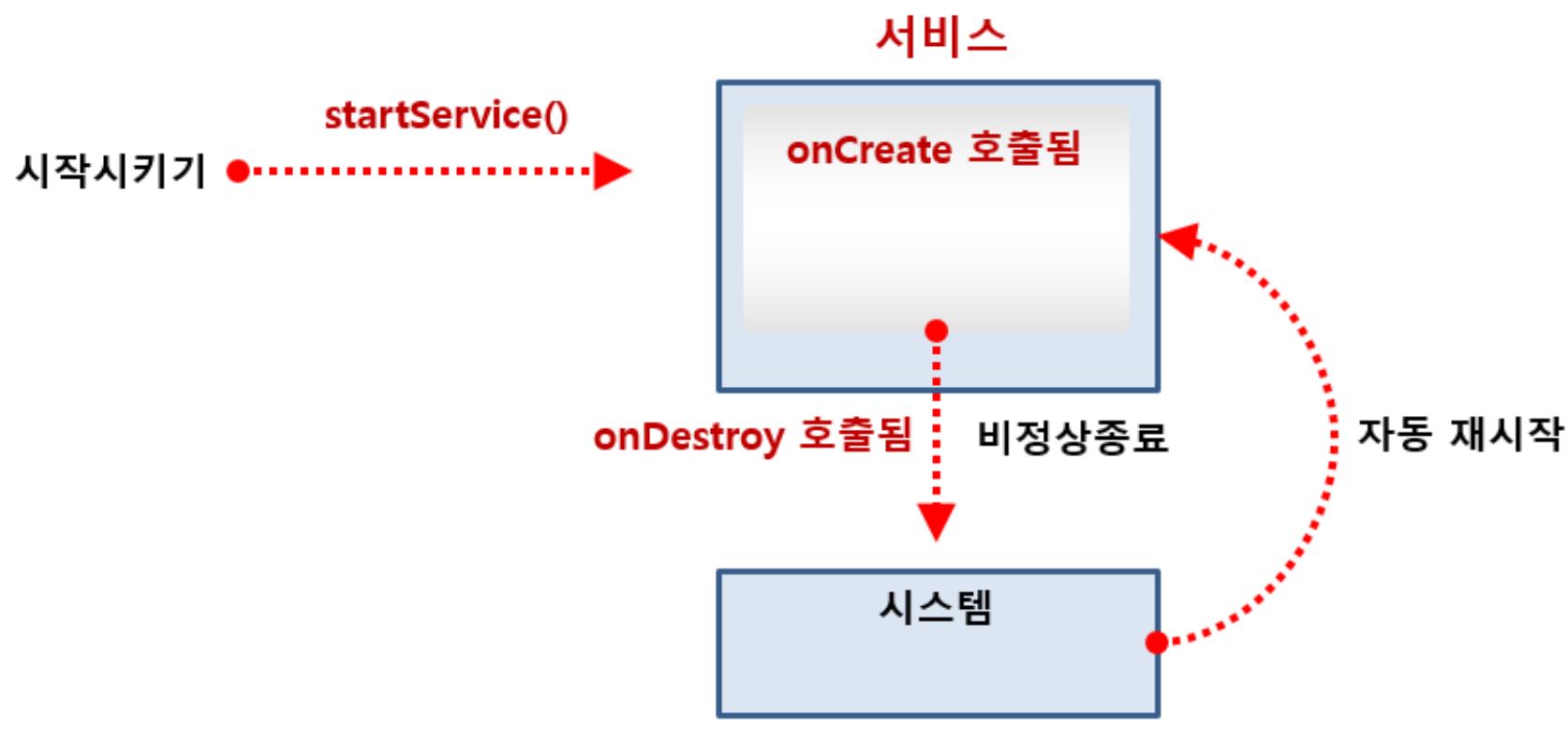
1.

서비스



자동으로 재 시작되는 서비스

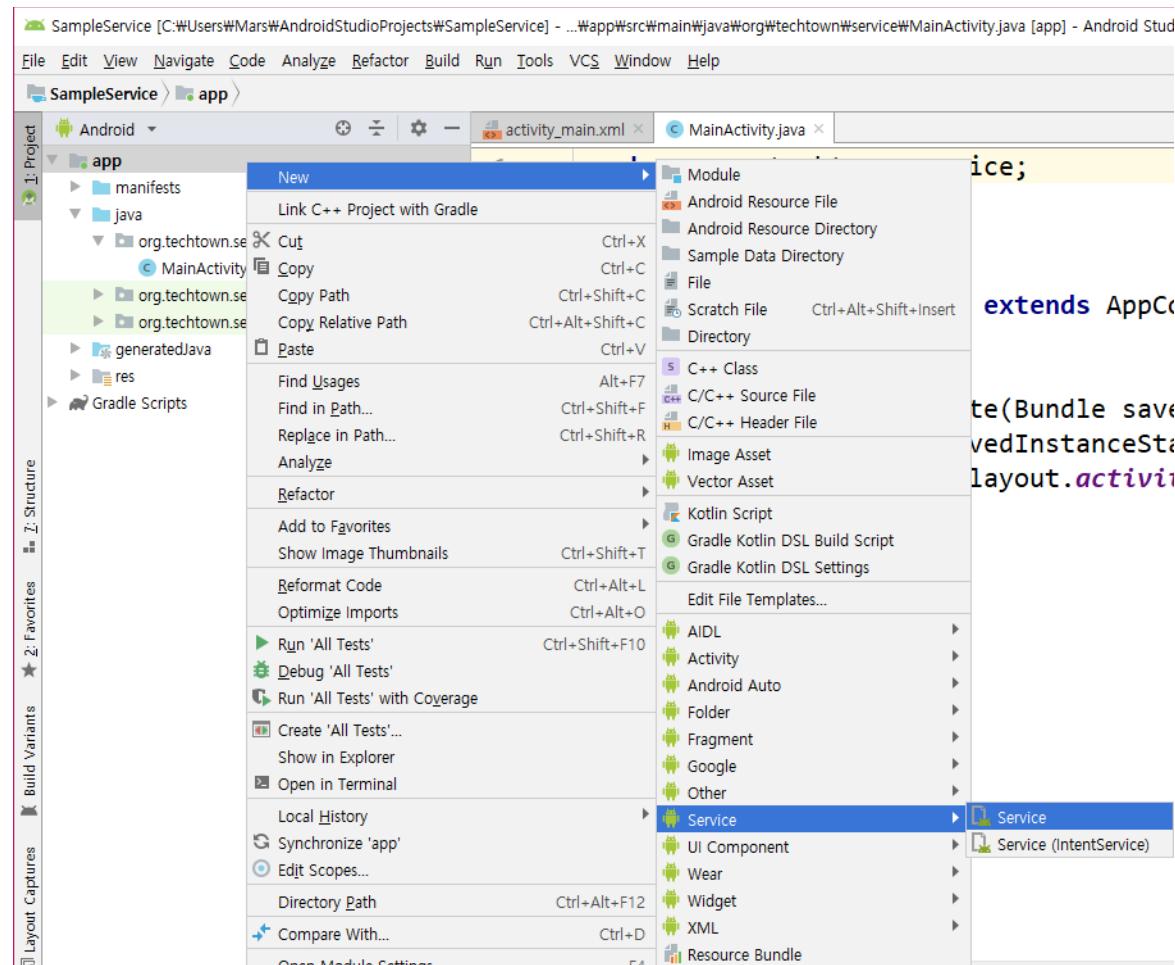
- 서비스는 화면이 없는 상태에서 백그라운드로 실행됨
- 서비스는 프로세스가 종료되어도 시스템에서 자동으로 재시작함





새로운 서비스 추가

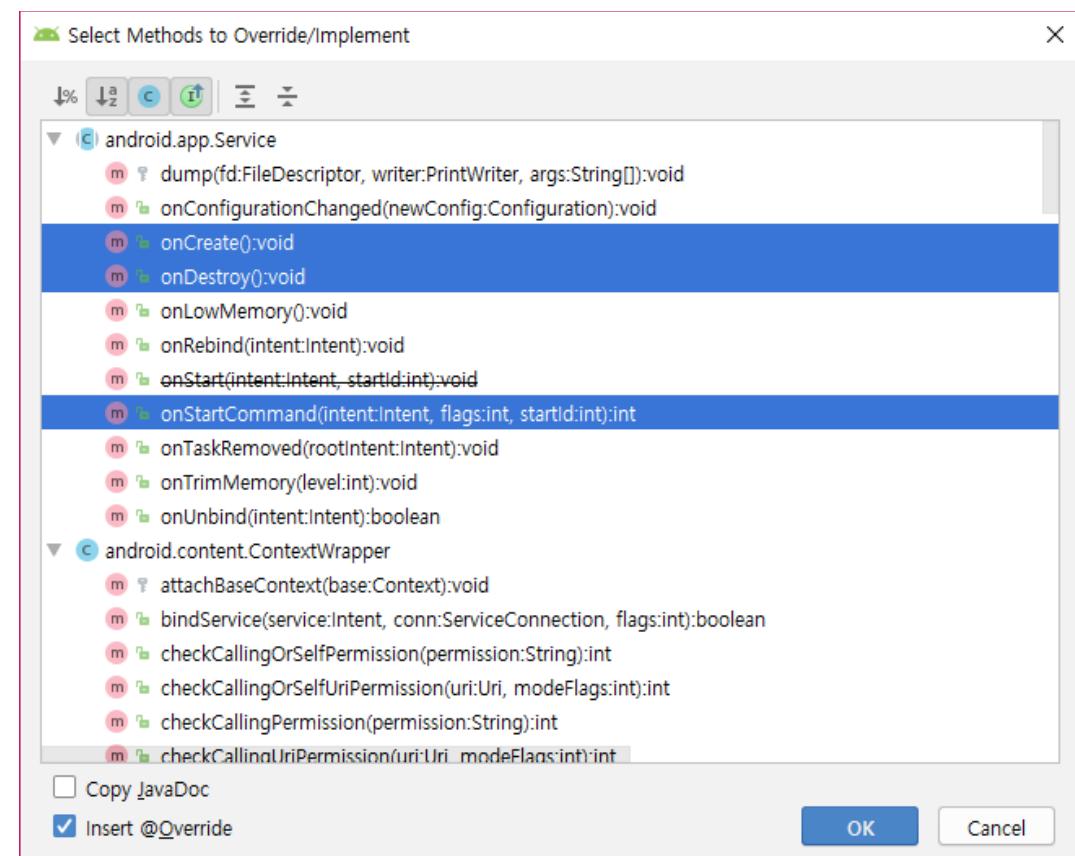
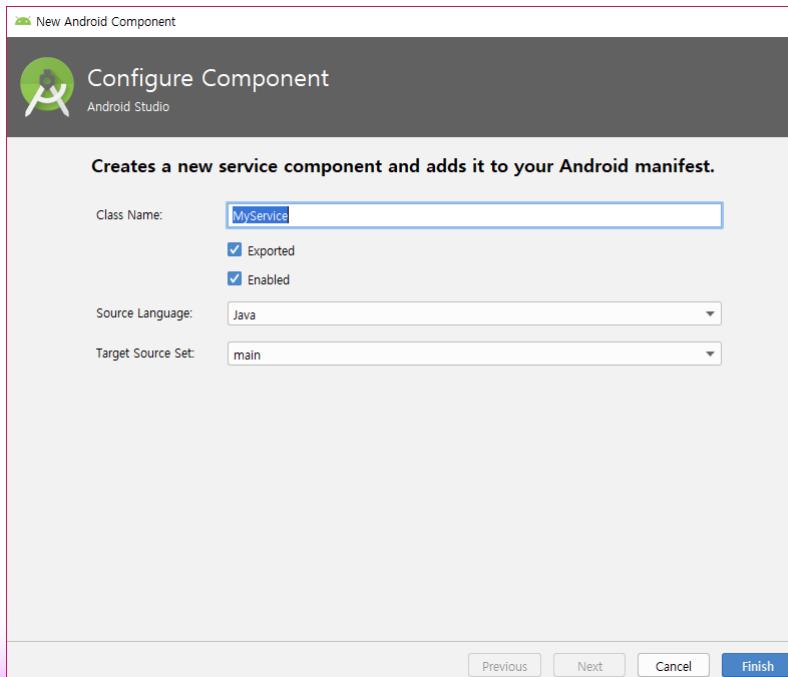
- 프로젝트 영역에서 New → Service → Service 메뉴를 이용해 서비스 추가





서비스 추가를 위한 대화상자

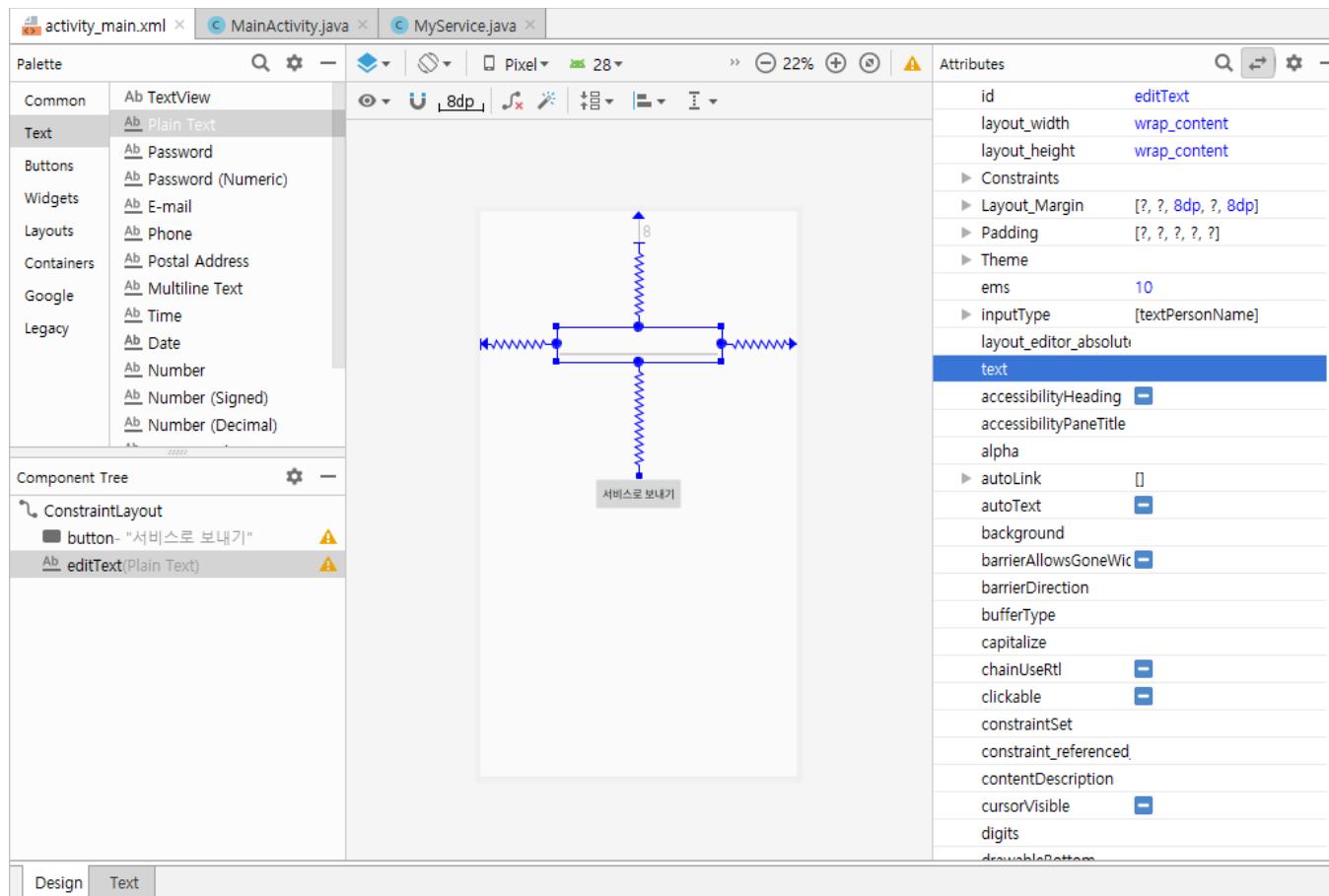
- 대화상자에서 서비스명을 입력하고 [Finish] 버튼을 누름
- 대표적인 세 개 메소드, `onCreate`, `onDestroy`, `onStartCommand` 를 재정의함





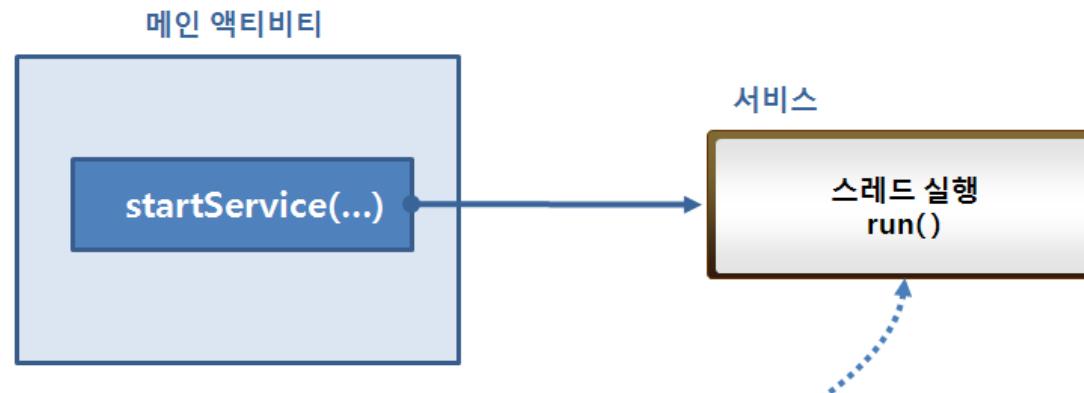
액티비티에서 서비스 시작

- 서비스 시작을 위해 `startService` 메소드를 호출
- 메인 액티비티에서 버튼 누르면 시작하도록 할 수 있음





서비스를 시작, 중지시키는 메소드



- 서비스는 **백그라운드**에서 실행되는 애플리케이션 구성 요소
- 서비스는 매니페스트 파일(AndroidManifest.xml) 안에 **<service>** 태그를 이용하여 선언
- 서비스를 시작/중지시키는 메소드
 - Context.startService()
 - Context.bindService()
 - stopService(...)
 - unbindService(...)
- 서비스는 다른 구성 요소들처럼 메인 스레드에서 동작

따라서 CPU를 많이 쓰거나 대기 상태(blocking)를 필요로 하는 작업들은 **스레드를 새로** 만들어 주어야 함



서비스 클래스 정의

```
public class MyService extends Service {  
    ...  
    @Override  
    public void onCreate() {  
        super.onCreate();  
        Log.d(TAG, "onCreate() 호출됨.");  
    }  
  
    @Override  
    public int onStartCommand(Intent intent, int flags, int startId) {  
        Log.d(TAG, "onStartCommand() 호출됨.");  
  
        if (intent == null) {  
            return Service.START_STICKY;  
        } else {  
            processCommand(intent);  
        }  
  
        return super.onStartCommand(intent, flags, startId);  
    }  
    ...
```

Continued..



서비스 클래스 정의

...

```
private void processCommand(Intent intent) {  
    String command = intent.getStringExtra("command");  
    String name = intent.getStringExtra("name");  
    Log.d(TAG, "command : " + command + ", name : " + name);  
  
    for (int i = 0; i < 5; i++) {  
        try {  
            Thread.sleep(1000);  
        } catch(Exception e) {};  
  
        Log.d(TAG, "Waiting " + i + " seconds.");  
    }  
}
```

...

Continued..



메인 액티비티 코드 만들기

참조파일 SampleService>/app/java/org.techtown.service/MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    EditText editText;

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

        editText = findViewById(R.id.editText);

        Button button = findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String name = editText.getText().toString();

                Intent intent = new Intent(getApplicationContext(), MyService.class); } ① 인텐트 객체 만들고
                intent.putExtra("command", "show");
                intent.putExtra("name", name); } ② 부가 데이터 넣기
                startService(intent);—→ ② 서비스 시작하기
            }
        });
    }
}
```



서비스는 매니페스트에 자동 추가됨

```
...  
<application ... >  
...  
<service android:name=".MyService" >  
</service>  
  
</application>  
...
```

1

서비스 등록



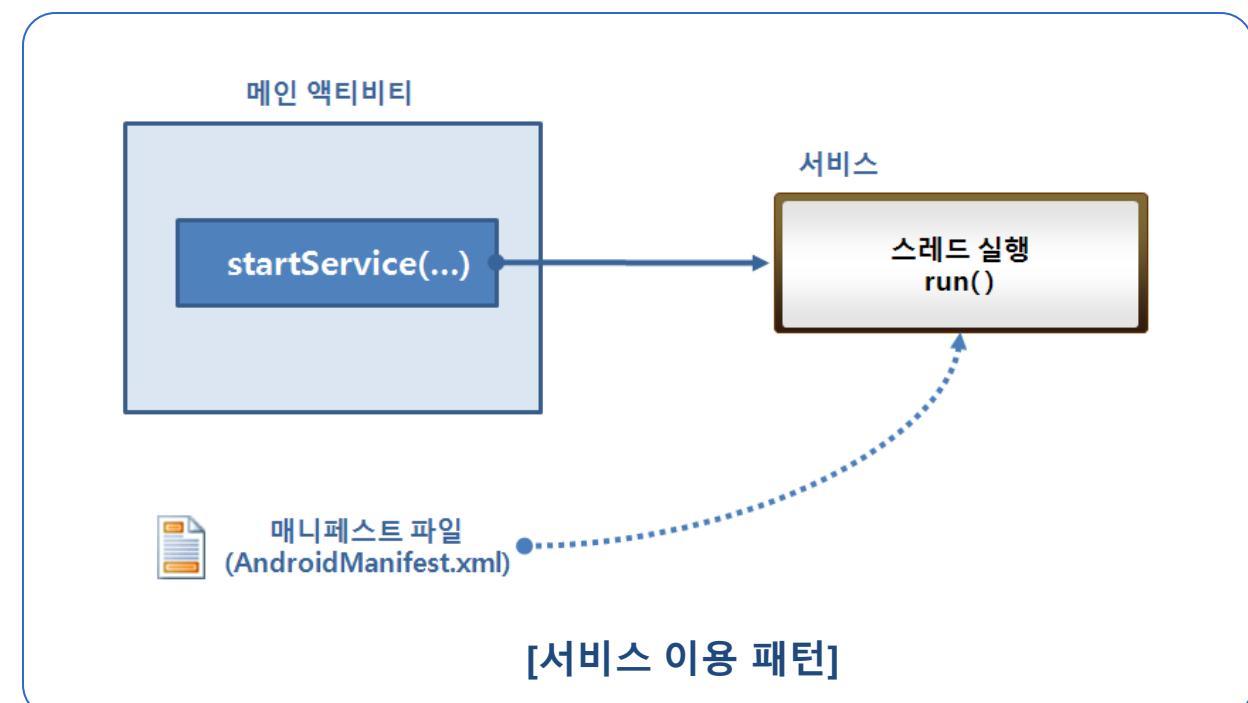
서비스 실행하여 로그 확인

Logcat

Emulator Nexus_5X_API_28 Andi org.techtown.service (9847) Verbose Regex Unnamed-0

```
2018-11-29 19:48:23.305 9847-9847/org.techtown.service D/MyService: onCreate() 호출됨.  
2018-11-29 19:48:23.306 9847-9847/org.techtown.service D/MyService: onStartCommand() 호출됨.  
2018-11-29 19:48:23.309 9847-9847/org.techtown.service D/MyService: command : show, name : mike  
2018-11-29 19:48:24.311 9847-9847/org.techtown.service D/MyService: Waiting 0 seconds.  
2018-11-29 19:48:25.313 9847-9847/org.techtown.service D/MyService: Waiting 1 seconds.  
2018-11-29 19:48:26.314 9847-9847/org.techtown.service D/MyService: Waiting 2 seconds.  
2018-11-29 19:48:27.326 9847-9847/org.techtown.service D/MyService: Waiting 3 seconds.
```

TODO Terminal Build Logcat Profiler Run Event Log





서비스에서 화면 띄우기

- 서비스에서 액티비티를 띄울 수 있음
- 플래그를 이용해 한 번 만들어진 액티비티를 그대로 띄움

```
private void processCommand(Intent intent) {  
    ...  
  
    Intent showIntent = new Intent(getApplicationContext(), MainActivity.class);  
    showIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK |  
        Intent.FLAG_ACTIVITY_SINGLE_TOP |  
        Intent.FLAG_ACTIVITY_CLEAR_TOP);  
    showIntent.putExtra("command", "show");  
    showIntent.putExtra("name", name + " from service.");  
    startActivity(showIntent);  
}
```



액티비티에서 인텐트 받아 처리하기

- 액티비티가 이미 메모리에 만들어져 있는 경우 `onNewIntent` 메소드 호출됨

```
@Override  
protected void onNewIntent(Intent intent) {  
    processIntent(intent);  
    super.onNewIntent(intent);  
}  
  
private void processIntent(Intent intent) {  
    if (intent != null) {  
        String command = intent.getStringExtra("command");  
        String name = intent.getStringExtra("name");  
  
        Toast.makeText(this, "command : " + command + ", name : " + name,  
                      Toast.LENGTH_LONG).show();  
    }  
}
```



서비스에서 띄운 화면



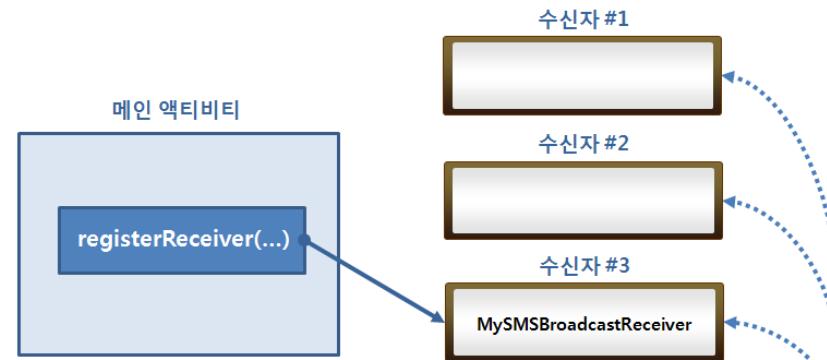


2.

브로드캐스트 수신자 이해하기



브로드캐스트 수신자



- 애플리케이션이 글로벌 이벤트(global event)를 받아서 처리하려면 브로드캐스트 수신자로 등록
- 글로벌 이벤트란 “전화가 왔습니다.”, “문자 메시지가 도착했습니다.”와 같이 안드로이드 시스템 전체에 보내지는 이벤트
- 브로드캐스트 수신자는 인텐트필터를 포함하며, 매니페스트 파일에 등록함으로써 인텐트를 받을 준비를 함
- 수신자가 매니페스트 파일에 등록되었다면 따로 시작시키지 않아도 됨
- 애플리케이션은 컨텍스트 클래스의 registerReceiver 메소드를 이용하면 런타임 시에도 수신자를 등록할 수 있음
- 서비스처럼 브로드캐스트 수신자도 UI가 없음



브로드캐스트의 구분

• 인텐트와 브로드캐스트

- 인텐트를 이용해서 액티비티를 실행하면 포그라운드(foreground)로 실행되어 사용자에게 보여지지만
- 브로드캐스트를 이용해서 처리하면 백그라운드(background)로 동작하므로 사용자가 모름
- 인텐트를 받으면 onReceive() 메소드가 자동으로 호출됨

• 브로드캐스트의 구분

브로드캐스트는 크게 두 가지 클래스로 구분됨

- 일반 브로드캐스트 (sendBroadcast() 메소드로 호출)

비동기적으로 실행되며 모든 수신자는 순서없이 실행됨 (때로는 동시에 실행됨)

효율적이나, 한 수신자의 처리 결과를 다른 수신자가 이용할 수 없고 중간에 취소불가

- 순차 브로드캐스트 (sendOrderedBroadcast() 메소드로 호출)

한 번에 하나의 수신자에만 전달되므로 순서대로 실행됨. 중간에 취소하면 그 다음

수신자는 받지 못함. 수신자가 실행되는 순서는 인텐트필터의 속성으로 정할 수 있음

순서가 같으면 임의로 실행됨.



브로드캐스트 수신자 예제

브로드캐스트 수신자 예제

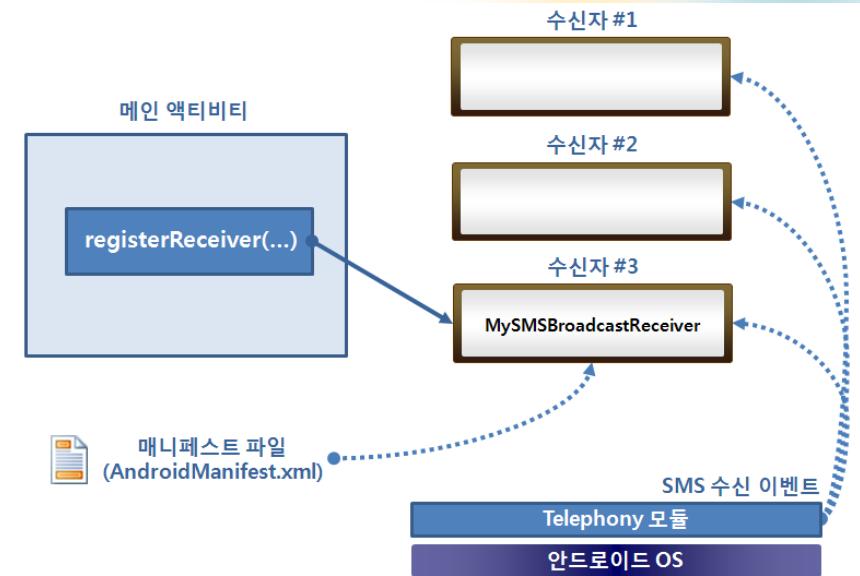
- 브로드캐스트 수신자로 SMS 수신 확인하기
- 브로드캐스트 수신자 정의

브로드캐스트 수신자 정의

- 일정 시간간격으로 메시지를 보여 주는 서비스 클래스 정의

매니페스트에 추가

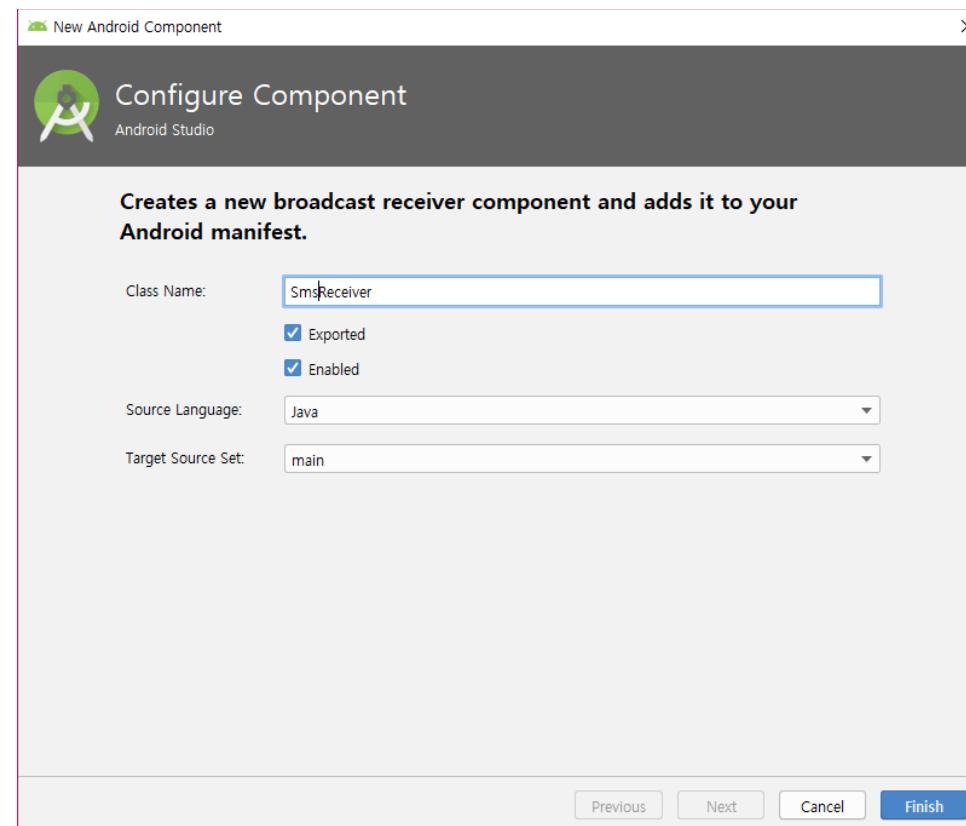
- 343 -





새로운 브로드캐스트 수신자 추가

- SampleReceiver 프로젝트 생성
- New → Other → Broadcast Receiver 메뉴로 브로드캐스트 수신자 추가





매니페스트에 추가

```
<application  
    ...  
  
    <receiver  
        android:name=".SmsReceiver"  
        android:enabled="true"  
        android:exported="true">  
        <intent-filter>  
            <action android:name="android.provider.Telephony.SMS_RECEIVED"/>  
        </intent-filter>  
    </receiver>  
</application>
```



브로드캐스트 수신자 클래스 정의

```
public class SmsReceiver extends BroadcastReceiver {
    public static final String TAG = "SmsReceiver";

    @Override
    public void onReceive(Context context, Intent intent) {
        Log.i(TAG, "onReceive() 메소드 호출됨.");

        Bundle bundle = intent.getExtras();
        SmsMessage[] messages = parseSmsMessage(bundle);
        if (messages != null && messages.length > 0) {
            String sender = messages[0].getOriginatingAddress();
            Log.i(TAG, "SMS sender : " + sender);
            String contents = messages[0].getMessageBody().toString();
            Log.i(TAG, "SMS contents : " + contents);
            Date receivedDate = new Date(messages[0].getTimestampMillis());
            Log.i(TAG, "SMS received date : " + receivedDate.toString());
        }
    }
    ...
}
```



브로드캐스트 수신자 클래스 정의

...

```
private SmsMessage[] parseSmsMessage(Bundle bundle) {
    Object[] objs = (Object[]) bundle.get("pdus");
    SmsMessage[] messages = new SmsMessage[objs.length];

    int smsCount = objs.length;
    for (int i = 0; i < smsCount; i++) {
        // PDU 포맷으로 되어 있는 메시지를 복원합니다.
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) { // API 23 이상
            String format = bundle.getString("format");
            messages[i] = SmsMessage.createFromPdu((byte[]) objs[i], format);
        } else {
            messages[i] = SmsMessage.createFromPdu((byte[]) objs[i]);
        }
    }

    return messages;
}
```

...



권한 추가

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="org.androidtown.samplereceiver">

    <uses-permission android:name="android.permission.RECEIVE_SMS" />

    ...

```



Sync Now 버튼 클릭

The screenshot shows the Android Studio interface with the build.gradle file open. The status bar at the top displays a yellow message: "Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly." To the right of this message is a "Sync Now" button. The code in the build.gradle file is as follows:

```
20
21 dependencies {
22     implementation fileTree(dir: 'libs', include: ['*.jar'])
23     implementation 'com.android.support:appcompat-v7:28.0.0'
24     implementation 'com.android.support.constraint:constraint-layout:1.1.3'
25     testImplementation 'junit:junit:4.12'
26     androidTestImplementation 'com.android.support.test:runner:1.0.2'
27     androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
28
29     implementation 'com.github.pedroSG94:AutoPermissions:1.0.3'
30 }
31
```



위험 권한 부여 코드 추가

참조파일 SampleReceiver>/app/java/org.techtown.receiver/MainActivity.java

```
public class MainActivity extends AppCompatActivity
    implements AutoPermissionsListener { ] ① MainActivity가 인터페이스
    구현하도록 하기

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

        AutoPermissions.Companion.loadAllPermissions(this, 101); → ② 모든 위험 권한을 자동 부여하도록
    }                                         하는 메서드 호출하기

    @Override public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        AutoPermissions.Companion.parsePermissions(this, requestCode, permissions, this);
    }

    @Override
    public void onDenied(int requestCode, @NotNull String[] permissions) {
        Toast.makeText(this, "permissions denied : " + permissions.length,
            Toast.LENGTH_LONG).show();
    }

    @Override public void onGranted(int requestCode, @NotNull String[] permissions) {
        Toast.makeText(this, "permissions granted : " + permissions.length,
            Toast.LENGTH_LONG).show();
    }
}
```



브로드캐스트 수신자 실행 화면

The screenshot illustrates the execution environment for a Broadcast Receiver. On the left, a permission dialog from the app 'SampleReceiver' asks for permission to send and view SMS messages. The 'ALLOW' button is highlighted. In the center, the 'Extended controls' window is open, showing a simulated SMS message being sent from '(650) 555-1212' with the content 'Hello!'. On the right, the Logcat window displays the following log entries:

```
2018-11-29 20:37:16.568 11311-11311/org.techtown.receiver I/SmsReceiver: onReceive() 메소드 호출됨.  
2018-11-29 20:37:16.617 11311-11311/org.techtown.receiver I/SmsReceiver: SMS sender : 6505551212  
2018-11-29 20:37:16.617 11311-11311/org.techtown.receiver I/SmsReceiver: SMS contents : Hello!  
2018-11-29 20:37:16.638 11311-11311/org.techtown.receiver I/SmsReceiver: SMS received date : Thu Nov 29 20:3
```

[Extended Controls에서 에뮬레이터로 SMS를 보내고 로그가 출력된 화면]



SMS 문자를 보여줄 새로운 화면 추가

New Android Activity

Configure Activity

Creates a new empty activity

Activity Name: SmsActivity

Generate Layout File (checked)

Layout Name: activity_sms

Launcher Activity (unchecked)

Backwards Compatibility (AppCompat) (checked)

Package name: org.techtown.receiver

Source Language: Java

Target Source Set: main

The name of the activity class to create

Previous Next Cancel

Palette

Common: TextView, Button, ImageView, RecyclerView, <fragment>, ScrollView, Switch

Text, Buttons, Widgets, Layouts, Containers, Google, Legacy

Component Tree

- ConstraintLayout
 - editText (Plain Text)
 - editText2 (Plain Text)
 - editText3 (Plain Text)
 - button - "확인"

activity_main.xml MainActivity.java SmsReceiver.java activity_sms.xml SmsActivity.java AndroidManifest.xml app

Attributes

- filterTouchesWhenOb []
- firstBaselineToTopHei []
- fitsSystemWindows []
- focusable []
- focusableInTouchMod []
- foreground []
- foregroundGravity []
- freezesText []
- hapticFeedbackEnable []
- height []
- imeActionId []
- imeActionLabel []
- imeOptions []
- includeFontPadding []
- inputMethod []
- isScrollContainer []
- keepScreenOn []
- lastBaselineToBottom []
- layerType []
- layout_constrainedHe []
- layout_constrainedWi []
- layout_editor_absolut []
- layout_goneMarginBc []
- layout_goneMarginEn []
- layout_goneMarginLe []
- layout_goneMarginRi []
- layout_goneMarginSt []
- layout_goneMarginTo []

Design Text



SMS 문자를 보여줄 새로운 화면 추가

```
...
@Override
protected void onNewIntent(Intent intent) {
    processIntent(intent);

    super.onNewIntent(intent);
}

private void processIntent(Intent intent) {
    if (intent != null) {
        String sender = intent.getStringExtra("sender");
        String contents = intent.getStringExtra("contents");
        String receivedDate = intent.getStringExtra("receivedDate");

        editText.setText(sender);
        editText2.setText(contents);
        editText3.setText(receivedDate);
    }
}
...
}
```



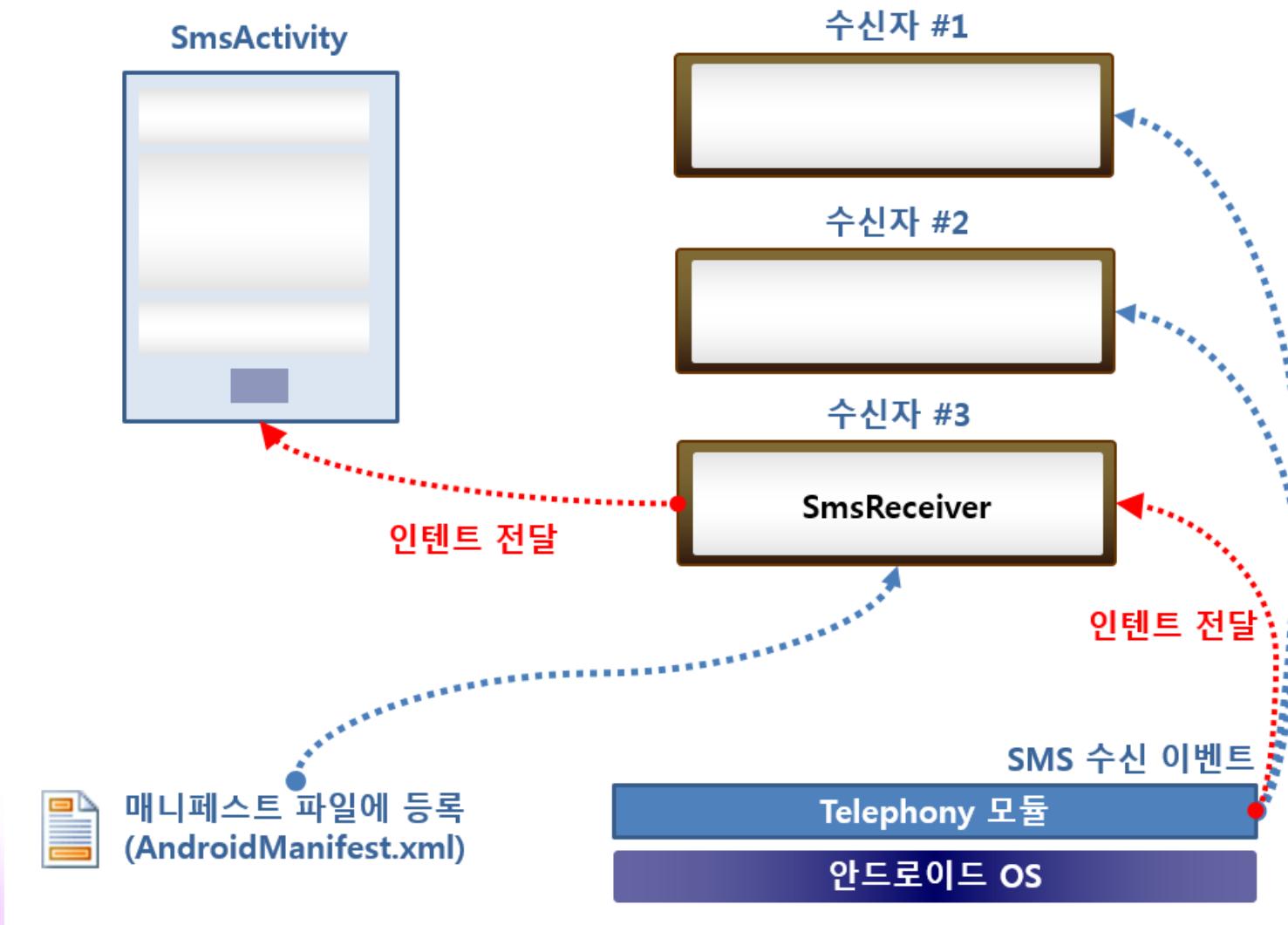
서비스에서 화면 띄우기

```
...  
private void sendToActivity(Context context, String sender, String contents, Date receivedDate) {  
    // 메시지를 보여줄 액티비티를 띄워줍니다.  
    Intent myIntent = new Intent(context, SmsActivity.class);  
  
    // 플래그를 이용합니다.  
    myIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK  
        | Intent.FLAG_ACTIVITY_SINGLE_TOP | Intent.FLAG_ACTIVITY_CLEAR_TOP);  
  
    myIntent.putExtra("sender", sender);  
    myIntent.putExtra("contents", contents);  
    myIntent.putExtra("receivedDate", format.format(receivedDate));  
  
    context.startActivity(myIntent);  
}  
...
```





브로드캐스트 수신자 동작 방식

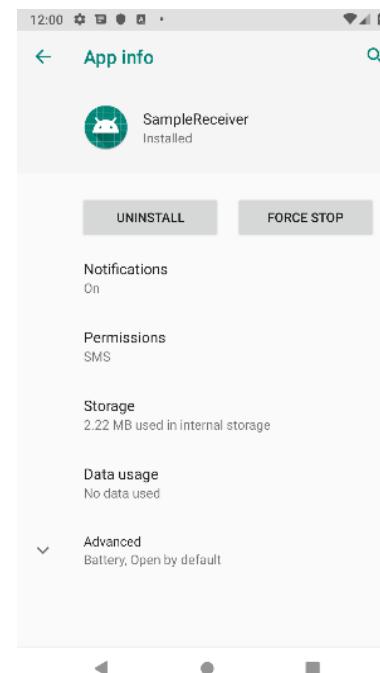




화면을 수정해도 이전 화면이 계속 보여요!

- **프로젝트를 복사하여 새로운 프로젝트를 만든 경우**

- SMS를 수신하는 동일한 앱이 설치되어 있다면 SMS 수신 시 그 화면이 보여질 수 있음
- 따라서, 이전에 설치한 앱을 삭제한 후 새로운 앱을 실행해야 함
- 단말의 설정에서 앱을 삭제할 수 있음





3.

위험 권한 부여하기



일반 권한과 위험 권한 (마시멜로 API23부터)

- 위험 권한은 실행 시 권한 부여





대표적인 위험 권한들



LOCATION (위치)

- ACCESS_FINE_LOCATION
- ACCESS_COARSE_LOCATION

CAMERA

- CAMERA

MICROPHONE

- RECORD_AUDIO

CONTACTS

- READ_CONTACTS

- WRITE_CONTACTS

- GET_ACCOUNTS

PHONE

- READ_PHONE_STATE

- CALL_PHONE

- READ_CALL_LOG

- WRITE_CALL_LOG

- ADD_VOICEMAIL

- USE_SIP

- PROCESS_OUTGOING_CALLS

SMS

- SEND_SMS

- RECEIVE_SMS

- READ_SMS

- RECEIVE_WAP_PUSH

- RECEIVE_MMS

CALENDAR

- READ_CALENDAR

- WRITE_CALENDAR

SENSORS

- BODY_SENSORS

STORAGE

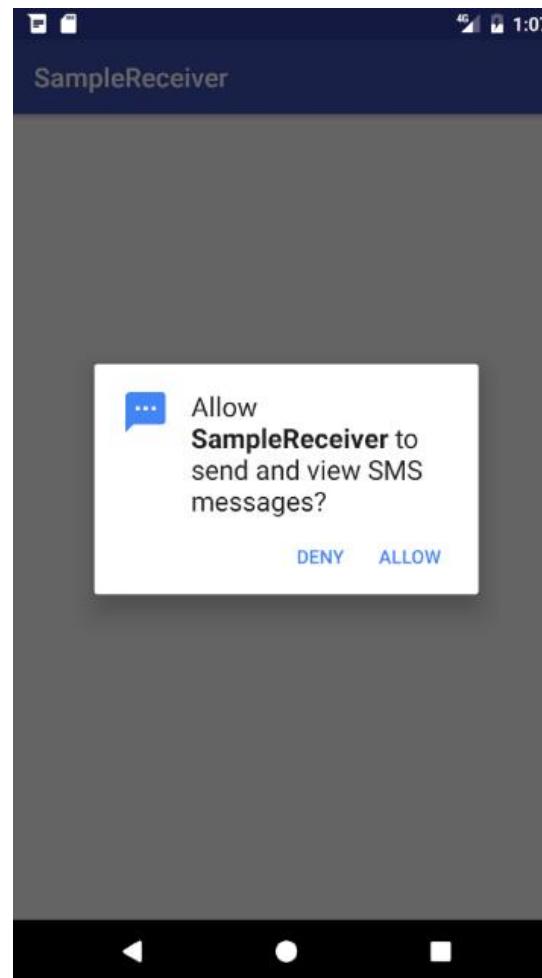
- READ_EXTERNAL_STORAGE

- WRITE_EXTERNAL_STORAGE



실행 시 권한 부여

- 실행 시 권한 부여를 묻는 대화상자 표시





매니페스트에 권한 추가

참조파일 SamplePermission>/app/manifests/AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="org.techtown.permission">

    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

```
<application>
```

중략...



위험 권한 부여 요청 코드 추가

```
public void checkPermissions(String[] permissions) {  
    ArrayList<String> targetList = new ArrayList<String>();  
  
    for (int i = 0; i < permissions.length; i++) {  
        String curPermission = permissions[i];  
        int permissionCheck = ContextCompat.checkSelfPermission(this, curPermission);  
        if (permissionCheck == PackageManager.PERMISSION_GRANTED) {  
            Toast.makeText(this, curPermission + " 권한 있음.", Toast.LENGTH_LONG).show();  
        } else {  
            Toast.makeText(this, curPermission + " 권한 없음.", Toast.LENGTH_LONG).show();  
            if (ActivityCompat.shouldShowRequestPermissionRationale(this, curPermission)) {  
                Toast.makeText(this, curPermission + " 권한 설명 필요함.",  
                    Toast.LENGTH_LONG).show();  
            } else {  
                targetList.add(curPermission);  
            }  
        }  
    }  
  
    String[] targets = new String[targetList.size()];  
    targetList.toArray(targets);  
  
    ActivityCompat.requestPermissions(this, targets, 101); —→ ② 위험 권한 부여 요청하기
```



권한 요청 결과 확인 코드

```
@Override
public void onRequestPermissionsResult(int requestCode, String permissions[],
                                         int[] grantResults) {
    switch (requestCode) { ↗①
        case 1: {
            if (grantResults.length > 0 &&
                grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                Toast.makeText(this, "SMS 권한을 사용자가 승인함.",
                               Toast.LENGTH_LONG).show();
            } else {
                Toast.makeText(this, "SMS 권한 거부됨.", Toast.LENGTH_LONG).show();
            }
        }
    }
}
```





위험 권한 자동 부여

참조파일 SamplePermission2>Gradle Scripts>build.gradle(Module:app)

중략...

```
allprojects {  
    repositories {  
        maven { url 'https://jitpack.io' }  
    }  
}  
  
dependencies {  
  
    중략...  
  
    implementation 'com.github.pedroSG94:AutoPermissions:1.0.3'  
}
```



위험 권한 자동 부여 코드 추가

참조파일 SamplePermission2>/app/java/org.techtown.permission2/MainActivity.java

```
public class MainActivity extends AppCompatActivity
    implements AutoPermissionsListener {

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

        AutoPermissions.Companion.loadAllPermissions(this, 101); —→①
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        AutoPermissions.Companion.parsePermissions(this, requestCode, permissions, this);
    }

    @Override
    public void onDenied(int requestCode, @NotNull String[] permissions) {
        Toast.makeText(this, "permissions denied : " + permissions.length,
            Toast.LENGTH_LONG).show();
    }

    @Override
    public void onGranted(int requestCode, @NotNull String[] permissions) {
        Toast.makeText(this, "permissions granted : " + permissions.length,
            Toast.LENGTH_LONG).show();
    }
}
```

②



4.

리소스와 매니페스트 이해하기



매니페스트의 태그 항목

[Reference]

```
<action><permission>
<activity><permission-group>
<activity-alias>
<application><provider>
<category><receiver>
<data><service>
<grant_uri_permission><uses_configuration>
<instrumentation><uses-library>
<intent-filter><uses-permission>
<manifest><uses-sdk>
<meta-data>
```



매니페스트 파일의 역할

- 애플리케이션의 자바 패키지 이름 지정
- 애플리케이션 구성요소에 대한 정보 등록(액티비티, 서비스, 브로드캐스트 수신자, 내용 제공자)
- 각 구성요소를 구현하는 클래스 이름 지정
- 애플리케이션이 가져야 하는 권한에 대한 정보 등록
- 다른 애플리케이션이 접근하기 위해 필요한 권한에 대한 정보 등록
- 애플리케이션 개발 과정에서 프로파일링을 위해 필요한 instrumentation 클래스 등록
- 애플리케이션에 필요한 안드로이드 API의 레벨 정보 등록
- 애플리케이션에서 사용하는 라이브러리 리스트

[Code]

```
<manifest ... >  
<application ... >  
...  
<service android:name="org.androidtown.service.MyService" ... >  
...  
</service>  
...  
</application>  
</manifest>
```

[매니페스트 파일의 기본 구조]



메인 액티비티 정의

[Code]

```
<activity android:name="org.androidtown.basicMainActivity"  
         android:label="@string/app_name">  
  
    <intent-filter>  
  
        <action android:name="android.intent.action.MAIN" />  
  
        <category android:name="android.intent.category.LAUNCHER" />  
  
    </intent-filter>  
  
</activity>
```



리소스 사용

- 리소스를 자바 코드와 분리하는 이유는 이해하기 쉽고 유지관리가 용이하기 때문임
- 프로젝트를 처음에 만들면 **[/res]** 폴더와 **[/assets]** 폴더가 따로 분리되어 있는데
두 가지 모두 리소스라고 할 수 있으며 대부분은 **[/res]** 폴더 밑에서 관리됨

- 애셋(Asset)은 동영상이나 웹페이지와 같이 용량이 큰 데이터를 의미함
- 리소스는 빌드되어 설치파일에 추가되지만 애셋은 빌드되지 않음



스타일과 테마

- 스타일과 테마는 여러 가지 속성들을 한꺼번에 모아서 정의한 것
- 대표적인 예로는 대화상자를 들 수 있음

[Code]

```
<style name="Alert" parent="android:Theme.Dialog">  
    <item name="android:windowBackground">@drawable/alertBackground</item>  
</style>
```



5.

그래들 이해하기



애플리케이션 빌드 설정

참조파일 SamplePermission2>/Gradle Scripts/build.gradle (Project: SamplePermission2)

```
buildscript {
    repositories {
        google()
        jcenter()
    }

    dependencies {
        classpath 'com.android.tools.build:gradle:3.3.0'
    }
}

allprojects {
    repositories {
        google()
        jcenter()
    }
}

task clean(type: Delete) {
    delete rootProject.buildDir
}
```



앱 빌드 설정

참조파일 SamplePermission2>/Gradle Scripts/build.gradle (Module: app)

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "org.techtown.permission"
        minSdkVersion 15
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles
                getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
}

allprojects {
    repositories {
        maven { url 'https://jitpack.io' }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
```



모듈 포함 정보

참조파일 SamplePermission2>/Gradle Scripts/settings.gradle

```
include ':app'
```



대표적인 선택 위젯의 이해와 실습



- 1 나인패치 이미지 알아보기
- 2 새로운 뷰 만들기
- 3 레이아웃 정의하고 카드뷰 넣기
- 4 리싸이클러뷰 만들기
- 5 스피너 사용하기



예쁜 모양의 버튼을 디자인해서 레이아웃에 넣고 싶은데요?

- 나인 패치 이미지 이해하기



제공되는 뷰 말고 직접 뷰를 정의할 수도 있나요?

- 새로운 뷰 만들기



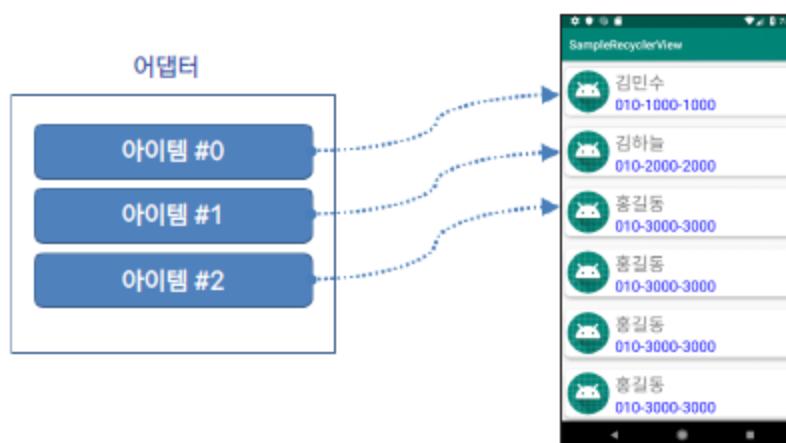
리스트 모양으로 보여주고 싶어요.

- 리사이클러뷰 사용하기



콤보박스 형태로 간단하게 보여줄 수 있나요?

- 스피너 사용하기





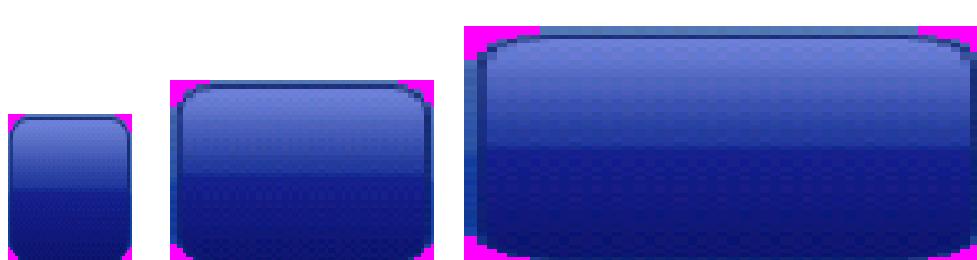
1.

나인패치 이미지 알아보기

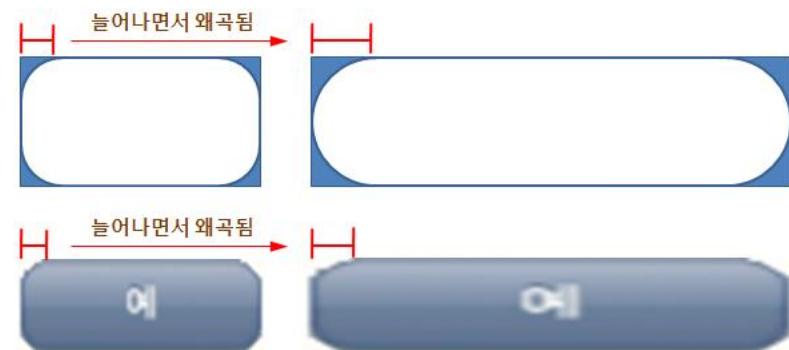


나인패치(Nine Patch) 이미지란?

- 이미지가 늘어나거나 줄어들 때 생기는 이미지 왜곡을 해결하는 방법을 정의한 것
- 서로 다른 해상도를 가진 여러 단말에 dp 단위로 뷰의 크기를 맞추다 보면 이미지 크기가 자동 조절되면서 왜곡되는 현상 발생 → 나인패치 이미지로 해결



[이미지의 크기에 따른 원본 이미지의 왜곡]



[이미지의 크기를 늘릴 때 왜곡되는 영역]



나인패치 이미지 예제

나인패치 이미지 예제

- 일반 이미지와 나인패치 이미지 비교
- 레이아웃에 여러 개의 버튼 추가

메인 액티비티의
XML 레이아웃

- 일반 버튼 추가

메인 액티비티의
XML 레이아웃

- 나인패치 버튼 추가





XML 레이아웃 조정

- 일반 이미지를 배경으로 사용하는 버튼과 나인패치 이미지를 배경으로 사용하는 버튼 추가

```
<Button  
    android:layout_width= "wrap_content"  
    android:layout_height= "wrap_content"  
    android:text= "Small"  
    android:textColor= "#ffffffff"  
    android:background= "@drawable/button_image_01"  
/>
```

1

일반 이미지를 배경으로 만든
작은 버튼 정의

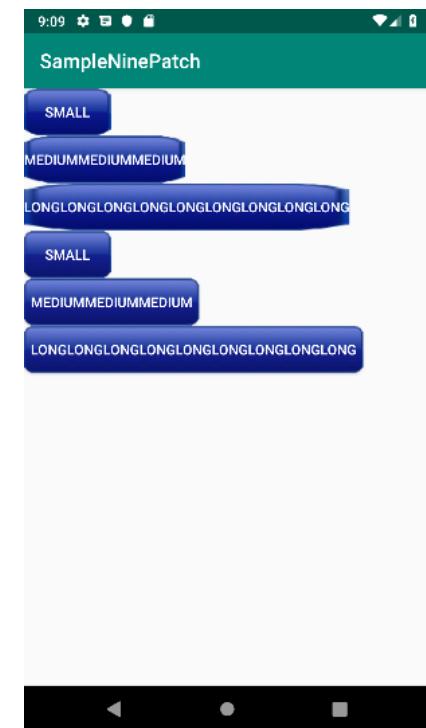


실행 화면

- 일반 이미지를 배경으로 사용하는 버튼은 모서리 부분의 이미지 깨짐 현상 발생
- 나인패치 이미지를 배경으로 사용하는 버튼은 크게 이상하게 보이지 않음

[Reference]

```
void setBackgroundColor (int color)  
void setBackgroundDrawable (Drawable d)  
void setBackgroundResource (int resid)
```



[일반 이미지와 나인패치 이미지를 적용한 버튼 모양 비교]



2.

새로운 뷰 만들기



- 새로운 뷰를 직접 만들 수 있음

- 뷰를 상속하면 새로운 뷰를 정의할 수 있음
- 뷰의 크기를 결정할 수도 있고 뷰 위에 그래픽을 그릴 수도 있음

[Reference]

```
public void onMeasure (int widthMeasureSpec, int heightMeasureSpec)  
public void onDraw(Canvas canvas)
```

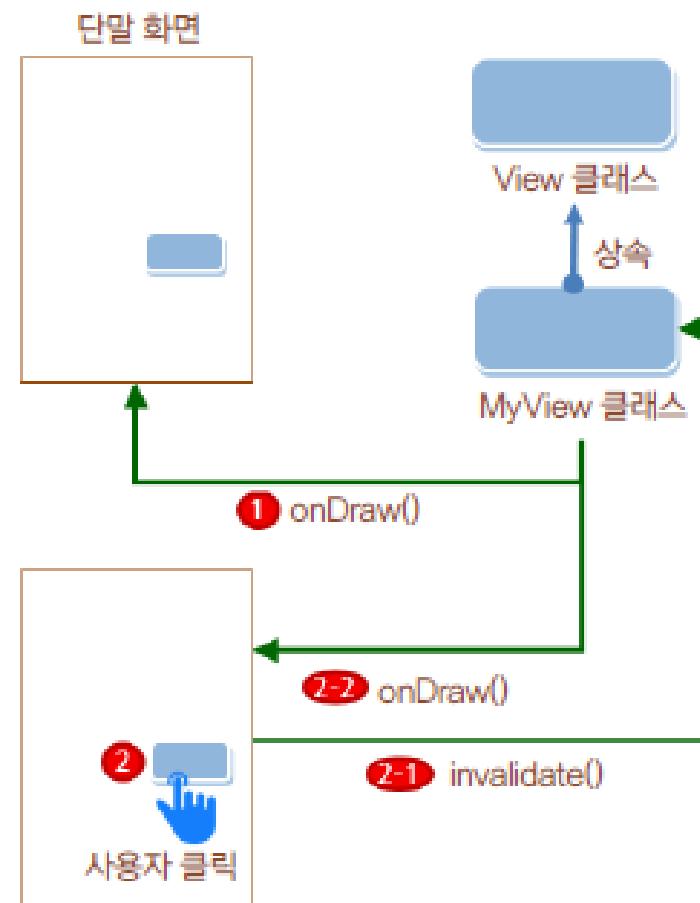
[Reference]

```
void setMeasuredDimension (int measuredWidth, int measuredHeight)
```



뷰 위에 그래픽을 그리는 과정

- 뷰에 그래픽이 그려질 때 `onDraw()` 메소드 호출됨
- 다시 그리기는 `invalidate()` 메소드 사용





버튼 만들기 예제

버튼 만들기 예제

-이미지를 보여주는 버튼

버튼을 상속한
새로운 클래스 정의

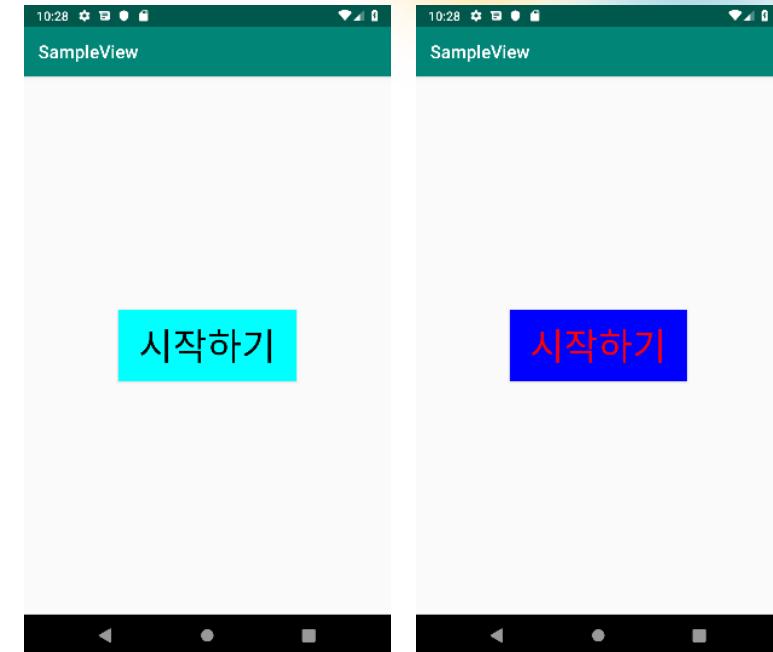
-새로운 버튼 클래스 정의

XML 레이아웃에 추가

-새로운 버튼 태그를 XML 레이
아웃에 추가

메인 액티비티 코드 작성

-메인 액티비티 코드에서 참조하여 사용





새로운 버튼 클래스 정의

- 터치 이벤트에 따라 배경 이미지를 바꾸어주는 버튼 클래스 정의

참조파일 SampleView>/java/org.techtown.view/MyButton.java

```
public class MyButton extends AppCompatButton { —→ ❶ AppCompatButton 클래스 상속하여 새로운  
클래스 정의하기  
    public MyButton(Context context) {  
        super(context);  
        init(context);  
    }  
  
    public MyButton(Context context, AttributeSet attrs) {  
        super(context, attrs);  
        init(context);  
    }  
  
    private void init(Context context) {  
        setBackgroundColor(Color.CYAN);  
        setTextColor(Color.BLACK);  
  
        float textSize = getResources().getDimension(R.dimen.text_size);  
        setTextSize(textSize);  
    }  
}
```

❷ 초기화를 위한
메서드 정의하기



새로운 버튼 클래스 정의

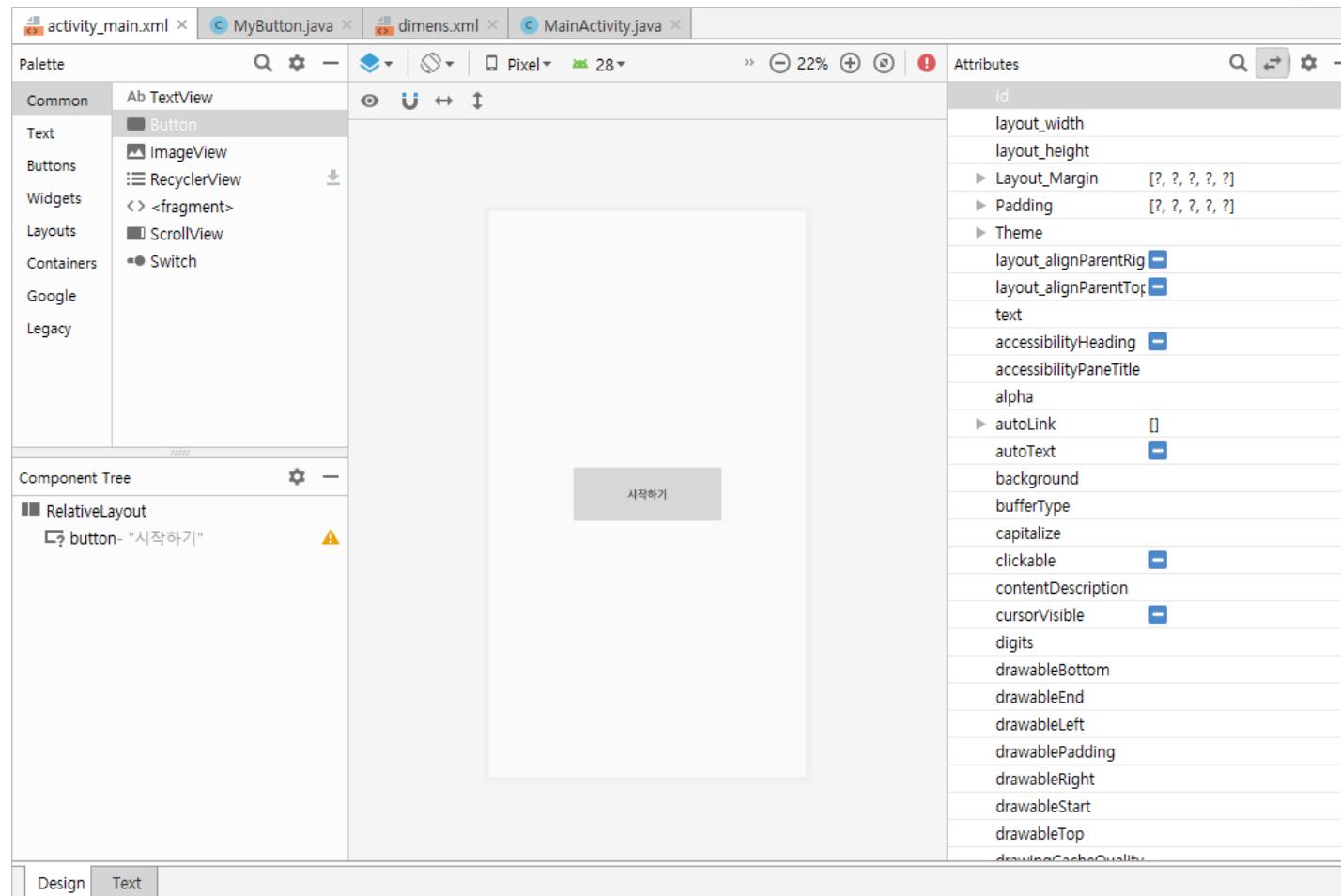
- 터치 이벤트에 따라 배경 이미지를 바꾸어 줌

```
@Override  
public boolean onTouchEvent(MotionEvent event) { } ② 뷰가 터치될 때 호출되는 함수에 기능 추가하기  
    Log.d("MyButton", "onTouchEvent 호출됨");  
  
    int action = event.getAction();  
    switch (action) {  
        case MotionEvent.ACTION_DOWN:  
            setBackgroundColor(Color.BLUE);  
            setTextColor(Color.RED);  
  
            break;  
        case MotionEvent.ACTION_OUTSIDE:  
        case MotionEvent.ACTION_CANCEL:  
        case MotionEvent.ACTION_UP:  
            setBackgroundColor(Color.CYAN);  
            setTextColor(Color.BLACK);  
  
            break;  
    }  
  
    invalidate();  
  
    return true;  
}
```

증략...



레이아웃 만들기





레이아웃 만들기

참조파일 SampleView>/app/res/layout/activity_main.xml

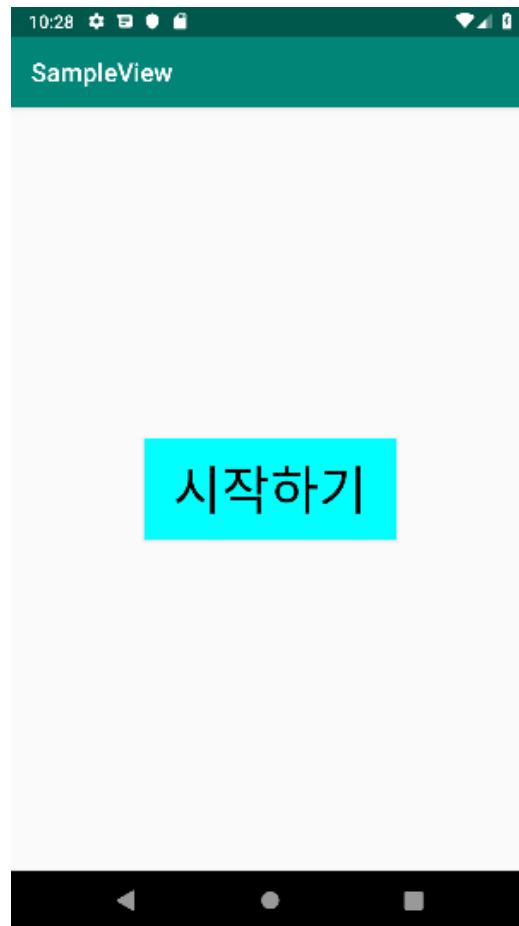
```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent" >  
  
    <org.techtown.view.MyButton  
        android:id="@+id/button"  
        android:layout_width="200dp"  
        android:layout_height="80dp"  
        android:layout_centerInParent="true"  
        android:text="시작하기"  
    />  
  
</RelativeLayout>
```

새로 만든 MyButton 클래스를 태그로 추가하기





앱 실행





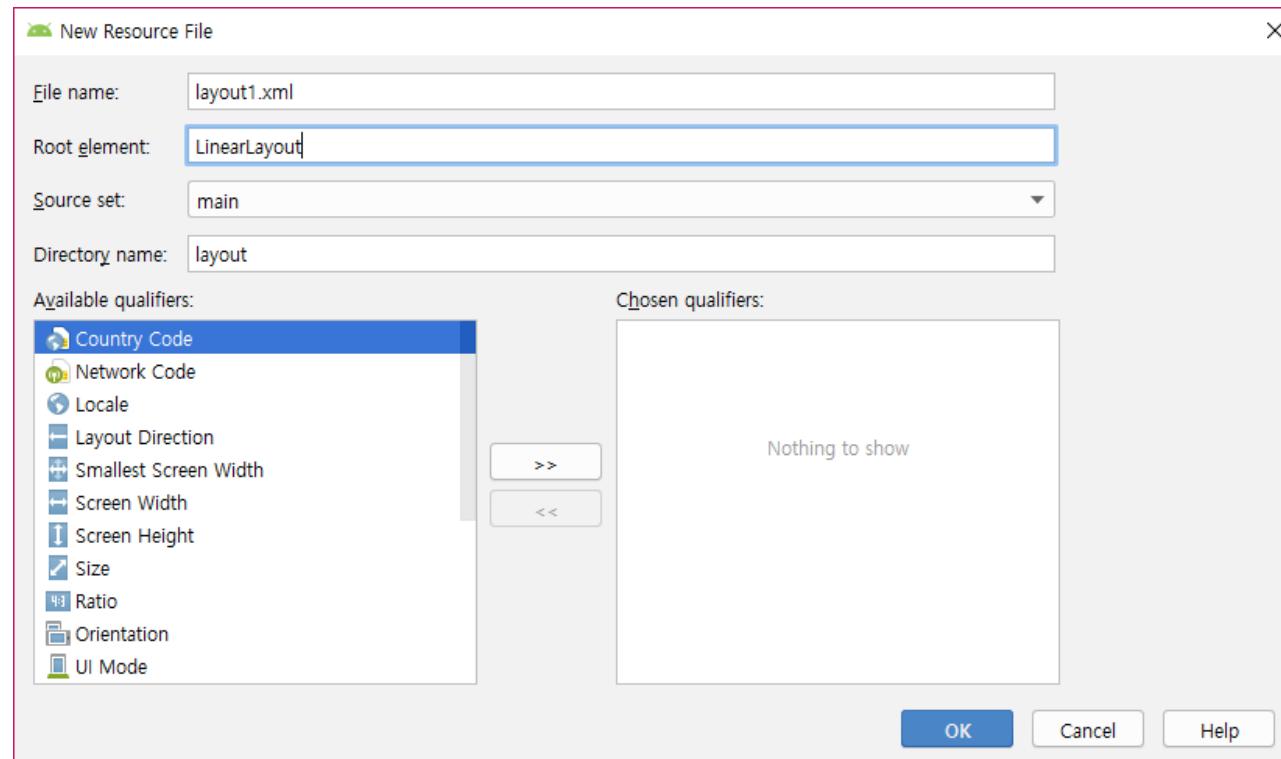
3.

레이아웃 정의하고 카드뷰 넣기



레이아웃 파일 추가

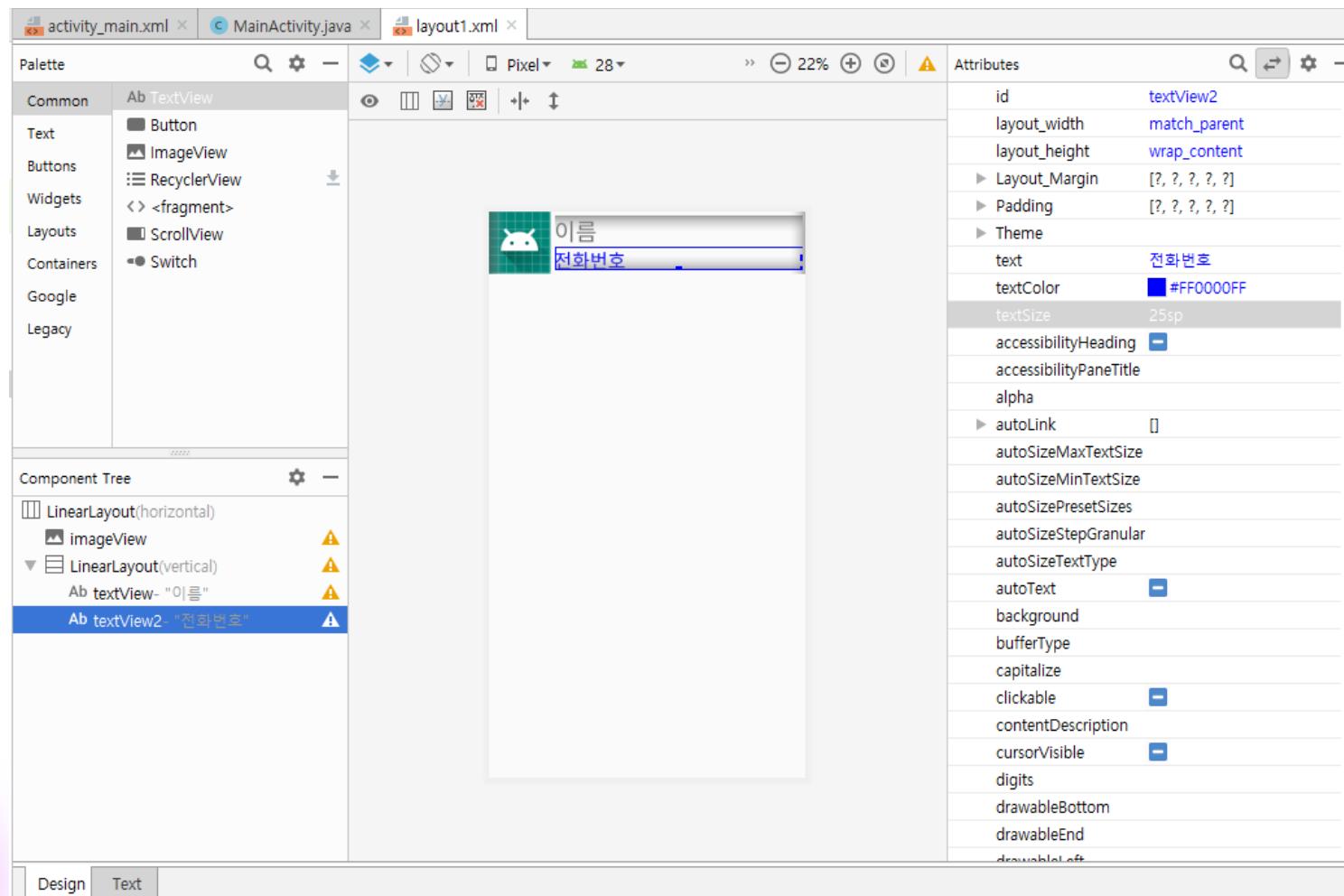
- /app/res/layout 폴더 안에 새로운 레이아웃 파일 만들기





레이아웃 파일 추가

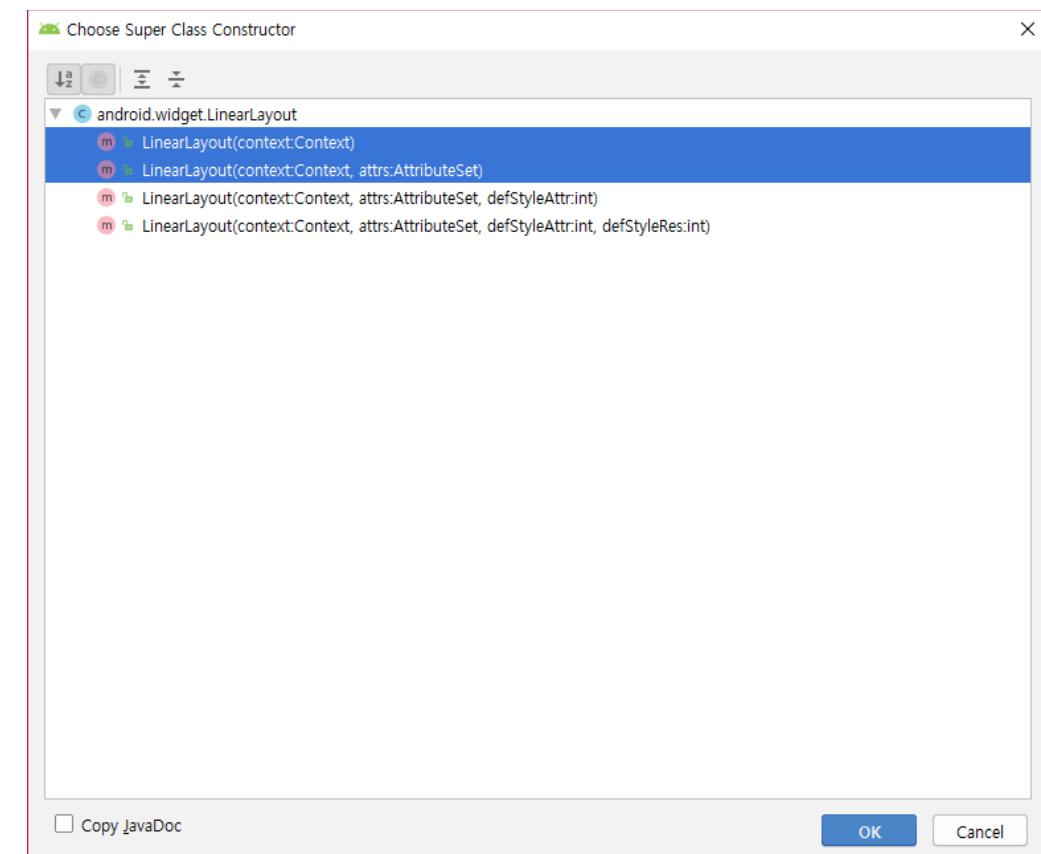
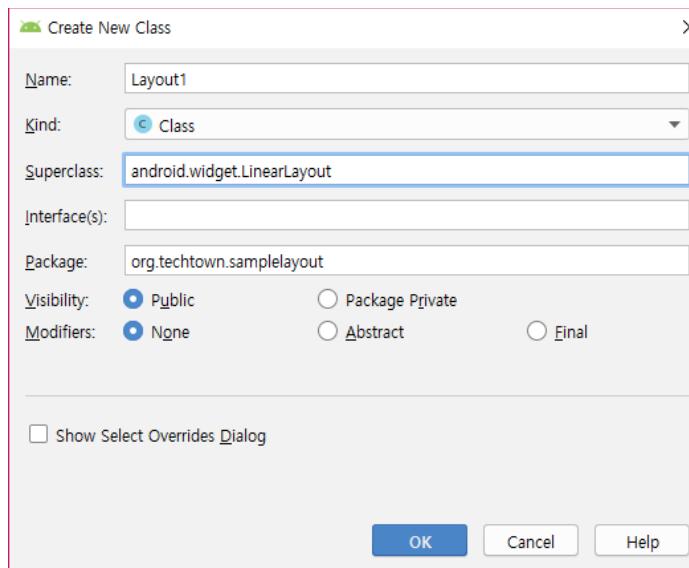
- LinearLayout 안에 하나의 이미지뷰와 텍스트뷰 두 개 추가하기





레이아웃 파일 추가

- 새로운 클래스 파일 추가하기
- 생성자는 2개 재정의





레이아웃 파일 추가

- 생성자 안에 인플레이션 진행하는 init 메서드 호출

참조파일 SampleLayout>/app/java/org.techtown.samplelayout/Layout1.java

```
public class Layout1 extends LinearLayout { → ① LinearLayout 클래스 상속하여 새로운 클래스 정의하기  
    public Layout1(Context context) {  
        super(context);  
        init(context);  
    }  
  
    public Layout1(Context context, AttributeSet attrs) {  
        super(context, attrs);  
        init(context);  
    }  
  
    private void init(Context context) {  
        LayoutInflator inflater = (LayoutInflator) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);  
        inflater.inflate(R.layout.layout1, this, true); → ② 인플레이션 진행하기  
    }  
}
```

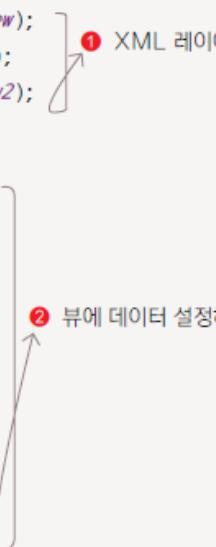


레이아웃 파일 추가

- XML 레이아웃 안에 들어있는 뷰 객체들을 찾아 변수에 할당

참조파일 SampleLayout>/app/java/org.techtown.samplelayout/Layout1.java

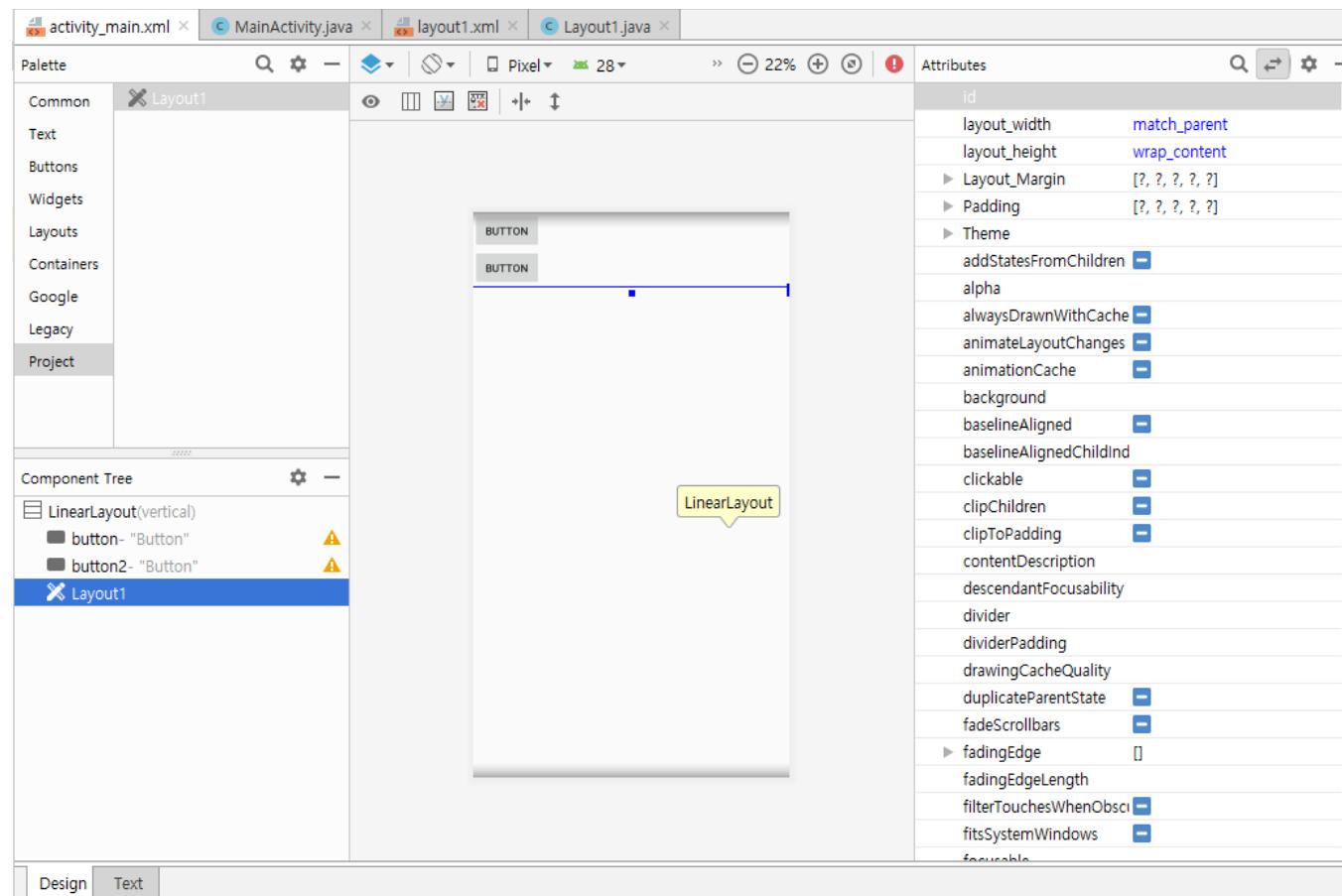
```
public class Layout1 extends LinearLayout {  
    ImageView imageView;  
    TextView textView;  
    TextView textView2;  
    중략...  
    private void init(Context context) {  
        LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);  
        inflater.inflate(R.layout.layout1, this, true);  
  
        imageView = findViewById(R.id.imageView);  
        textView = findViewById(R.id.textView);  
        textView2 = findViewById(R.id.textView2);  
    }  
  
    public void setImage(int resId) {  
        imageView.setImageResource(resId);  
    }  
  
    public void setName(String name) {  
        textView.setText(name);  
    }  
  
    public void setMobile(String mobile) {  
        textView2.setText(mobile);  
    }  
}
```





레이아웃 파일 추가

- activity_main.xml 파일에 새로 만든 뷰 추가하기





레이아웃 파일 추가

• 메인 액티비티 소스에서 데이터 설정

참조파일 SampleLayout>/app/java/org.techtown.samplelayout/MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        Layout1 layout1 = findViewById(R.id.layout1);————→ ① XML 레이아웃에 추가한 뷰 참조하기  
  
        layout1.setImageResource(R.drawable.ic_launcher_foreground);  
        layout1.setName("김민수");  
        layout1.setMobile("010-1000-1000");  
    }  
}
```



② 뷰의 메서드 호출하여 데이터 설정하기



레이아웃 파일 추가

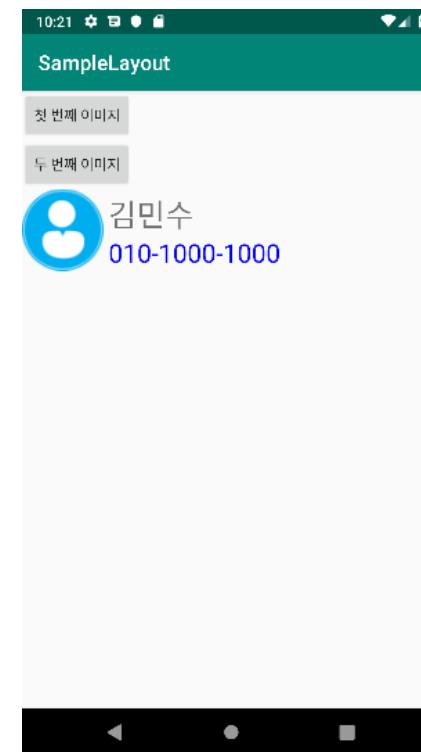
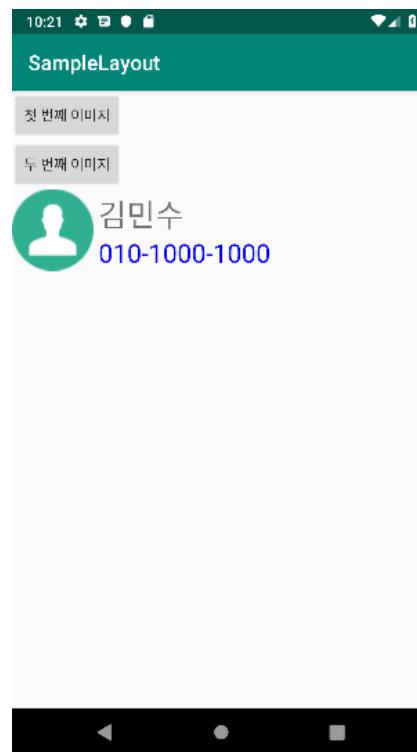
- 버튼 클릭 시의 이벤트 처리

```
Button button2 = findViewById(R.id.button2);
button2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        layout1.setImageResource(R.drawable.profile2);
    }
});
```



레이아웃 파일 추가

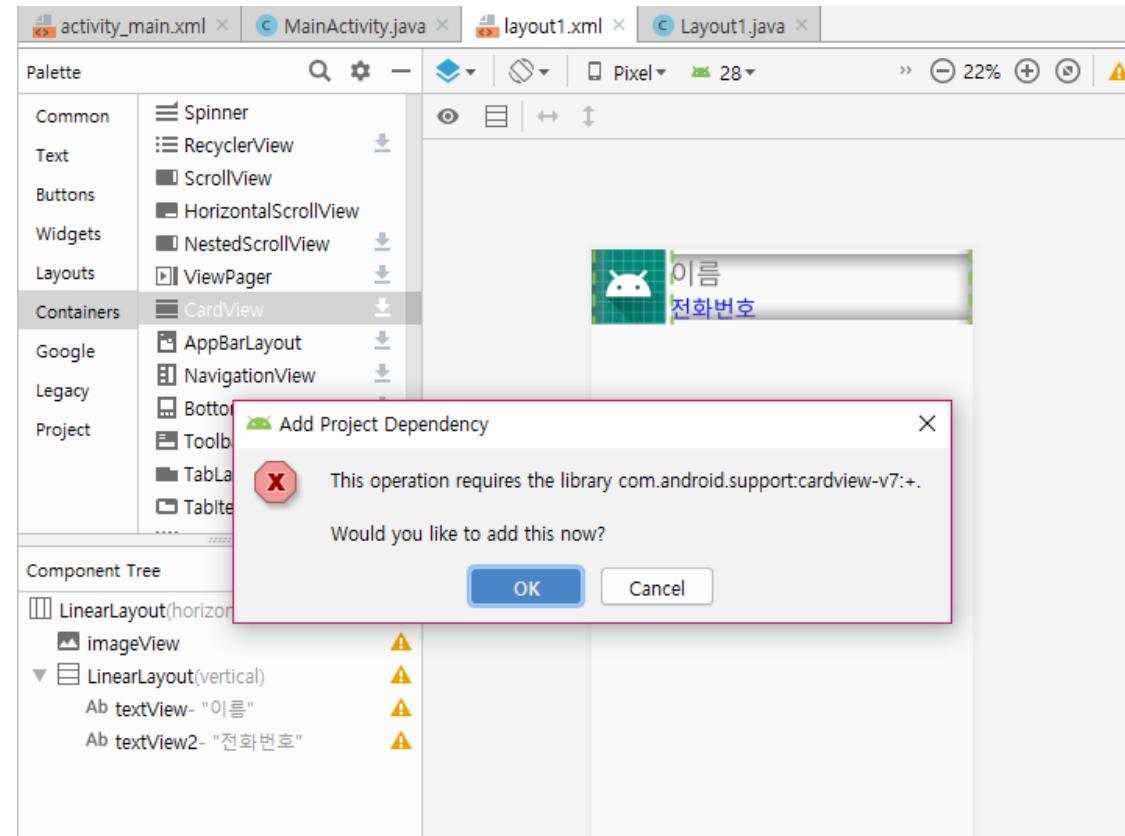
• 앱 실행





카드뷰 모양으로 바꾸기

- 카드뷰를 위한 외부 라이브러리 추가





카드뷰 모양으로 바꾸기

• 카드뷰 태그 추가

참조파일 SampleLayout>/app/res/layout/layout1.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="wrap_content"
    android:orientation="vertical">

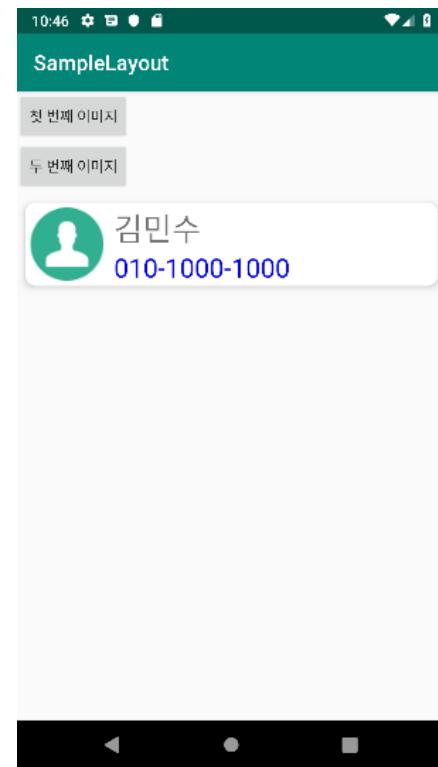
    <android.support.v7.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:cardBackgroundColor="#FFFFFF"
        app:cardCornerRadius="10dp"
        app:cardElevation="5dp"
        app:cardUseCompatPadding="true" >
```

CardView 태그 추가하기



카드뷰 모양으로 바꾸기

• 카드뷰 모양으로 바꾼 결과





4.

리싸이클러뷰 만들기



왜 굳이 선택위젯이라는 이름으로 구분할까?

- 안드로이드에서는 여러 아이템 중의 하나를 선택하는 선택위젯은 별도의 패턴을 사용함
- 여러 개의 아이템 중에서 하나를 선택하는 방식의 선택 위젯은 어댑터를 사용하여야 함
- 이 어댑터에서 데이터를 관리하도록 해야 할 뿐만 아니라 화면에 보여지는 뷰도 어댑터의 `getView()` 메소드에서 결정함
- 선택위젯의 가장 큰 특징은 원본 데이터를 위젯에 직접 설정하지 않고 어댑터라는 클래스를 사용하도록 되어 있다는 점으로 이 패턴을 잘 기억해 두어야 함

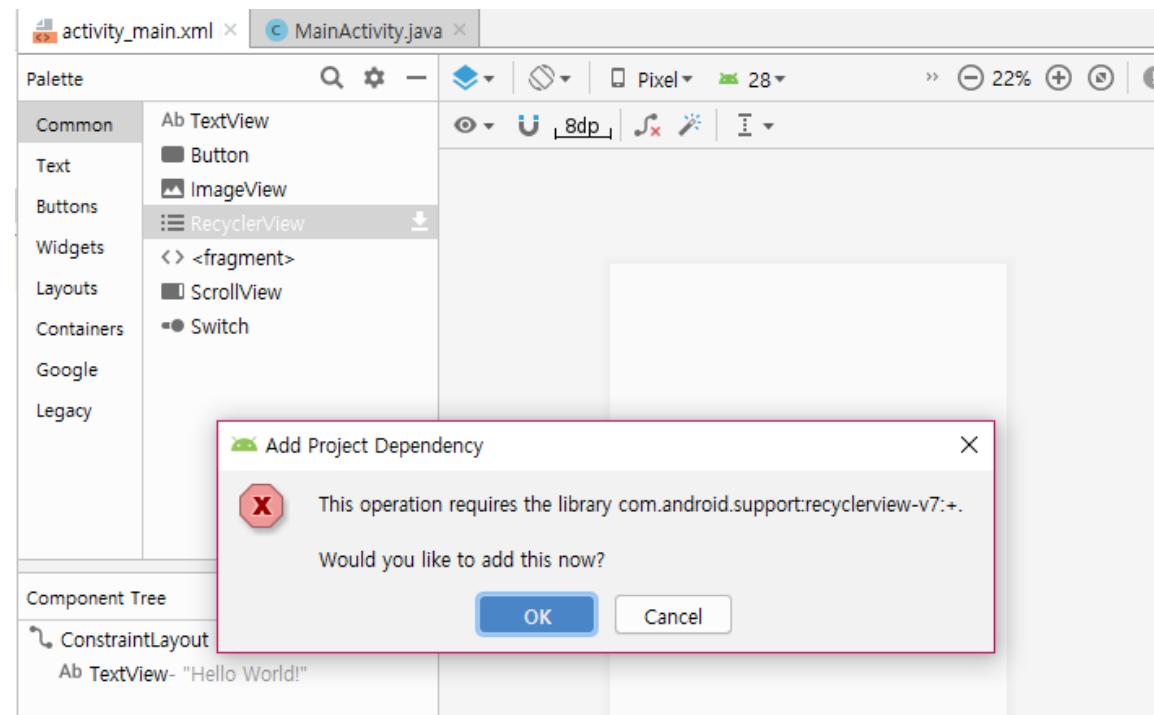


[선택 위젯과 어댑터]



리싸이클러뷰 만들기

- 빨렛에서 RecyclerView 오른쪽에 있는 아이콘 눌러 외부 라이브러리 추가





리싸이클러뷰 만들기

- activity_main.xml에 RecyclerView 태그 추가

참조파일 SampleRecyclerView>/app/res/layout/activity_main.xml

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

    <android.support.v7.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
```

} 리싸이클러뷰 태그 추가하기

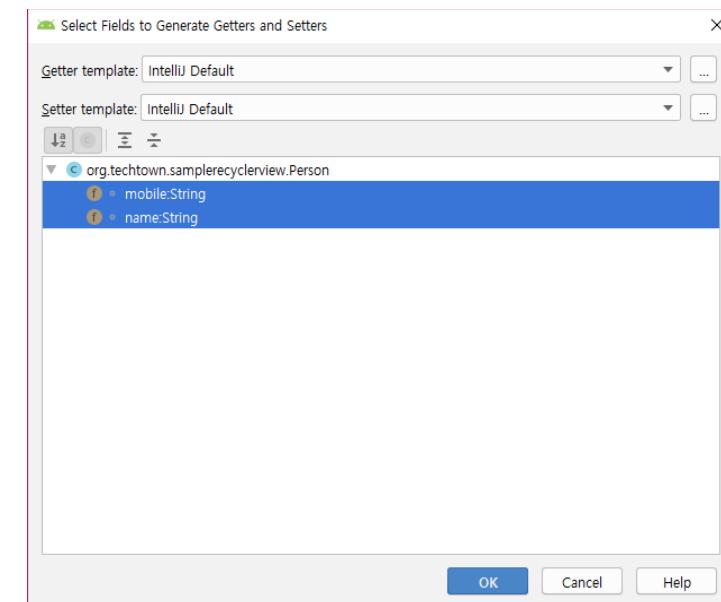
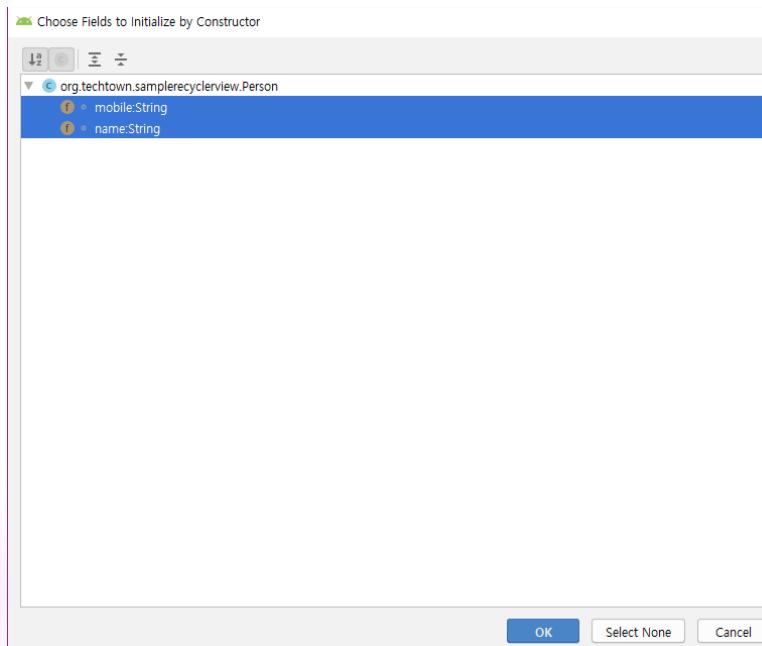


리싸이클러뷰 만들기

- 각 아이템을 위한 데이터를 담아두기 위해 Person 클래스 추가
- 생성자와 get, set 메서드 추가

참조파일 SampleRecyclerView>/app/java/org.techtown.samplerecyclerview/Person.java

```
public class Person {  
    String name;  
    String mobile;  
}
```





리싸이클러뷰 만들기

• 어댑터 소스 추가하고 ViewHolder 정의

참조파일 SampleRecyclerView>/app/java/org.techtown.samplerecyclerview/PersonAdapter.java

```
public class PersonAdapter {

    static class ViewHolder extends RecyclerView.ViewHolder {
        TextView textView;
        TextView textView2;

        public ViewHolder(View itemView) { ] ① 뷰홀더 생성자로 전달되는 뷰 객체 참조하기
            super(itemView);

            textView = itemView.findViewById(R.id.textView);
            textView2 = itemView.findViewById(R.id.textView2); ] ② 뷰 객체에 들어 있는 텍스트뷰 참조하기
        }

        public void setItem(Person item) {
            textView.setText(item.getName());
            textView2.setText(item.getMobile());
        }
    }
}
```



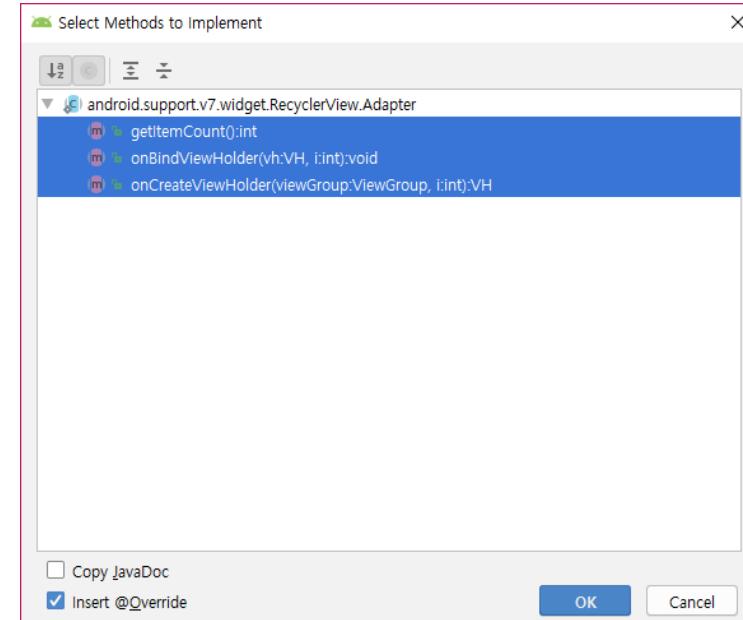
리싸이클러뷰 만들기

- PersonAdapter가 상속할 클래스 지정

참조파일 SampleRecyclerView>/app/java/org.techtown.samplerecyclerview/PersonAdapter.java

```
public class PersonAdapter extends RecyclerView.Adapter<PersonAdapter.ViewHolder> {
```

증략...





리싸이클러뷰 만들기

• onCreateViewHolder와 onBindViewHolder 메서드 재정의

참조파일 SampleRecyclerView>/app/java/org.techtown.samplerecyclerview/PersonAdapter.java

```
public class PersonAdapter extends RecyclerView.Adapter<PersonAdapter.ViewHolder> {
    ArrayList<Person> items = new ArrayList<Person>();

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int viewType) {
        LayoutInflater inflater = LayoutInflater.from(viewGroup.getContext());
        View itemView = inflater.inflate(R.layout.person_item, viewGroup, false);→ ❶ 인플레이션을
                                                                통해 뷰 객체
        return new ViewHolder(itemView);→ ❷ 뷰홀더 객체를 생성하면서 뷰 객체를
                                         만들기
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolder viewHolder, int position) {
        Person item = items.get(position);
        viewHolder.setItem(item);
    }

    @Override
    public int getItemCount() {
        return items.size();
    }
}
```

① 인플레이션을
통해 뷰 객체
만들기
② 뷰홀더 객체를 생성하면서 뷰 객체를
전달하고 그 뷰홀더 객체를 반환하기

중략...



리싸이클러뷰 만들기

• person_item.xml 파일 정의

참조파일 SampleRecyclerView>/app/res/layout/person_item.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="wrap_content"
    android:orientation="vertical">

    <android.support.v7.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:cardBackgroundColor="#FFFFFF"
        app:cardCornerRadius="10dp"
        app:cardElevation="5dp"
        app:cardUseCompatPadding="true" >

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">

            <ImageView
                android:id="@+id/imageView"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:padding="5dp"
                app:srcCompat="@mipmap/ic_launcher" />

        
```



리싸이클러뷰 만들기

- add, set, get 등의 메서드 추가

```
public void addItem(Person item) {  
    items.add(item);  
}  
  
public void setItems(ArrayList<Person> items) {  
    this.items = items;  
}  
  
public Person getItem(int position) {  
    return items.get(position);  
}  
  
public void setItem(int position, Person item) {  
    return items.set(position, item);  
}
```

중략...



리싸이클러뷰 만들기

- MainActivity의 onCreate 메서드 안에 리싸이클러뷰와 어댑터를 위한 코드 추가

참조파일 SampleRecyclerView>/app/java/org.techtown.samplerecyclerview/MainActivity.java

```
public class MainActivity extends AppCompatActivity {

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

        RecyclerView recyclerView = findViewById(R.id.recyclerView);

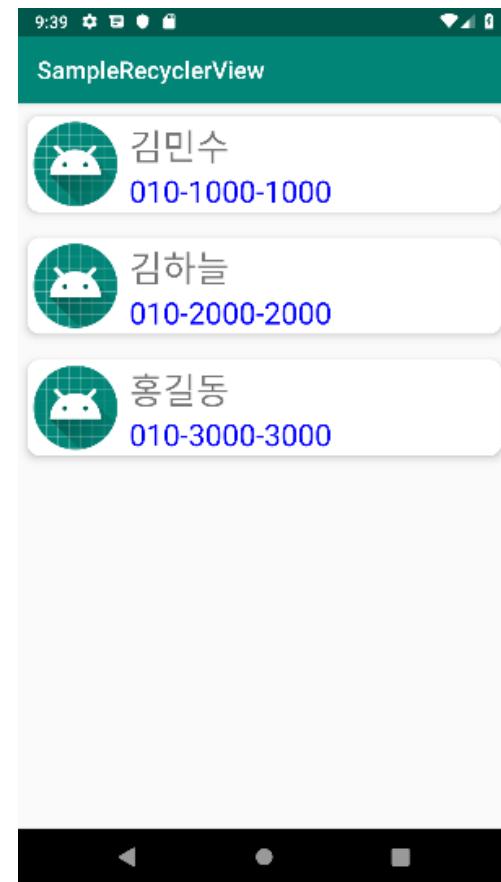
        LinearLayoutManager layoutManager =
                new LinearLayoutManager(this, LinearLayoutManager.VERTICAL, false); ①
        recyclerView.setLayoutManager(layoutManager);
        PersonAdapter adapter = new PersonAdapter();

        adapter.addItem(new Person("김민수", "010-1000-1000"));
        adapter.addItem(new Person("김하늘", "010-2000-2000"));
        adapter.addItem(new Person("홍길동", "010-3000-3000"));
    }
}
```



리싸이클러뷰 만들기

• 앱 실행





리싸이클러뷰 만들기

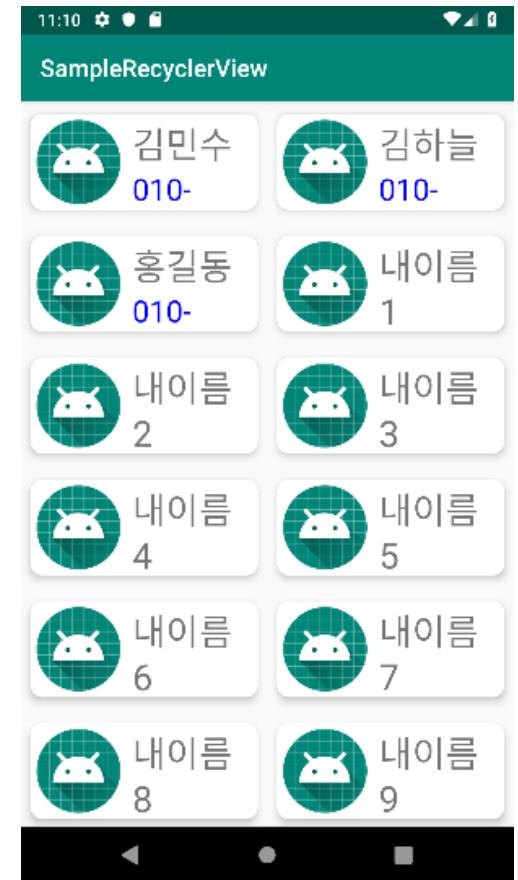
• 여러 칼럼의 뷰로 보여주기

참조파일 SampleRecyclerView2>/app/java/org.techtown.recyclerview/MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        RecyclerView recyclerView = findViewById(R.id.recyclerView);  
  
        GridLayoutManager layoutManager = new GridLayoutManager(this, 2);  
        recyclerView.setLayoutManager(layoutManager);  
  
    }  
}
```

중략...

리싸이클러뷰에 GridLayoutManager를 레이아웃 매니저로 설정하기





리싸이클러뷰 만들기

• 클릭했을 때 이벤트 처리

참조파일 SampleRecyclerView2>

중략...

```
static class ViewHolder extends RecyclerView.ViewHolder {
    TextView textView;
    TextView textView2;

    public ViewHolder(View itemView, final OnPersonItemClickListener listener) {
        super(itemView);

        textView = itemView.findViewById(R.id.textView);
        textView2 = itemView.findViewById(R.id.textView2);

        itemView.setOnClickListener(new View.OnClickListener() { → ① 아이템 뷰에 OnClickListener
            @Override
            public void onClick(View view) {
                int position = getAdapterPosition();

                if (listener != null) {
                    listener.onItemClick(ViewHolder.this, view, position); } → ② 아이템 뷰 클릭 시
                } 미리 정의한 다른 리스너의
            });
        });
});
```

참조파일 SampleRecyclerView2>/app/java/org.techtown.recyclerview/OnPersonItemClickListener.java

```
public interface OnPersonItemClickListener {
    public void onItemClick(PersonAdapter.ViewHolder holder, View view, int position);
}
```



리싸이클러뷰 만들기

• 어댑터에 리스너를 위한 변수 선언

참조파일 SampleRecyclerView2>/app/java/org.techtown.recyclerview/PersonAdapter.java

증략...

```
public class PersonAdapter extends RecyclerView.Adapter<PersonAdapter.ViewHolder> implements OnPersonItemClickListener {
    ArrayList<Person> items = new ArrayList<Person>();
    OnPersonItemClickListener listener;

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int viewType) {
        LayoutInflater inflater = LayoutInflater.from(viewGroup.getContext());
        View itemView = inflater.inflate(R.layout.person_item, viewGroup, false);

        return new ViewHolder(itemView, this);
    }
}
```

①

증략...



리싸이클러뷰 만들기

- 어댑터에 리스너를 위한 변수 선언

```
public void setOnItemClickListener(OnPersonItemClickListener listener) {  
    this.listener = listener;  
}  
  
@Override  
public void onItemClick(ViewHolder holder, View view, int position) {  
    if (listener != null) {  
        listener.onItemClick(holder, view, position);  
    }  
}
```

2

중략...



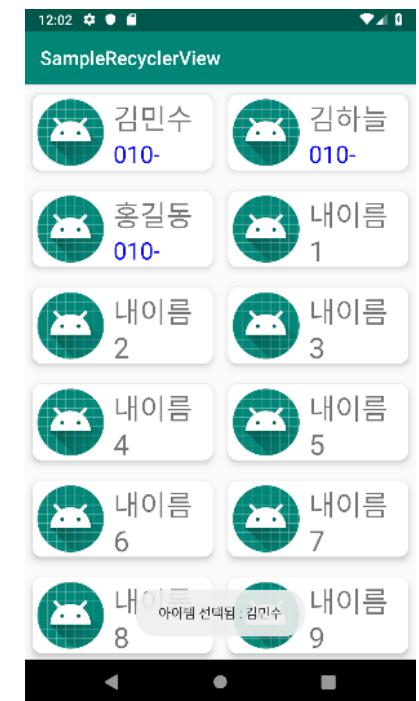
리싸이클러뷰 만들기

• 메인 액티비티에 추가

```
recyclerView.setAdapter(adapter);

adapter.setOnItemClickListener(new OnPersonItemClickListener() { → ①
    @Override
    public void onItemClick(PersonAdapter.ViewHolder holder, View view, int position) {

        Person item = adapter.getItem(position); → ②
        Toast.makeText(getApplicationContext(), "아이템 선택됨: " + item.getName(),
            Toast.LENGTH_LONG).show();
    }
});
```





5.

스피너 사용하기



스피너 사용하기 예제

스피너 사용하기 예제

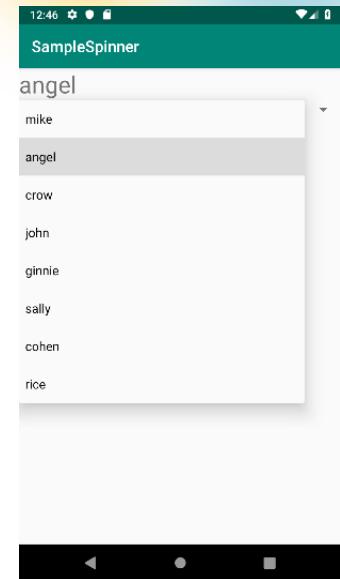
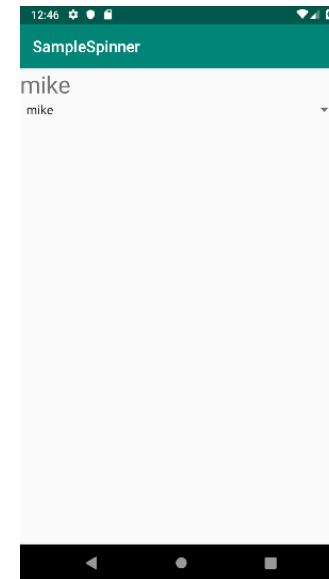
- 콤보박스처럼 선택할 수 있는 스피너 사용
- XML 레이아웃에 정의한 스피너 참조

메인 액티비티의
XML 레이아웃

- 스피너를 포함하는 메인 화면의
XML 레이아웃 정의

메인 액티비티 코드 작성

- 스피너 객체를 참조하여 설정





XML 레이아웃 만들기

- <Spinner> 태그를 사용하여 레이아웃에 추가

```
<Spinner  
    android:id="@+id/spinner"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
/>
```

1

스피너 정의



메인 액티비티 코드 만들기

- 안드로이드에서 미리 만들어 제공하는 어댑터를 사용할 수 있음

```
String[] items = { "mike", "angel", "crow", "john", "ginnie", "sally", "cohen", "rice" };
```

1 스피너에 표시될 아이템들의
데이터를 배열로 정의

```
selection = (TextView) findViewById(R.id.selection);
Spinner spinner = (Spinner) findViewById(R.id.spinner);
spinner.setOnItemSelectedListener(this);
```

Continued..



메인 액티비티 코드 만들기 (계속)

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>(  
    this, android.R.layout.simple_spinner_item, items);  
adapter.setDropDownViewResource(  
    android.R.layout.simple_spinner_dropdown_item);
```

```
    spinner.setAdapter(adapter);  
}
```

```
public void onItemSelected(AdapterView<?> parent, View v, int position, long id) {  
    selection.setText(items[position]);  
}
```

```
public void onNothingSelected(AdapterView<?> parent) {  
    selection.setText("");  
}
```

2 **ArrayAdapter를 이용해 어댑터
객체 생성**

3 **스피너에 어댑터 설정**

4 **스피너의 아이템이
선택되었을 때 처리하는
메소드 정의**

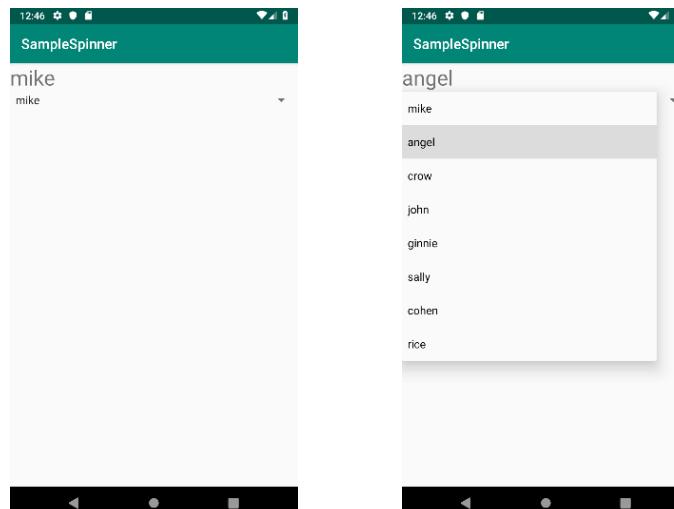


ArrayAdapter의 사용

[Reference]

```
public ArrayAdapter (Context context, int textViewResourceId, T[] objects)
```

- 첫 번째 파라미터는 Context 객체이므로 액티비티인 this를 전달하면 됨
- 두 번째 파라미터는 뷰를 초기화할 때 사용되는 XML 레이아웃의 리소스 ID 값으로 이 코드에서는 android.R.layout.simple_spinner_item을 전달하였음
- 세 번째 파라미터는 아이템으로 보일 문자열 데이터들의 배열임





애니메이션과 다양한 위젯에 대해 알아보기



목 차

1

애니메이션 사용하기

2

페이지 슬라이딩 사용하기

3

앱 화면에 웹브라우저 넣기

4

시크바 사용하기

5

키패드 제어하기



간단하게 애니메이션을 동작시킬 수 있나요?



페이지가 스윽~ 나타나도록 하려면 어떻게 해야 하나요?



앱 화면 안에서 웹사이트를 보여줄 수도 있나요?



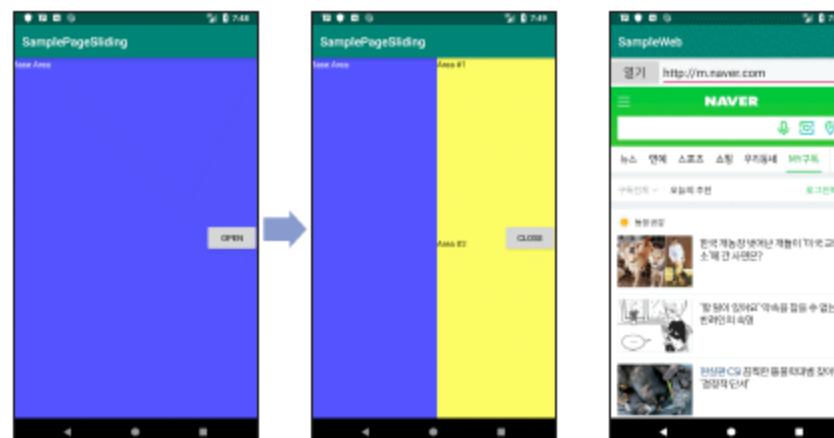
시크바나 키패드에 대해 더 알고 싶어요

- 애니메이션 사용하기

- 페이지 슬라이딩 만들기

- 앱 화면에 웹브라우저 넣기

- 시크바 사용하기
- 키패드 제어하기





1.

애니메이션 사용하기



자주 사용하는 애니메이션 방식

- 확대/축소, 이동, 회전, 투명도 조절 애니메이션을 자주 사용함
- 뷰 객체나 그리기 객체에 애니메이션을 적용할 수 있음



확대/축소



이동



회전



투명도



- **트윈 애니메이션(Tweened Animation)**

- 뷰 애니메이션이라고도 하며, 보여줄 대상을 적절하게 연산한 후 그 결과를 연속적으로 디스플레이하는 방식임
- 애니메이션 대상과 변환 방식을 지정하면 애니메이션 효과를 낼 수 있도록 만들어 줌
- 따라서 프레임 애니메이션처럼 변경하면서 보여줄 각각의 이미지를 추가할 필요 없이 대상만 지정하면 시스템이 내부적으로 적절하게 연산하는 과정을 거치게 됨

- **트윈 애니메이션을 위한 액션(Action) 정보**

- XML 리소스로 정의하거나 자바 코드에서 직접 객체로 만듬
- 애니메이션을 위한 XML 파일은 [/app/res/anim] 폴더의 밑에 두어야 하며 확장자를 xml로 함
- 리소스로 포함된 애니메이션 액션 정의는 다른 리소스와 마찬가지로 빌드할 때 컴파일되어 설치 파일에 포함됨



트윈 애니메이션 대상과 애니메이션 효과

구 분	이 름	설 명
대상	뷰	<ul style="list-style-type: none">- View는 위젯이나 레이아웃을 모두 포함- 예를 들어, 텍스트뷰나 리니어 레이아웃에 애니메이션을 적용할 수 있음
	그리기 객체	<ul style="list-style-type: none">- 다양한 Drawable에 애니메이션을 적용할 수 있음- ShapeDrawable은 캔버스에 그릴 도형을 지정할 수 있음- BitmapDrawable은 비트맵 이미지를 지정할 수 있음
효과	위치 이동	<ul style="list-style-type: none">- Translate로 정의되는 액션은 대상의 위치를 이동하기 위해 사용되는 효과
	확대 / 축소	<ul style="list-style-type: none">- Scale로 정의되는 액션은 대상의 크기를 크게하거나 작게 하기 위해 사용되는 효과
	회전	<ul style="list-style-type: none">- Rotate로 정의되는 액션은 대상을 회전하기 위해 사용되는 효과
	투명도	<ul style="list-style-type: none">- Alpha로 정의되는 액션은 대상의 투명도를 조절하는데 사용되는 효과



버튼 확대 애니메이션 예제

버튼 확대 애니메이션 예제

-버튼에 간단한 트윈 애니메이션 적용

메인 액티비티의
XML 레이아웃 정의

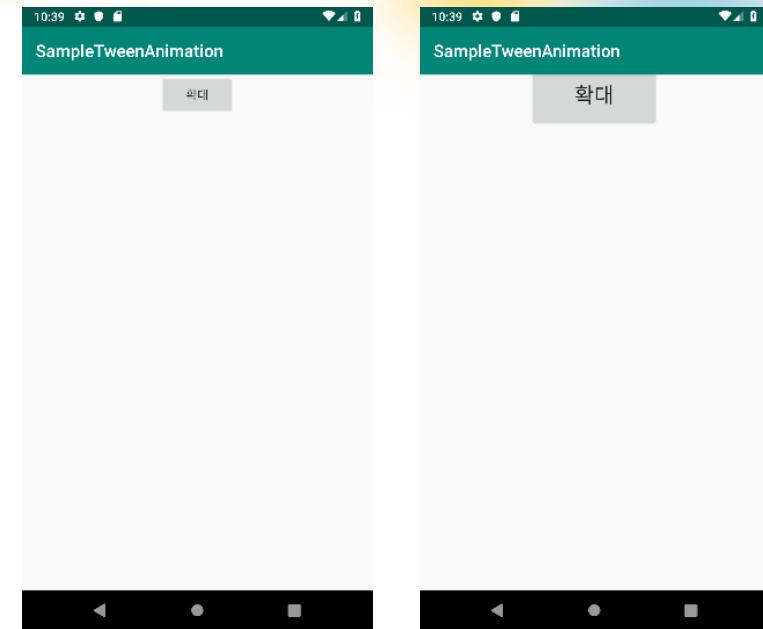
애니메이션 액션 정의

-버튼을 포함하는 레이아웃 정의

-XML로 애니메이션 액션 정의

메인 액티비티 코드 작성

-버튼에 애니메이션 적용





애니메이션 액션 XML 정의

- /app/res/anim 폴더 안에 새로운 XML 파일 생성

참조파일 SampleTweenAnimation>/app/res/anim/scale.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <scale
        android:duration="2500"
        android:pivotX="50%"
        android:pivotY="50%"
        android:fromXScale="1.0"
        android:fromYScale="1.0"
        android:toXScale="2.0"
        android:toYScale="2.0"
    />
</set>
```



애니메이션 <scale> 태그 사용

- **startOffset**

- 시작할 시간을 지정하는 것
- 애니메이션이 시작한 지 얼마 후에 이 액션이 수행될 것인지를 알 수 있도록 함

- **duration**

- 애니메이션이 지속되는 시간으로 여기에서는 2.5초 동안 지속되도록 되어 있음

- <**scale**> 태그

- 대상을 확대하거나 축소하는데 사용
- 크기를 변경하기 위한 축의 정보는 X축과 Y축에 대하여 각각 pivotX와 pivotY로 지정됨

- **fromXScale**과 **fromYScale**

- 시작할 때의 확대/축소 비율

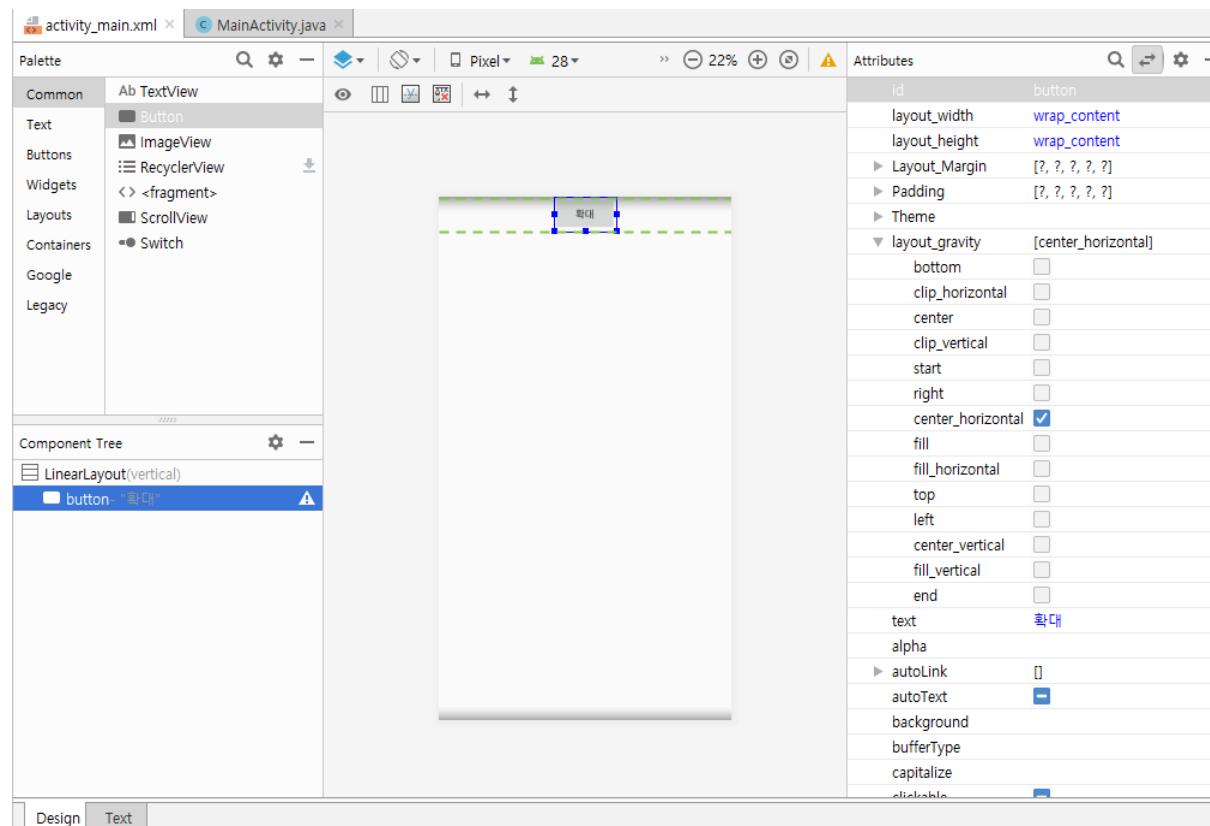
- **toXScale**과 **toYScale**

- 끝날 때의 확대/축소 비율



메인 액티비티의 화면 레이아웃 만들기

- 버튼 하나 추가





메인 액티비티 코드 만들기

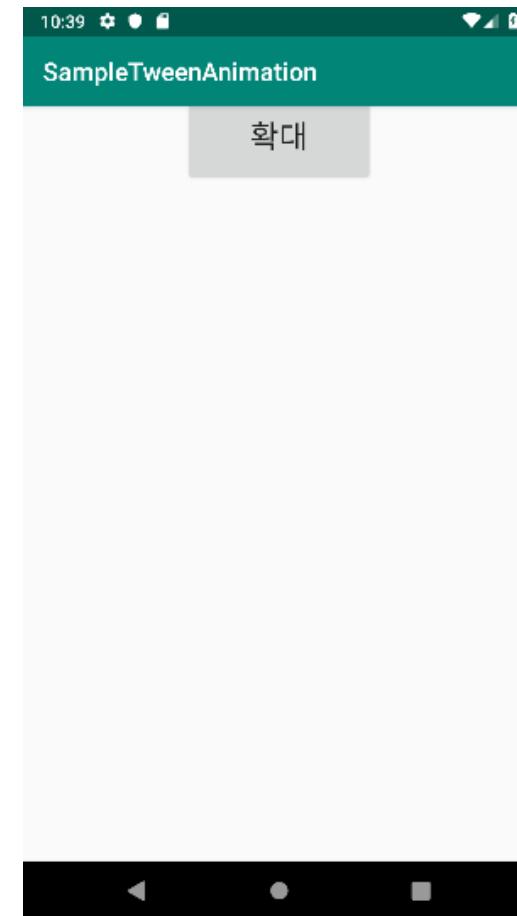
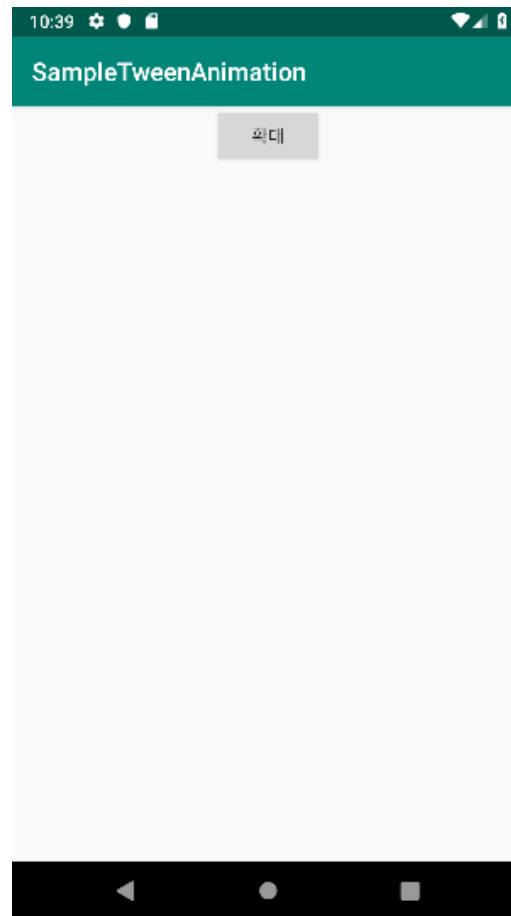
참조파일 SampleTweenAnimation>/app/java/org.techtown.sampletweenanimation/MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        Button button = findViewById(R.id.button);  
        button.setOnClickListener(new View.OnClickListener() {  
            public void onClick(View v) {  
  
                Animation anim =  
                    AnimationUtils.loadAnimation(getApplicationContext(),R.anim.scale);  
                v.startAnimation(anim); → ② 뷰의 애니메이션 시작  
            }  
        });  
    }  
}
```

① 리소스에 정의
한 애니메이션
액션 로딩



확대/축소 애니메이션 실행 화면





offset을 주어서 일정 시간 후 동작시키기

```
<scale
    android:startOffset="2500"
    android:duration="2500"
    android:pivotX="50%"
    android:pivotY="50%"
    android:fromXScale="1.0"
    android:fromYScale="1.0"
    android:toXScale="0.5"
    android:toYScale="0.5"
/>
</set>
```

② 2.5초 후에 시작할 확대/축소 애니메이션 액션 정의



메인 액티비티에 두 번째 버튼 추가

The screenshot shows the Android Studio interface with the following details:

- Top Bar:** activity_main.xml, scale.xml, MainActivity.java, scale2.xml
- Palette:** Common (selected), Text, Buttons, Widgets, Layouts, Containers, Google, Legacy.
- Component Tree:** LinearLayout(vertical) containing button-“확대” and button2-“확대/축소”.
- Design View:** Shows a vertical layout with two buttons. The top button is labeled “확대” and the bottom button is labeled “확대/축소”. The bottom button is currently selected.
- Attributes Panel:** Lists attributes for the selected button:
 - id: button2
 - layout_width: wrap_content
 - layout_height: wrap_content
 - Layout_Margin: [?, ?, ?, ?, ?]
 - Padding: [?, ?, ?, ?, ?]
 - Theme
 - text: 확대/축소
 - alpha
 - autoLink
 - autoText
 - background
 - bufferType
 - capitalize
 - clickable
 - contentDescription
 - cursorVisible
 - digits
 - drawableBottom
 - drawableEnd
 - drawableLeft
 - drawablePadding
 - drawableRight
 - drawableStart
 - drawableTop
 - drawingCacheQuality
 - duplicateParentState
 - editable
 - editorExtras
 - ellipsize
- Bottom Navigation:** Design (selected), Text



애니메이션 적용

참조파일 SampleTweenAnimation>/app/java/org.techtown.sampletweenanimation/MainActivity.java

```
public class MainActivity extends AppCompatActivity {

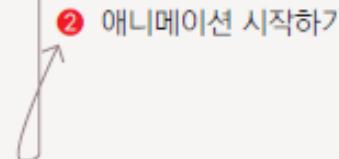
    @Override
    public void onCreate(Bundle savedInstanceState) {
        중략...

        Button button2 = findViewById(R.id.button2);
        button2.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {

                Animation anim =
                    AnimationUtils.loadAnimation(getApplicationContext(),R.anim.scale2); ←
                v.startAnimation(anim); } } ); }

}
```

① 애니메이션 정의한 것 로딩하기



② 애니메이션 시작하기



트윈 애니메이션 – 위치 이동 액션

참조파일 SampleTweenAnimation>/app/res/anim/translate.xml

```
<?xml version="1.0" encoding="utf-8"?>
<translate xmlns:android="http://schemas.android.com/apk/res/android"
    android:fromXDelta="0%p"
    android:toXDelta="-100%p"
    android:duration="20000"
    android:repeatCount="-1"
    android:fillAfter="true"
/>
```

- 위치 이동은 대상의 위치를 변경하는 것으로 한 곳에서 다른 곳으로 부드럽게 움직이는 효과를 낼 수 있음
- 위치 이동 액션은 <translate> 태그를 사용하여 정의하는데 시작 위치는 fromXDelta와 fromYDelta, 종료 위치는 toXDelta와 toYDelta라는 이름을 가진 속성으로 지정할 수 있음
- fromXDelta 속성이 0%이므로 시작 위치의 X 좌표는 원래 위치의 X 좌표가 됨
- toXDelta 속성이 -100%이므로 대상의 크기만큼 왼쪽으로 이동하게 됨
- 지속 시간은 duration의 값이 20000이므로 20초가 되며 repeatCount 속성이 -1이므로 무한반복하게 됨
- 애니메이션이 끝난 후에 대상이 원래의 위치로 돌아오는 것을 막기 위해서는 fillAfter 속성을 true로 하면 됨



트윈 애니메이션 – 회전 액션

참조파일 SampleTweenAnimation>/app/res/anim/rotate.xml

```
<?xml version="1.0" encoding="utf-8"?>
<rotate xmlns:android="http://schemas.android.com/apk/res/android"
    android:fromDegrees="0"
    android:toDegrees="360"
    android:pivotX="50%"
    android:pivotY="50%"
    android:duration="10000"
    />
```

- 회전은 한 점을 중심으로 대상을 회전시키는 효과를 만드는 액션으로써 시작 각도와 종료 각도를 지정할 수 있음
- 한 바퀴 회전시키려 한다면 fromDegrees 속성의 값을 0으로 하고 toDegrees 속성의 값을 360으로 함
- 시계 반대 방향으로 회전시키고 싶을 경우에는 toDegrees 속성의 값을 -360으로 함. 회전의 중심이 되는 점은 디폴트 값이 (0, 0)이므로 대상의 왼쪽 상단 끝 지점이 됨.
- 대상의 중앙 부분을 회전의 중심으로 만들고 싶다면 pivotX와 pivotY 속성의 값을 지정함
- 값의 단위는 좌표 값 또는 백분율(%)을 사용할 수 있음
- duration 속성의 값이 10000으로 설정되어 있으므로 10초 동안 애니메이션이 진행된 후 원래대로 돌아오게 됨



- **스케일**

- 대상을 크게하거나 작게 할 수 있는 액션
- 확대/축소의 정도는 대상이 갖는 원래 크기에 대한 비율로 결정
- 1.0이라는 값은 원래 크기와 동일하다는 의미이며, 2.0은 원래 크기의 두 배로 크게 만든다는 의미

- **X축으로 늘리거나 줄이고 싶으면?**

- `fromXScale`과 `toXScale` 속성을 이용하여 값을 설정

- **Y축으로 늘리거나 줄이고 싶으면?**

- `fromYScale`과 `toYScale` 속성을 이용하여 값을 설정

- **확대/축소의 경우**

- 중심이 되는 점을 지정할 수 있는데 앞에서와 마찬가지로 `pivotX`와 `pivotY` 속성을 이용



트윈 애니메이션 – 투명도 액션

참조파일 SampleTweenAnimation>/app/res/anim/alpha.xml

```
<?xml version="1.0" encoding="utf-8"?>
<alpha xmlns:android="http://schemas.android.com/apk/res/android"
    android:fromAlpha="0.0"
    android:toAlpha="1.0"
    android:duration="10000"
/>
```

- 투명도를 결정하는 알파 값도 뷰나 그리기 객체의 투명도를 점차적으로 바꿀 수 있는 애니메이션 액션으로 정의
- 알파 값을 이용한 투명도 변환은 대상을 천천히 보이게 하거나 보이지 않게 하고 싶을 때 또는 하나의 뷰 위에 다른 뷰를 겹쳐 보이게 할 경우에 사용됨
- 알파 값의 범위는 0.0부터 1.0까지이며 0.0은 알파 값이 0일 때와 마찬가지이므로 완전히 투명한 상태(뷰나 그리기 객체가 보이지 않음)이며 1.0은 알파 값이 1일 때와 마찬가지이므로 완전히 보이는 상태(투명 효과가 적용되지 않음)임



- **accelerate_interpolator**
 - 애니메이션 효과를 점점 빠르게 나타나도록 만듬
- **decelerate_interpolator**
 - 애니메이션 효과를 점점 느리게 나타나도록 만듬
- **accelerate_decelerate_interpolator**
 - 애니메이션 효과를 점점 빠르다가 느리게 나타나도록 만듬
- **anticipate_interpolator**
 - 애니메이션 효과를 시작 위치에서 조금 뒤로 당겼다가 시작하도록 만듬
- **overshoot_interpolator**
 - 애니메이션 효과를 종료 위치에서 조금 지나쳤다가 종료되도록 만듬



트윈 애니메이션 – 인터polator (계속)

- **anticipate_interpolator**

- 애니메이션 효과를 시작 위치에서 조금 뒤로 당겼다가 시작한 후 종료 위치에서 조금 지나쳤다가 종료되도록 만듬

- **bounce_interpolator**

- 애니메이션 효과를 종료 위치에서 튀도록 만듬

- 위치 이동 <translate> → TranslateAnimation

- 회전 <rotate> → RotateAnimation

- 확대/축소 <scale> → ScaleAnimation

- 투명도 <alpha> → AlphaAnimation

- 애니메이션 집합 <set> → AnimationSet



리스너 사용하기

- 리스너를 사용하면 애니메이션이 시작되거나 끝나는 시점을 알 수 있음

메서드	설명
<code>public void onAnimationStart(Animation animation)</code>	애니메이션이 시작되기 전에 호출됩니다.
<code>public void onAnimationEnd(Animation animation)</code>	애니메이션이 끝났을 때 호출됩니다.
<code>public void onAnimationRepeat(Animation animation)</code>	애니메이션이 반복될 때 호출됩니다.



2.

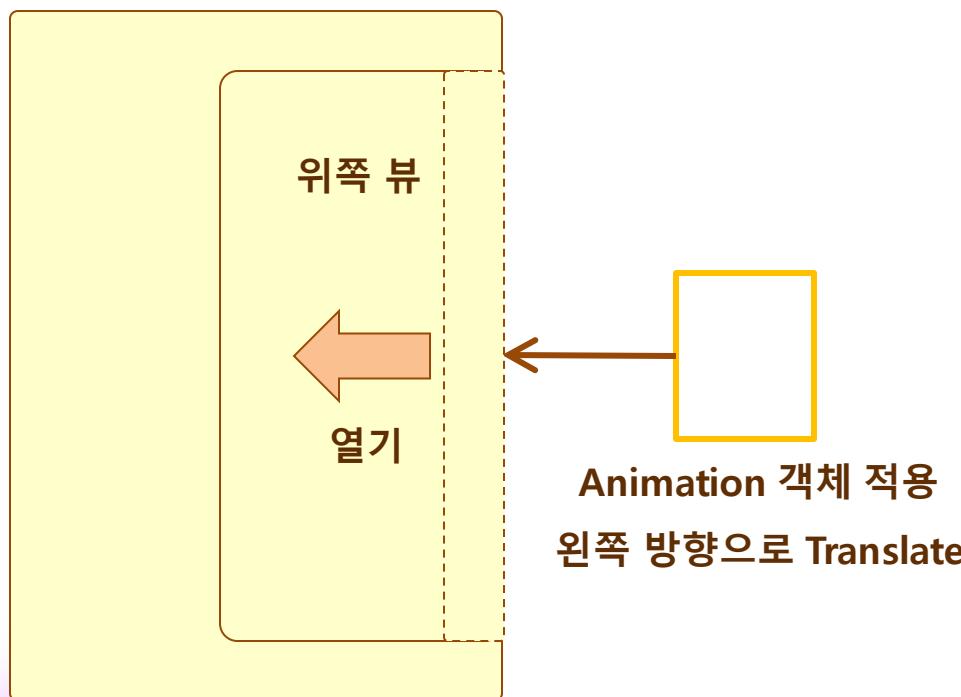
페이지 슬라이딩 사용하기



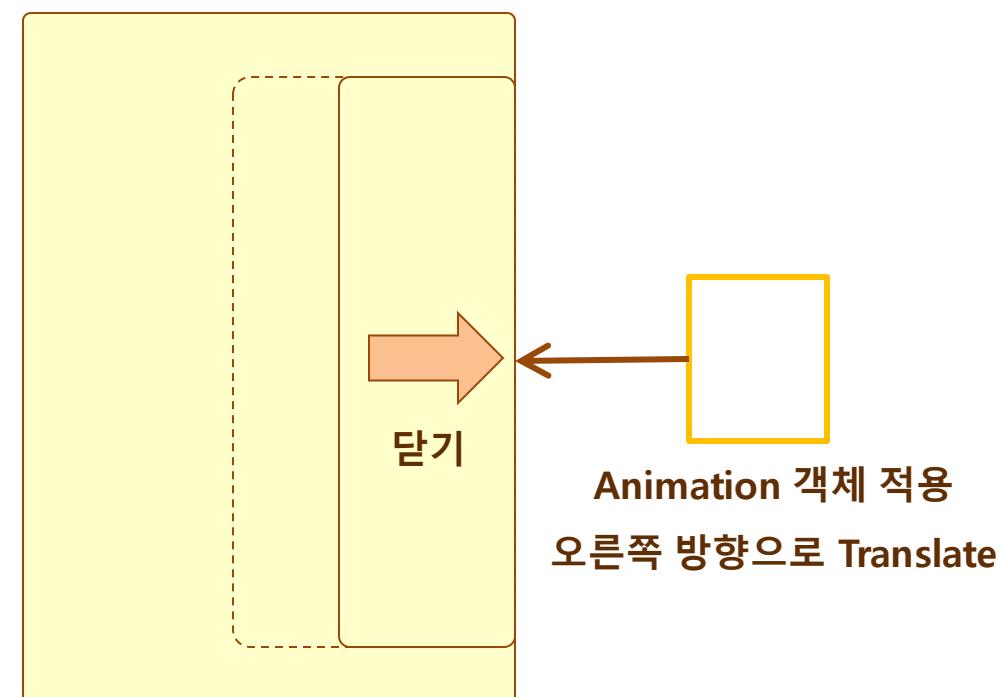
페이지 슬라이딩

- ▶ 뷰의 중첩과 애니메이션을 접목한 방식
- ▶ 하나의 뷰 위에 다른 뷰를 올라가 있을 때 보이거나 보이지 않는 과정을 애니메이션으로 적용

아래쪽 뷰



아래쪽 뷰





페이지 슬라이딩 사용하기

페이지 슬라이딩 사용하기 예제

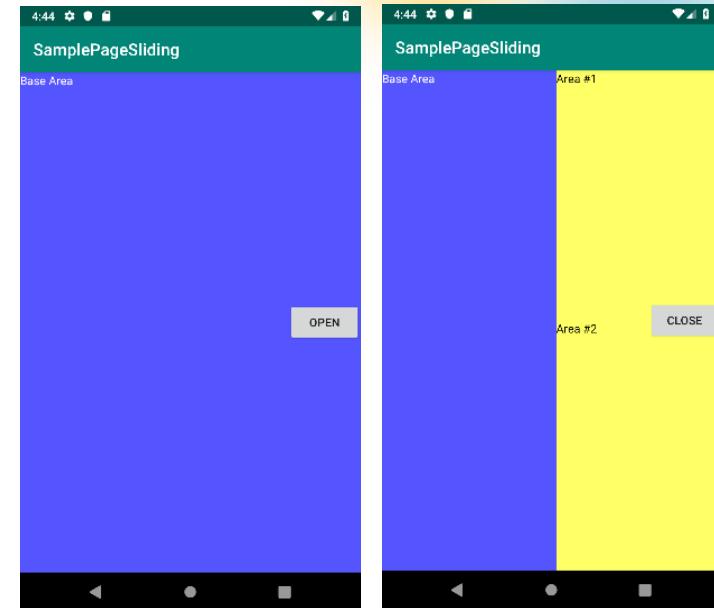
-페이지 슬라이딩을 이용해 뷰 보여주기

메인 액티비티의
XML 레이아웃 정의

메인 액티비티 코드 작성

-메인 액티비티 레이아웃 정의

-슬라이딩 기능 넣기





레이아웃 만들기

```
<LinearLayout  
    android:orientation= "vertical"  
    android:layout_width= "match_parent"  
    android:layout_height= "match_parent"  
    android:background= "#ff5555ff">  
    <TextView  
        android:layout_width= "wrap_content"  
        android:layout_height= "wrap_content"  
        android:text= "Base Area"  
        android:textColor= "#ffffffff"  
    />  
</LinearLayout>
```

1 첫 번째 레이아웃 : 바탕 레이아웃

Continued..



레이아웃 만들기 (계속)

```
<LinearLayout  
    android:id="@+id/slidingPage01"  
    android:orientation="vertical"  
    android:layout_width="200dp"  
    android:layout_height="match_parent"  
    android:layout_gravity="right"  
    android:background="#fffff66"  
    android:visibility="gone">  
  
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_weight="1"  
    android:text="Area #1"  
    android:textColor="#ff000000"  
/>
```

2 번째 레이아웃 :
슬라이딩으로 보일 레이아웃

Continued..



레이아웃 만들기 (계속)

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_weight="1"  
    android:text="Area #2"  
    android:textColor="#ff000000"  
/>  
</LinearLayout>
```

```
<LinearLayout  
    android:orientation="vertical"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="right|center_vertical"  
    android:background="#00000000">
```

```
<Button  
    android:id="@+id/openBtn01"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Open"  
/>  
</LinearLayout>  
</FrameLayout>
```

3

세 번째 레이아웃 :
버튼이 들어 있는 레이아웃



메인 액티비티 코드 만들기

참조파일 SamplePageSliding>/app/java/org.techtown.sliding/MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
    boolean isPageOpen = false;  
  
    Animation translateLeftAnim;  
    Animation translateRightAnim;  
  
    LinearLayout page;  
    Button button;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        page = findViewById(R.id.page);  
  
        translateLeftAnim = AnimationUtils.loadAnimation(this, R.anim.translate_left);  
        translateRightAnim = AnimationUtils.loadAnimation(this, R.anim.translate_right);  
  
        SlidingPageAnimationListener animListener = new SlidingPageAnimationListener();  
        translateLeftAnim.setAnimationListener(animListener);  
        translateRightAnim.setAnimationListener(animListener);  
    }  
}
```





메인 액티비티 코드 만들기 (계속)

```
button = findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (isPageOpen) {
            page.startAnimation(translateRightAnim);

        } else {
            page.setVisibility(View.VISIBLE);
            page.startAnimation(translateLeftAnim);
        }
    });
}
```



메인 액티비티 코드 만들기 (계속)

```
private class SlidingPageAnimationListener implements Animation.AnimationListener {  
  
    public void onAnimationEnd(Animation animation) {  
        if (isPageOpen) {  
            page.setVisibility(View.INVISIBLE);  
  
            button.setText("Open");  
            isPageOpen = false;  
        } else {  
            button.setText("Close");  
            isPageOpen = true;  
        }  
    }  
  
    @Override  
    public void onAnimationStart(Animation animation) { }  
  
    @Override  
    public void onAnimationRepeat(Animation animation) { }  
}
```

②



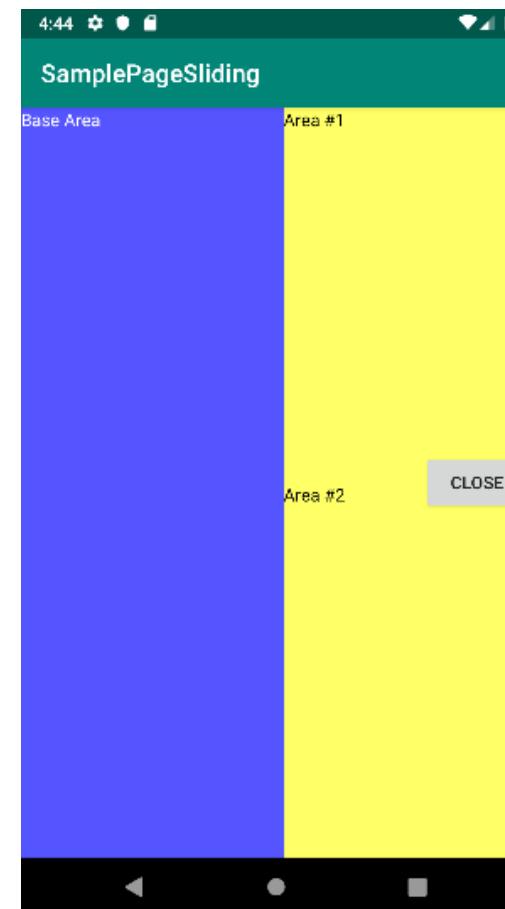
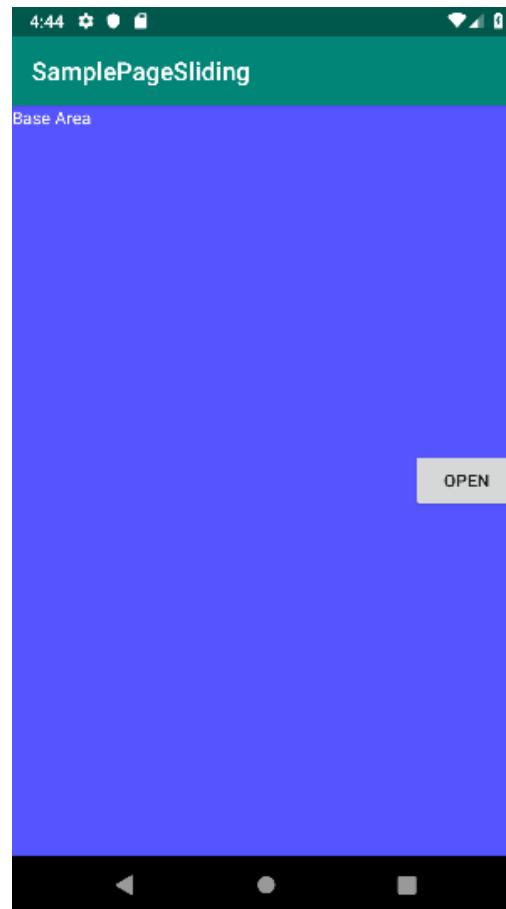
애니메이션을 위한 XML 정의

참조파일 SamplePageSliding>/app/res/anim/translate_left.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
      android:interpolator="@android:anim/accelerate_decelerate_interpolator">
    <translate
        android:fromXDelta="100%p"
        android:toXDelta="0%p"
        android:duration="500"
        android:repeatCount="0"
        android:fillAfter="true"
    />
</set>
```



실행 화면





3.

앱 화면에 웹브라우저 넣기



화면 레이아웃에 WebView 추가

- WebView는 웹브라우저를 보여주는 영역

```
<WebView  
    android:id="@+id/webView"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
/>
```

③ 웹뷰 정의



메인 액티비티에 설정 코드 추가

참조파일 SampleWeb>/app/java/org.techtown.web/MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    EditText editText;
    WebView webView;

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

        editText = findViewById(R.id.editText);
        webView = findViewById(R.id.webView);

        WebSettings webSettings = webView.getSettings(); ] ① 웹뷰의 설정 수정하기
        webSettings.setJavaScriptEnabled(true);

        webView.setWebViewClient(new ViewClient());

        Button button = findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                webView.loadUrl(editText.getText().toString()); —→ ② 버튼 클릭 시 사이트 로딩하기
            }
        });
    }
}
```



메인 액티비티에 설정 코드 추가

```
private class ViewClient extends WebViewClient {  
    @Override  
    public boolean shouldOverrideUrlLoading(final WebView view, final String url) {  
        view.loadUrl(url);  
  
        return true;  
    }  
}  
}
```

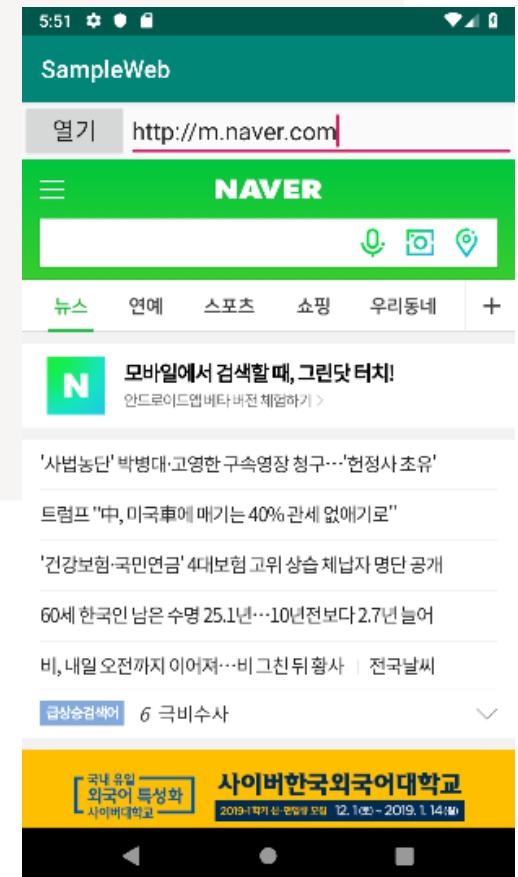


매니페스트에 인터넷 사용 권한 추가 후 실행

• INTERNET 권한

참조파일 SampleWeb>/app/manifests/AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="org.techtown.web">  
  
    <uses-permission android:name="android.permission.INTERNET"/>  
  
    <application  
        android:usesCleartextTraffic="true"  
  
        중략...
```





4.

시크바 사용하기



시크바를 위해 정의된 메서드들

- 시크바의 상태가 바뀌는 것을 확인할 수 있음

[Code]

```
void onStartTrackingTouch (SeekBar seekBar)  
void onStopTrackingTouch (SeekBar seekBar)  
void onProgressChanged (SeekBar seekBar, int progress, boolean fromUser)
```



XML 레이아웃에 SeekBar 추가

참조파일 SampleSeekBar>/app/res/layout/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">  
  
    <SeekBar  
        android:id="@+id/seekBar"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:max="100" />  
  
    <TextView  
        android:id="@+id/textView"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:layout_marginTop="20dp"  
        android:text="변경된 값"  
        android:textSize="30sp" />  
  
</LinearLayout>
```

SeekBar 태그 추가하기



시크바를 추가한 화면 레이아웃

The screenshot shows the Android Studio Layout Editor with the following details:

- Top Bar:** activity_main.xml (selected), MainActivity.java, Pixel: 28, AppTheme, 22% zoom.
- Palette:** Common (selected), showing Text, Buttons, Widgets, Layouts, Containers, Google, Legacy, and a section for Ab TextView.
- Component Tree:** LinearLayout (vertical) containing a seekBar and an Ab textView labeled "변경된 값".
- Preview:** A preview window showing a white box with a blue border containing the text "변경된 값".
- Attributes Panel:** Displays attributes for the selected textView:

id	textView
layout_width	match_parent
layout_height	wrap_content
Layout_Margin	[?, ?, 20dp, ?, ?]
all	
top	20dp
bottom	
left	
right	
Padding	[?, ?, ?, ?, ?]
Theme	
text	변경된 값
textSize	30sp
alpha	
autoLink	
autoSizeMaxTextSize	
autoSizeMinTextSize	
autoSizePresetSizes	
autoSizeStepGranular	
autoSizeTextType	
autoText	
background	
bufferType	
capitalize	
clickable	
contentDescription	
cursorVisible	
digits	
drawableBottom	
- Bottom:** Design (selected) and Text tabs.



메인 액티비티 코드에서 값 변경 시 동작 설정

```
SeekBar seekBar = findViewById(R.id.seekBar);

seekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() { → ① 시크바에
    @Override                                         리스너
    public void onProgressChanged(SeekBar seekBar, int i, boolean b) {           설정하기
        setBrightness(i);
        textView.setText("변경된 값: " + i);
    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) { }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) { }
});
```



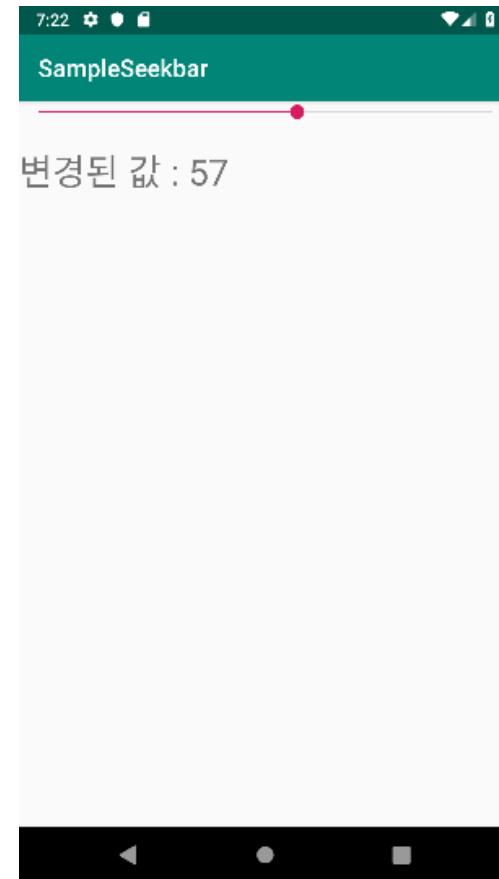
화면 밝기 설정하는 메서드

```
private void setBrightness(int value) {  
    if (value < 10) {  
        value = 10;  
    } else if (value > 100) {  
        value = 100;  
    }  
  
    WindowManager.LayoutParams params = getWindow().getAttributes();  
    params.screenBrightness = (float) value / 100;  
    getWindow().setAttributes(params);  
}
```

② 원도우 매니저를 이용해
화면 밝기 설정하기



앱 실행하여 시크바 값 변경





5.

키패드 제어하기



키패드 관련 속성

- `showSoftInput`, `hideSoftInputFromWindow` 메서드를 이용해 키패드를 보이거나 보이지 않도록 할 수 있음
- 입력상자에는 `inputType` 속성이 있어 입력될 값의 유형에 따라 다른 키패드가 보이도록 함

[Reference]

```
boolean showSoftInput(View view, int flags)
```

```
boolean hideSoftInputFromWindow(IBinder windowToken, int flags  
[, ResultReceiver resultReceiver ])
```

inputType 속성 값	설명
number	숫자
numberSigned	0보다 큰 숫자
numberDecimal	정수
text	텍스트
textPassword	패스워드로 표시
textEmailAddress	이메일로 표시
phone	전화번호로 표시
time	시간
date	날짜



화면 레이아웃 만들기

The screenshot shows the Android Studio Layout Editor interface. The top bar displays three tabs: `activity_main.xml`, `AndroidManifest.xml`, and `MainActivity.java`. The main area is divided into several sections:

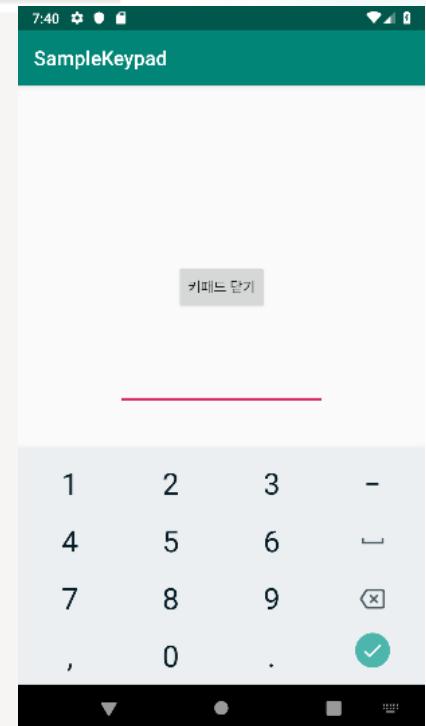
- Palette:** On the left, under the **Common** category, the **Button** item is selected.
- Component Tree:** Shows a tree structure with `ConstraintLayout` as the root, containing `editText(Plain Text)` and `button - "키패드 닫기"`.
- Design View:** The central view shows a white rectangular layout. Inside, there is a horizontal blue constraint chain consisting of two blue nodes connected by a blue line with arrows at both ends. A vertical dashed blue line labeled "52" extends downwards from the bottom node of the chain. The bottom node is also connected to a blue line with arrows pointing right and left.
- Attributes Panel:** On the right, it lists various attributes for the selected `button` element. Some attributes have their values changed to Korean text, such as `text` set to `키패드 닫기`.



버튼 클릭 시 키패드 숨기는 코드 추가

참조파일 SampleKeypad>/app/java/org.techtown.keypad/MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        Button button = findViewById(R.id.button);  
        button.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                ifgetCurrentFocus()!=null) {  
                    InputMethodManager inputMethodManager = (InputMethodManager)  
                        getSystemService(INPUT_METHOD_SERVICE);  
                }  
                inputMethodManager.hideSoftInputFromWindow(getCurrentFocus().getWindowToken(), 0);  
            }  
        });  
    }  
}
```





스레드와 핸들러에 대해 이해하기



- 1 핸들러 이해하기
- 2 일정시간 후에 실행하기
- 3 스레드로 메시지 전송하기
- 4 AsyncTask 사용하기
- 5 스레드로 애니메이션 만들기



스레드 결과를 화면에 보여주고 싶어요.

- 핸들러 이해하기

스레드를 다루는 방법을 구체적으로 알고 싶어요.

- 일정시간 후에 실행하기
- 스레드로 메시지 전송하기

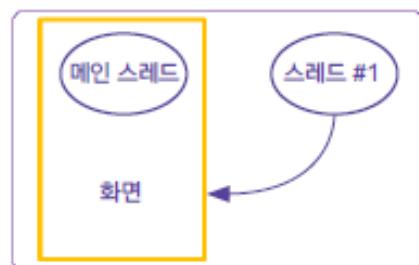
하나의 클래스에서 스레드 실행과 화면 표시를 같이 할 수 있나요?

- AsyncTask 사용하기

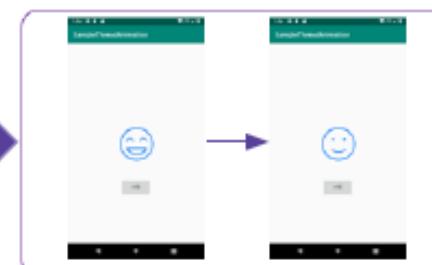
스레드를 이용해서 애니메이션을 만들 수 있나요?

- 스레드로 애니메이션 만들기

스레드에서 화면에 보여주기



스레드로 애니메이션 만들기



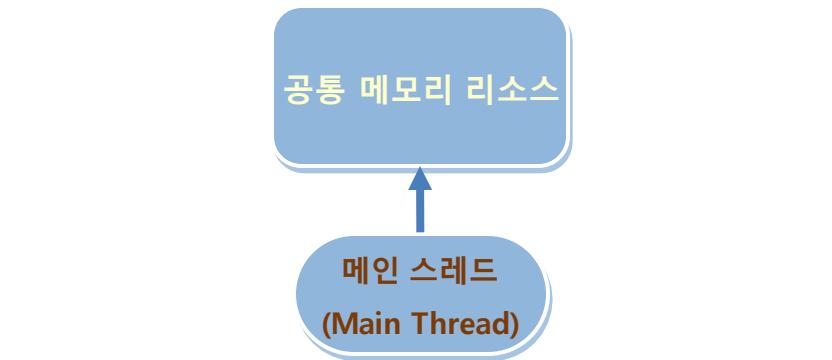


1.

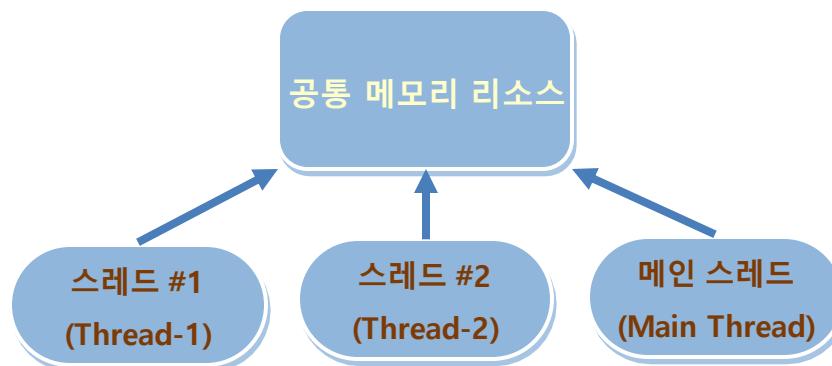
핸들러 이해하기



멀티 스레드



(1) 프로젝트 생성 시



(2) 별도의 스레드 생성 시

[멀티스레드 시스템에서 시스템에서
공통 메모리 리소스 접근]

• 메인 액티비티

- 애플리케이션이 실행될 때 하나의 프로세스에서 처리
- 이벤트를 처리하거나 필요한 메소드를 정의하여 기능을 구현하는 경우에도 동일한 프로세스 내에서 실행

• 문제점

- 대기 시간이 길어지는 네트워크 요청 등의 기능을 수행할 때는 화면에 보이는 UI도 멈춤 상태로 있게 됨

• 해결 방안

- 하나의 프로세스 안에서 여러 개의 작업이 동시에 수행되는 멀티 스레드 방식을 사용

• 멀티 스레드

- 같은 프로세스 안에 들어 있으면서 메모리 리소스를 공유하게 되므로 효율적인 처리가 가능
- 동시에 리소스를 접근할 경우 데드락(DeadLock) 발생



핸들러 사용하기

시나리오	설명
서비스 사용	백그라운드 작업은 서비스로 실행하고 사용자에게는 알림 서비스를 이용해 알려줌 만약 메인 액티비티로 결과값을 전달하고 이를 이용해 다른 작업을 수행하고자 한다면 브로드캐스팅을 이용하거나 하여 결과값을 전달할 수 있음
스레드 사용	스레드는 동일 프로세스 내에 있기 때문에 작업 수행의 결과를 바로 처리할 수 있음 그러나 UI 객체는 직접 접근할 수 없으므로 핸들러(Handler) 객체를 사용함



표준 자바에서 스레드 사용 방법

- 스레드는 new 연산자를 이용하여 객체를 생성한 후 start() 메소드를 호출하면 시작함
- Thread 클래스에 정의된 생성자는 크게 파라미터가 없는 경우와 Runnable 객체를 파라미터로 가지는 두 가지로 구분함

```
running = true;
```

```
Thread thread1 = new BackgroundThread();
thread1.start();
```



표준 자바에서 스레드 사용 방법 (계속)

- 버튼을 누르면 value 변수에 들어있는 정수 타입의 값을 텍스트뷰에 보여줌
- 스레드를 'BackgroundThread'라는 이름으로 정의

```
class BackgroundThread extends Thread {  
    public void run() {  
        while (running) {  
            try {  
                Thread.sleep(1000);  
                value++;  
            } catch (InterruptedException ex) {  
                Log.e("SampleJavaThread", "Exception in thread.", ex);  
            }  
        }  
    }  
}
```



메시지 전송하여 실행하기

- **메인 스레드**

- 애플리케이션 객체인 액티비티, 브로드캐스트 수신자 등과 새로 만들어지는 윈도우를 관리하기 위한 메시지 큐(Message Queue)를 실행함

- **메시지 큐 : Message Queue**

- 순차적으로 코드를 수행함

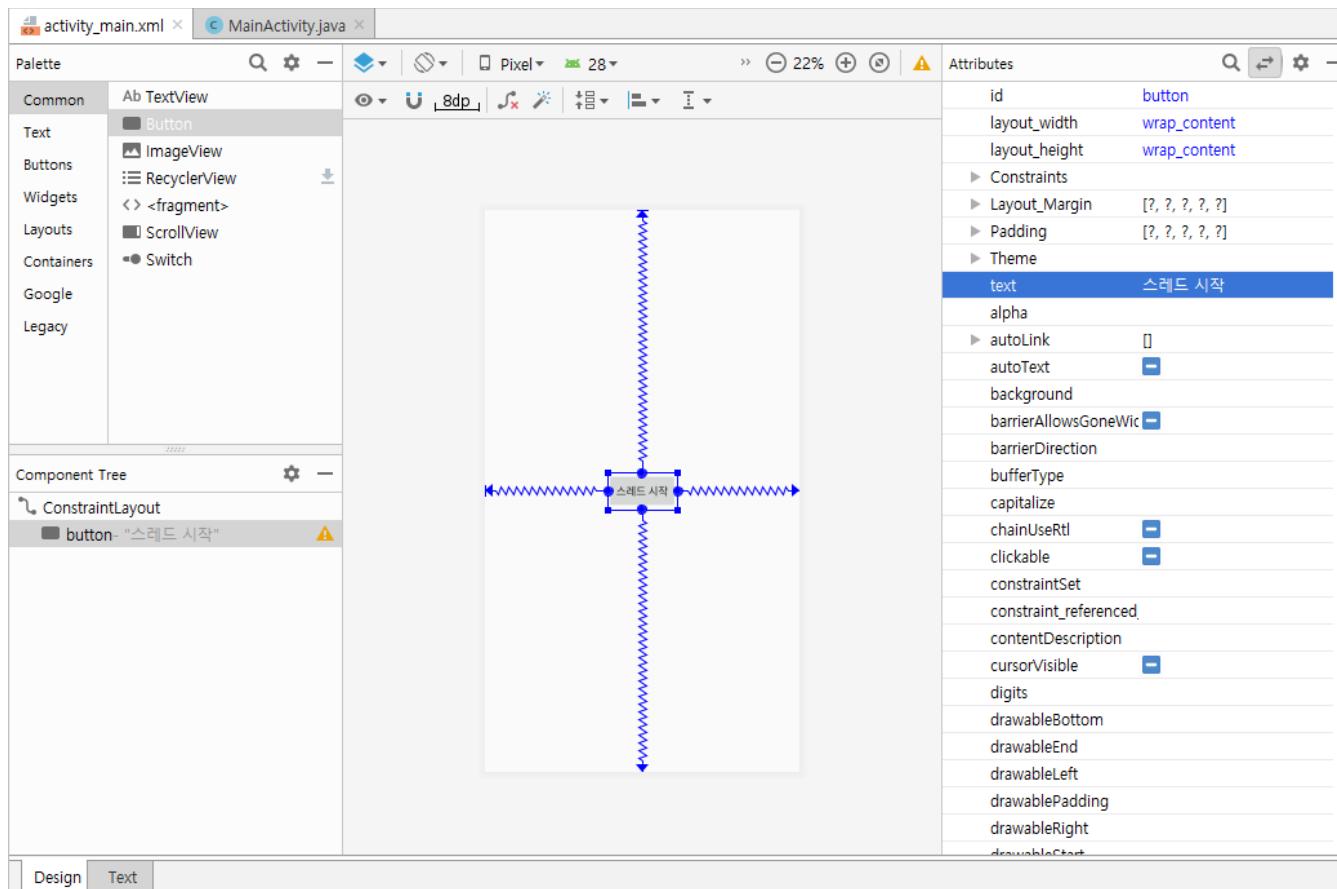
- **핸들러 : Handler**

- 메시지 큐를 이용해 메인 스레드에서 처리할 메시지를 전달하는 역할을 담당함
 - 특정 메시지가 미래의 어떤 시점에 실행되도록 스케줄링 할 수 있음



스레드 사용하기

- SampleThread 프로젝트 만들고 화면 레이아웃 구성





스레드 사용하기

- 메인 액티비티에서 버튼 클릭 시 스레드 동작하도록 코드 추가

참조파일 SampleThread>/app/java/org.techtown.thread/MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    int value = 0;

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

        Button button = findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                BackgroundThread thread = new BackgroundThread();
                thread.start(); } });
    }
}
```

↗ 스레드 객체 생성하고 시작시키기



스레드 사용하기

- 메인 액티비티에서 버튼 클릭 시 스레드 동작하도록 코드 추가
- 앱 실행하고 버튼 누르면 로그 정상 출력됨

```
class BackgroundThread extends Thread {  
    public void run() {  
        for (int i = 0; i < 100; i++) {  
            try {  
                Thread.sleep(1000);  
            } catch(Exception e) {}  
  
            value += 1;  
            Log.d("Thread", "value : " + value);  
        }  
    }  
}
```

The screenshot shows the Android Studio Logcat window. The window title is 'Logcat' and it displays log entries for the process 'org.techtown.thread (17864)'. The log entries show a sequence of values from 1 to 6, each preceded by the timestamp '2018-12-03 17:29:36.001' and the message 'D/Thread: value : [value]'. The logcat interface includes dropdown menus for 'Emulator Nexus_5X_API_28 Andri', 'org.techtown.thread (17864)', and 'Verbose', and a checkbox for 'Regex'.

Time	Message
2018-12-03 17:29:36.001	D/Thread: value : 1
2018-12-03 17:29:37.003	D/Thread: value : 2
2018-12-03 17:29:38.005	D/Thread: value : 3
2018-12-03 17:29:39.007	D/Thread: value : 4
2018-12-03 17:29:40.012	D/Thread: value : 5
2018-12-03 17:29:41.017	D/Thread: value : 6



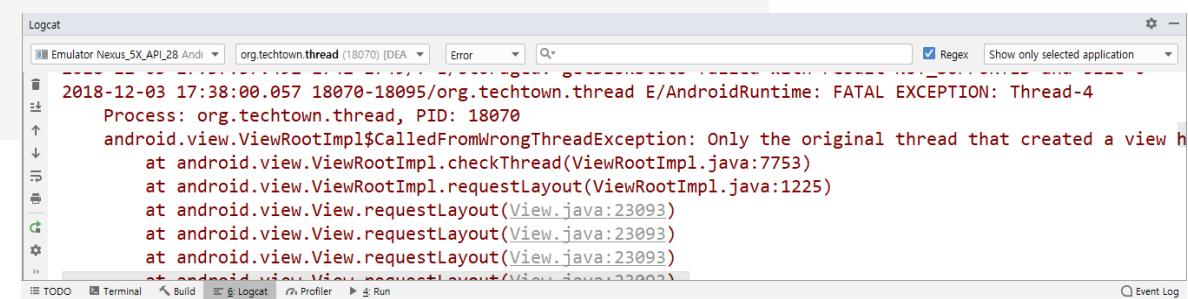
스레드 사용하기

- 스레드 안에서 텍스트뷰에 결과를 출력하도록 설정
- 앱 실행하면 오류 발생

```
class BackgroundThread extends Thread {  
    public void run() {  
        for (int i = 0; i < 100; i++) {  
            try {  
                Thread.sleep(1000);  
            } catch(Exception e) {}  
  
            value += 1;  
            Log.d("Thread", "value : " + value);  
        }  
    }  
}
```

textView.setText("value 값: " + value); → 스레드 안에서 텍스트뷰의 setText()

메서드 호출하기

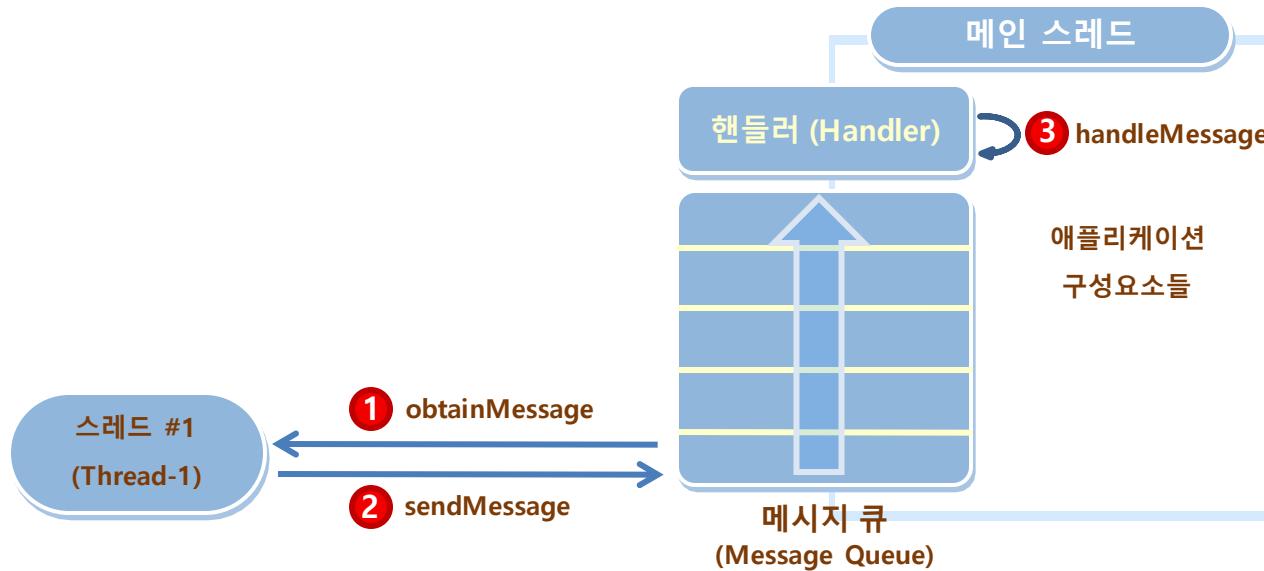


The screenshot shows the Android Studio Logcat window. The log output is as follows:

```
2018-12-03 17:38:00.057 18070-18095/org.techtown.thread E/AndroidRuntime: FATAL EXCEPTION: Thread-4  
Process: org.techtown.thread, PID: 18070  
    android.view.ViewRootImpl$CalledFromWrongThreadException: Only the original thread that created a view h  
        at android.view.ViewRootImpl.checkThread(ViewRootImpl.java:7753)  
        at android.view.ViewRootImpl.requestLayout(ViewRootImpl.java:1225)  
        at android.view.View.requestLayout(View.java:23093)  
        at android.view.View.requestLayout(View.java:23093)  
        at android.view.View.requestLayout(View.java:23093)  
        at android.view.View.requestLayout(View.java:23093)
```



핸들러로 메시지 전송하여 실행하는 구조



[핸들러를 사용할 때 필요한 세 가지 단계]

- **obtainMessage()**

- 호출의 결과로 메시지 객체를 리턴받게 함

- **sendMessage()**

- 메세지큐에 넣음

- **handleMessage()**

- 메소드에 정의된 기능이 수행됨
- 코드가 수행되는 위치는 새로 만든 스레드가 아닌 메인 스레드가 됨



핸들러 사용하기

- Handler 클래스 상속하여 새로운 클래스 정의
- 전달된 메시지 처리

```
class MainHandler extends Handler {  
  
    @Override  
    public void handleMessage(Message msg) {  
        super.handleMessage(msg);  
  
        Bundle bundle = msg.getData();  
        int value = bundle.getInt("value");  
        textView.setText("value 값: " + value);  
    }  
}  
}
```

② 핸들러 안에서 전달받은 메시지 객체 처리하기



핸들러 사용하기

- 스레드에서 핸들러로 전달
- 앱 실행하면 정상 동작

```
class BackgroundThread extends Thread {  
    int value = 0;  
  
    public void run() {  
        for (int i = 0; i < 100; i++) {  
            try {  
                Thread.sleep(1000);  
            } catch(Exception e) {}  
            value += 1;  
            Log.d("Thread", "value : " + value);  
  
            Message message = handler.obtainMessage();  
            Bundle bundle = new Bundle();  
            bundle.putInt("value", value);  
            message.setData(bundle);  
  
            handler.sendMessage(message); —→ ① 핸들러로 메시지 객체 보내기  
        }  
    }  
}
```





Runnable 객체 실행하기

- **핸들러 클래스**
 - 메시지 전송 방법 이외에 Runnable 객체를 실행시킬 수 있는 방법을 제공함
- **Runnable 객체**
 - 새로 만든 Runnable 객체를 핸들러의 post() 메소드를 이용해 전달해 주기만 하면 이 객체에 정의된 run() 메소드 내의 코드들은 메인 스레드에서 실행됨



Runnable 객체 실행하기

- Runnable 객체 안에서 처리하면 메인 스레드에서 실행되게 됨

```
class BackgroundThread extends Thread {  
    int value = 0;  
  
    public void run() {  
        for (int i = 0; i < 100; i++) {  
            try {  
                Thread.sleep(1000);  
            } catch(Exception e) {}  
  
            value += 1;  
            Log.d("Thread", "value : " + value);  
  
            handler.post(new Runnable() {————② 핸들러의 post() 메서드 호출하기  
                @Override  
                public void run() {  
                    textView.setText("value 값: " + value);  
                }  
            });  
        }  
    }  
}
```

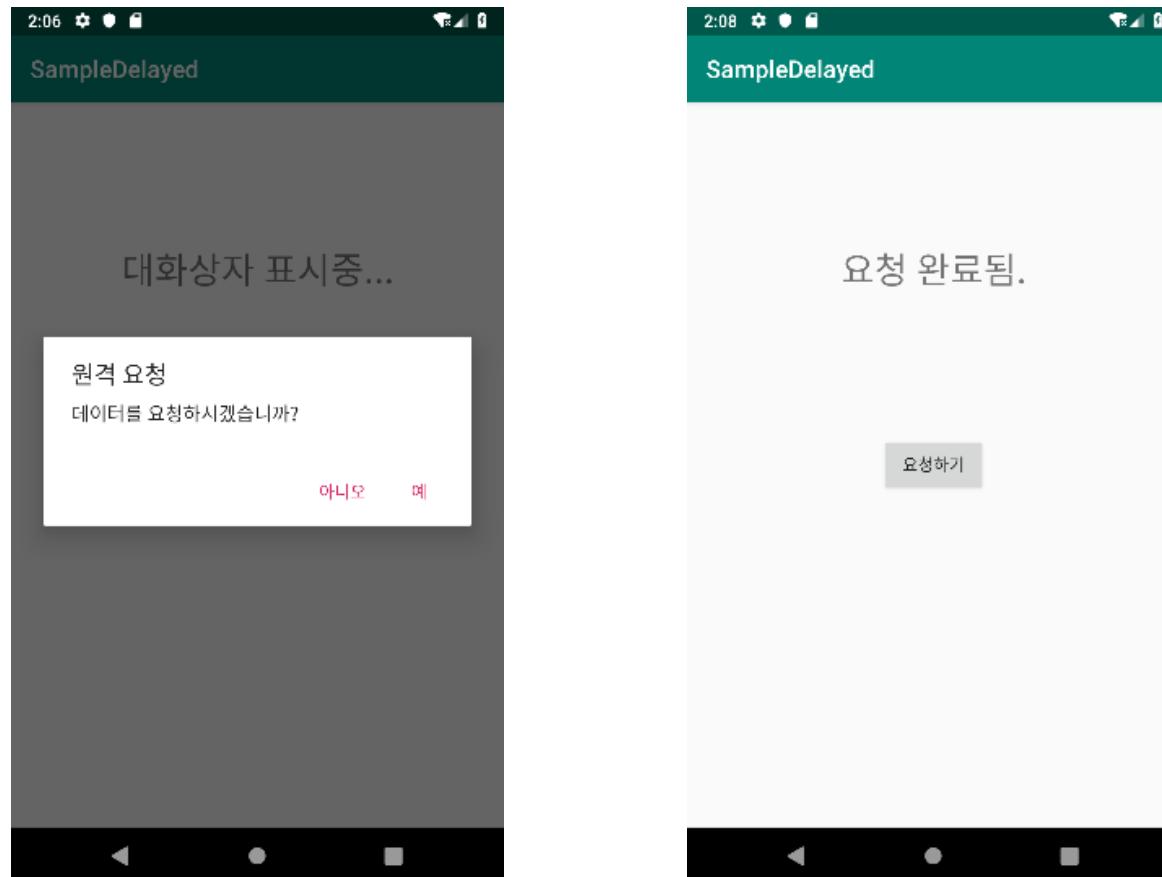


2.

일정시간 후에 실행하기



일정시간 후에 실행하기 – 가상의 네트워크 요청



[버튼을 클릭했을 때 가상으로 원격 서버 요청]



메인 액티비티 코드 만들기

```
private AlertDialog makeRequestDialog(CharSequence title, CharSequence message,
    CharSequence titleButtonYes, CharSequence titleButtonNo) {
    AlertDialog.Builder requestDialog = new AlertDialog.Builder(this);
    requestDialog.setTitle(title);
    requestDialog.setMessage(message);
    requestDialog.setPositiveButton(titleButtonYes, new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            textView.setText("5초 후에 결과 표시됨.");
        }
    });
}
```

handler.postDelayed(new Runnable() { —→ 핸들러의 postDelayed() 메서드 호출하기

```
@Override
public void run() {
    textView.setText("요청 완료됨.");
}
}, 5000);
})
```



일정시간 후에 실행하기 – 핸들러 사용하기

- 네트워킹을 위한 별도의 스레드를 만들지 않고 문제를 해결하는 방법

- 핸들러의 특정 메소드를 이용하여 일정 시간 후에 실행되도록 함

- 핸들러

- 핸들러는 메시지 큐를 사용하므로 메시지들을 순서대로 처리하지만 메시지를 넣을 때 시간을 지정하면 원하는 시간에 메시지를 처리하도록 만들 수 있으므로 일정 시간 후에 실행시킬 때 유용하게 사용됨.

- 시간을 지정하는 경우

- 핸들러의 sendMessage() 메소드와 유사한 이름을 가진 다음과 같은 두 가지 메소드를 사용할 수 있음

- API

- public boolean sendMessageAtTime(Message msg, long uptimeMillis)
 - public boolean sendMessageDelayed(Message msg, long delayMillis)

- 첫 번째 메소드는 메시지를 보낼 때 시간을 지정할 수 있으며, 두 번째 메소드는 메시지가 일정 시간이 지난 후 실행되도록 설정할 수 있음
- Runnable 객체를 실행하는 post() 메소드의 경우에도 postAtTime()과 postDelayed() 메소드가 있어 동일한 기능을 수행함



3.

스레드로 메시지 전송하기

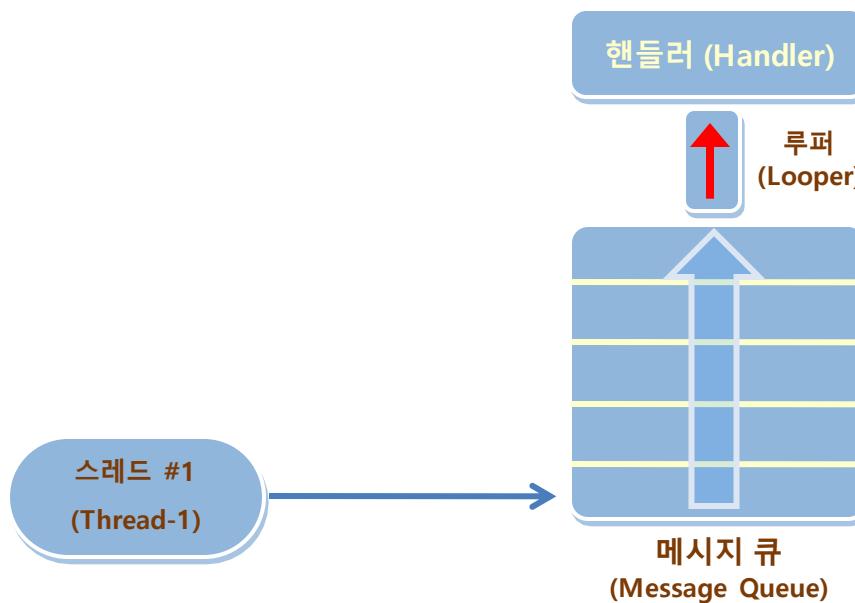


스레드로 메시지 전송하기

- **핸들러의 기능**
 - 새로 만든 스레드에서 메인 스레드로 메시지를 전달하는 것임
- **스레드의 작업 결과물을 메시지로 만들어 전달하는 이유**
 - 별도의 스레드에서 메인 스레드가 관리하는 UI 객체에 직접 접근할 수 없기 때문임
- **메인 스레드에서 별도의 스레드로 메시지를 전달하는 방법이 필요한 경우**
 - 일반적으로 변수 선언을 통해 데이터를 전달하는 것이 가장 쉬운 방법임
 - 핸들러가 사용하는 메시지 큐를 이용하여 순차적으로 메시지를 실행하는 방식이 필요한 경우가 발생함
 - 특히 별도의 스레드가 동일한 객체에 접근할 때 다른 스레드들이 동시에 메소드를 호출하는 경우가 있을 수 있으므로 메시지 큐를 이용한 접근 방식에 대해 이해가 필요함



루퍼 이해하기



- **루퍼**

- 무한 루프 방식을 이용해 메시지 큐에 들어오는 메시지를 지속적으로 보면서 하나씩 처리함

[루퍼를 이용한 메시지 처리]



화면 레이아웃 만들기

The screenshot shows the Android Studio Layout Editor interface. The top bar displays 'activity_main.xml' and 'MainActivity.java'. The left side features a 'Palette' with categories like Common, Text, Buttons, Widgets, Layouts, Containers, Google, and Legacy. A 'Component Tree' panel lists the current components: a button labeled '스레드로 보내기', a textView labeled '결과', and an editText labeled '(Plain Text)'. The main area is a 'ConstraintLayout' containing a white rectangular view. Inside this view, there is a 'textView' with the text '결과' and a 'button'. The 'button' has a tooltip '스레드로 보내기'. The 'Attributes' panel on the right shows properties for the textView, including 'text' set to '결과', 'textSize' set to '30sp', and 'layout_width' and 'layout_height' both set to 'wrap_content'. Other properties listed include 'Constraints', 'Layout_Margin', 'Padding', 'Theme', and various clickability and text-related settings.



메인 액티비티 코드 만들기

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    editText = findViewById(R.id.editText);  
    textView = findViewById(R.id.textView);  
  
    Button button = findViewById(R.id.button);  
    button.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            String input = editText.getText().toString();  
            Message message = Message.obtain();  
            message.obj = input;  
  
            thread.processHandler.sendMessage(message); —→ ❶ 새로 만든 스레드 안에 있는 핸들러로  
        }  
    });  
  
    thread = new ProcessThread();  
}
```

메시지 전송하기



메인 액티비티 코드 만들기 (계속)

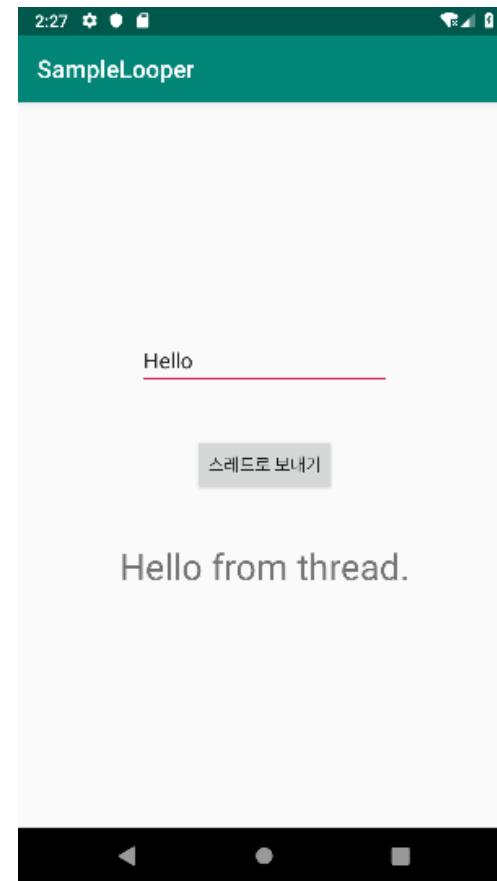
```
class ProcessThread extends Thread {  
    ProcessHandler processHandler = new ProcessHandler();  
  
    public void run() {  
        Looper.prepare();  
        Looper.loop();  
    }  
  
    class ProcessHandler extends Handler {  
  
        public void handleMessage(Message msg) {  
            final String output = msg.obj + " from thread.";  
        }  
    }  
}
```

② 새로 만든 스레드 안에서 전달받은
메시지 처리하기

```
handler.post(new Runnable() {  
    @Override  
    public void run() {  
        textView.setText(output);  
    }  
});  
}  
}
```



실행 화면



[루퍼를 이용한 스레드 간 문자열 전달]

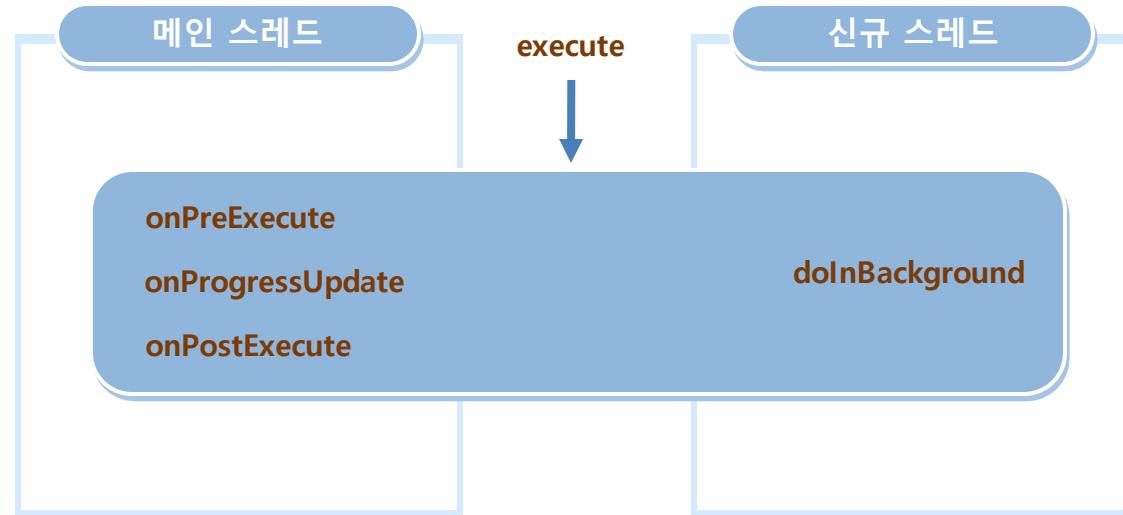


4.

AsyncTask 사용하기



AsyncTask 사용하기



[**AsyncTask**를 이용한 작업 수행 방식]

• **AsyncTask**

- 백그라운드 작업을 좀 더 쉽고 간단하게 하고 싶다면 **AsyncTask** 클래스를 사용할 수 있음
- **AsyncTask** 객체를 만들고 **execute()** 메소드를 실행하면 이 객체는 정의된 백그라운드 작업을 수행하고 필요한 경우 그 결과를 메인 스레드에서 실행하므로 UI 객체에 접근하는데 문제가 없게 됨



AsyncTask의 주요 메소드

메소드 이름	설명
doInBackground	<ul style="list-style-type: none">- 새로 만든 스레드에서 백그라운드 작업 수행- execute() 메소드를 호출할 때 사용된 파라미터를 배열로 전달받음
onPreExecute	<ul style="list-style-type: none">- 백그라운드 작업 수행 전 호출- 메인 스레드에서 실행되며 초기화 작업에 사용
onProgressUpdate	<ul style="list-style-type: none">- 백그라운드 작업 진행 상태를 표시하기 위해 호출- 작업 수행 중간 중간에 UI 객체에 접근하는 경우 사용- 이 메소드가 호출되도록 하려면 백그라운드 작업 중간에 publishProgress() 메소드를 호출
onPostExecute	<ul style="list-style-type: none">- 백그라운드 작업이 끝난 후 호출- 메인 스레드에서 실행되며 메모리 리소스를 해제하는 등의 작업에 사용- 백그라운드 작업의 결과는 Result 타입의 파라미터로 전달

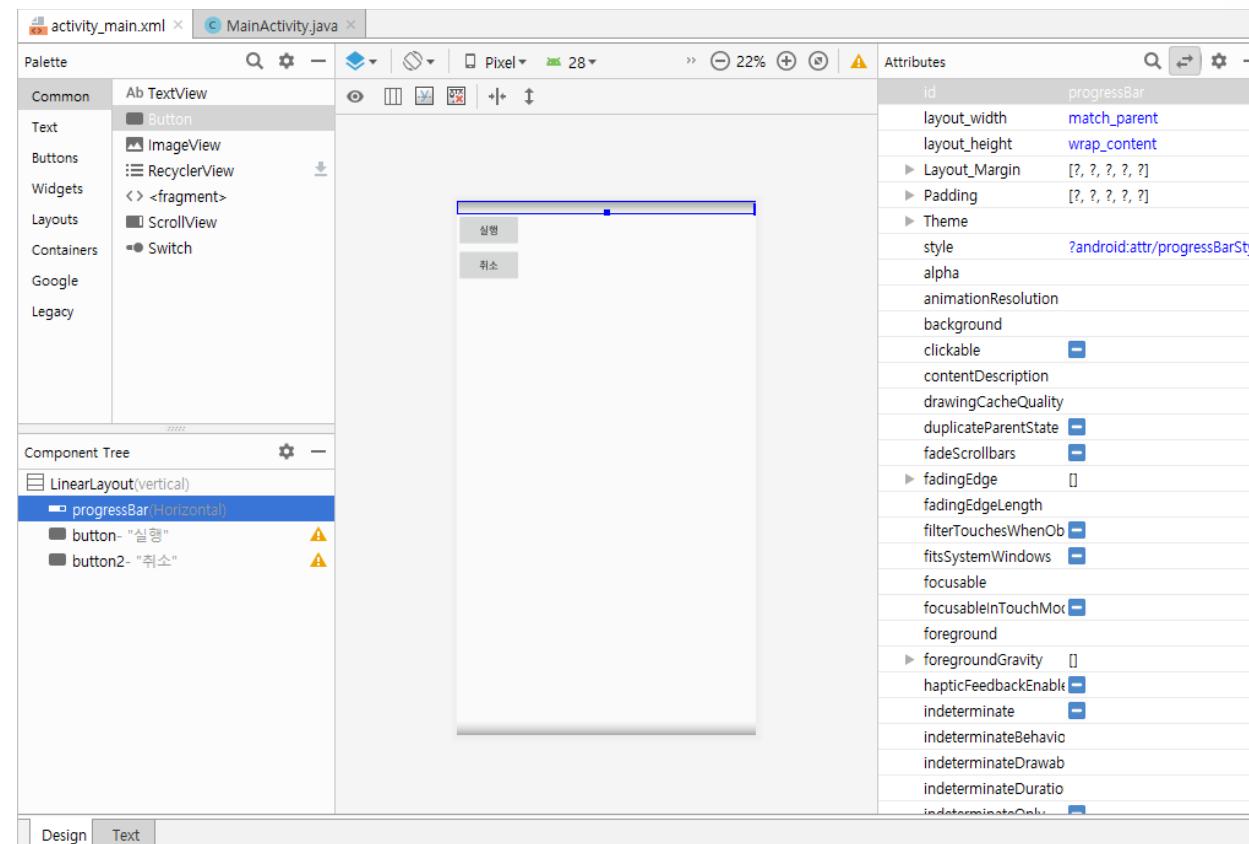


Async 메소드 사용하기

- **AsyncTask** 객체의 `cancel()` 메소드
 - 작업을 취소함, 이 메소드를 통해 작업을 취소했을 경우에는 `onCancelled()` 메소드가 호출됨
- **AsyncTask** 객체의 `getStatus()` 메소드
 - 작업의 진행 상황을 확인함
 - 메소드를 호출했을 때 리턴되는 `AsyncTask.Status` 객체는 상태를 표현함
 - 각각의 상태는 `PENDING`, `RUNNING`, `FINISHED`로 구분됨
- **PENDING**
 - 작업이 아직 시작되지 않았다는 것을 의미함
- **RUNNING**
 - 실행 중임을 의미함
- **FINISHED**
 - 종료되었음을 의미함



화면 레이아웃 구성하기



[AsyncTask를 이용해 진행 상태를 표시하는 화면 구성]

- 이 화면의 버튼을 클릭하면 별도의 스레드에서 값을 1씩 증가시키도록 함
- 100밀리 초마다 한 번씩 값을 증가시키므로 프로그래스바의 최대값으로 지정된 100이 될 때까지 10초가 걸리게 됨



메인 액티비티 코드 만들기

- 백그라운드 작업을 수행할 클래스는 BackgroundTask라는 이름의 클래스로 정의하고 AsyncTask 클래스를 상속받음
- onPreExecute() 메소드는 초기화 단계에서 사용되므로 값을 저장하기 위해 메인 액티비티에 정의한 value 변수의 값을 0으로 초기화하고 프로그레스바의 값도 0으로 만들어 줌
- dolnBackground() 메소드는 주 작업을 실행하는데 사용되므로 while 구문을 이용해 value의 값을 하나씩 증가시키도록 함

```
Button button = findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        task = new BackgroundTask();
        task.execute();
```

↗ ① 태스크 객체 만들어 실행하기

```
    }
});
```



AsyncTask에서 상속한 클래스 정의

```
class BackgroundTask extends AsyncTask<Integer , Integer , Integer> {  
    protected void onPreExecute() {  
        value = 0;  
        progressBar.setProgress(value);  
    }  
  
    protected Integer doInBackground(Integer ... values) {  
        while (isCancelled() == false) {  
            value++;  
            if (value >= 100) {  
                break;  
            } else {  
                publishProgress(value);  
            }  
  
            try {  
                Thread.sleep(100);  
            } catch (InterruptedException ex) {}  
        }  
  
        return value;  
    }  
}
```

- ② 태스크 객체 안에서 백그라운드 작업
수행하도록 하기



AsyncTask에서 상속한 클래스 정의 (계속)

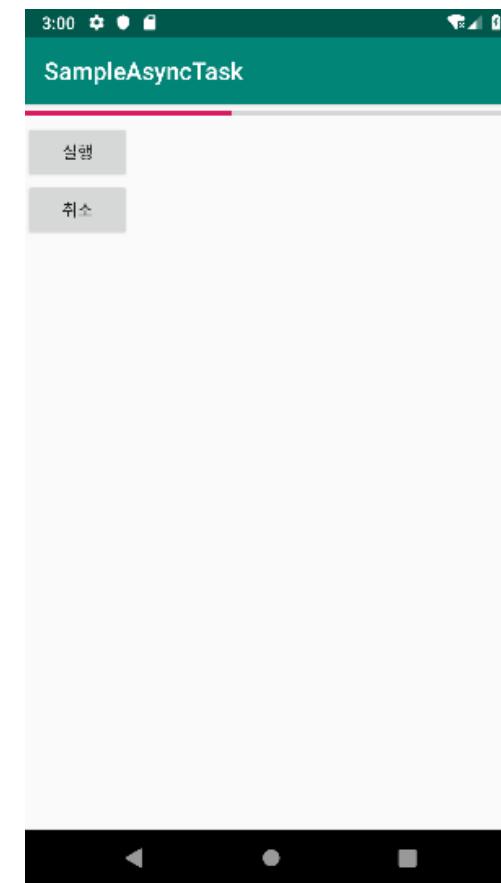
```
protected void onProgressUpdate(Integer ... values) {
    progressBar.setProgress(values[0].intValue());
}

protected void onPostExecute(Integer result) {
    progressBar.setProgress(0);
}

protected void onCancelled() {
    progressBar.setProgress(0);
}
}
```



앱 실행



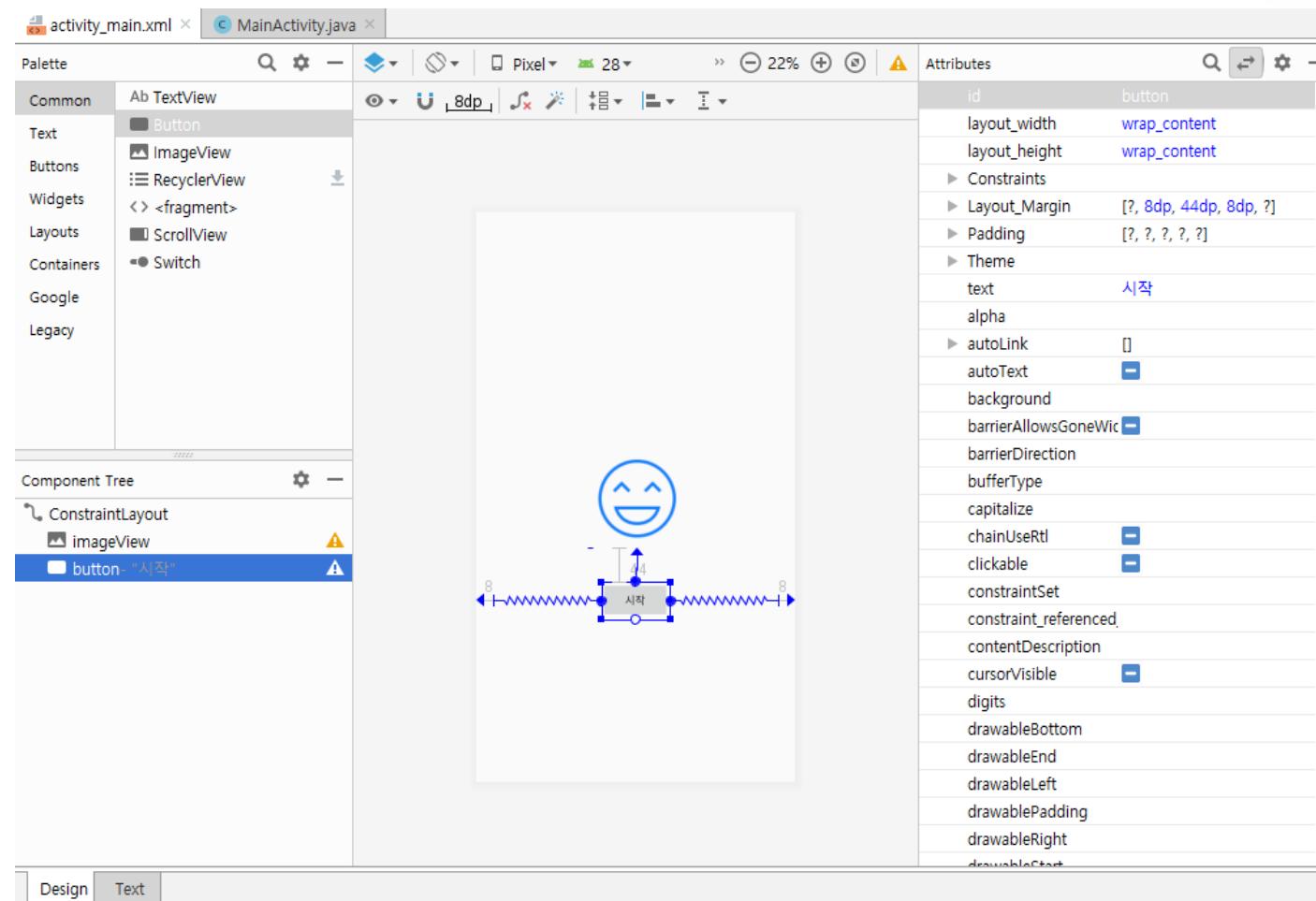


5.

스레드로 애니메이션 만들기



레이아웃 화면 만들기





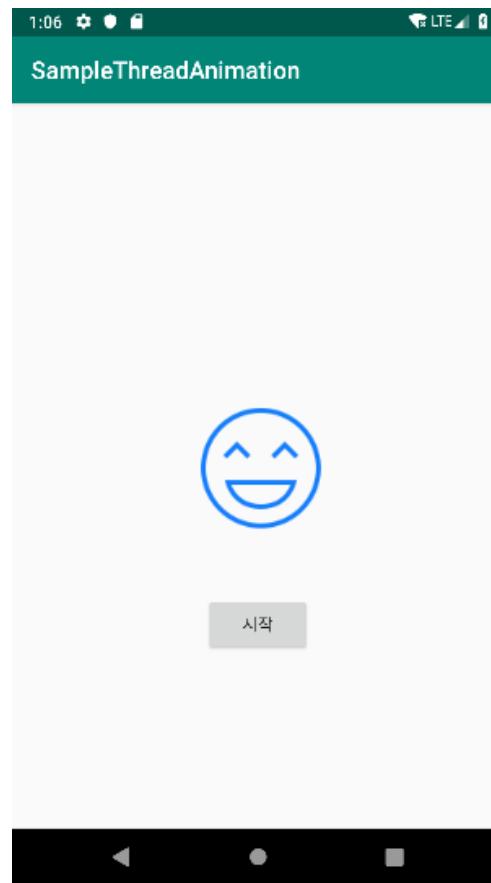
스레드 정의

```
class AnimThread extends Thread {  
    public void run() {  
        int index = 0;  
        for (int i = 0; i < 100; i++) {  
            final Drawable drawable = drawableList.get(index);  
            index += 1;  
            if (index > 4) {  
                index = 0;  
            }  
  
            handler.post(new Runnable() {  
                @Override  
                public void run() {  
                    imageView.setImageDrawable(drawable);  
                }  
            });  
  
            try {  
                Thread.sleep(1000);  
            } catch (Exception e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

② 화면에 이미지를 보여주기 위해 핸들러의 post() 메서드 호출하기



앱 실행





기본적인 네트워킹 방법에 대한 이해



목 차

- 1 네트워킹이란?
- 2 소켓 사용하기
- 3 웹으로 요청하기
- 4 Volley 사용하기
- 5 JSON 데이터 다루기
- 6 영화 정보 가져와 보여주기



네트워킹에 대해 어떻게 생각해야 하나요?

- 네트워킹이란?



소켓이나 웹 연결은 어떻게 하나요?

- 소켓 사용하기
- 웹으로 요청하기



라이브러리를 이용해 간단하게 가져오는 방법도 있나요?

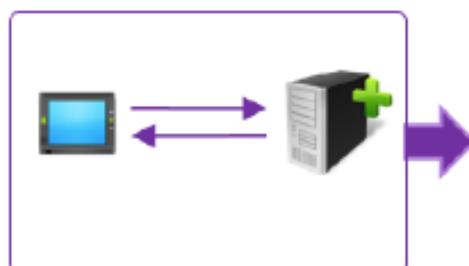
- Volley 사용하기
- JSON 데이터 다루기



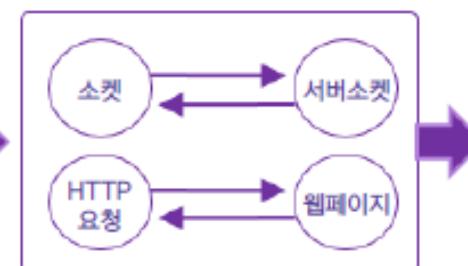
영화 정보를 가져와 보여주고 싶어요

- 영화 정보 가져와 보여주기

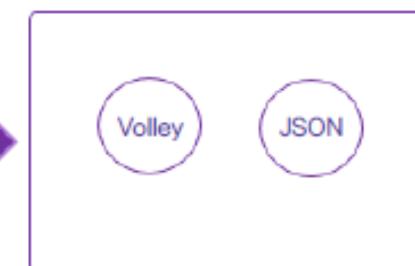
네트워킹이란?



소켓과 웹으로 연결하기



Volley와 JSON 사용하기





1.

네트워킹이란?

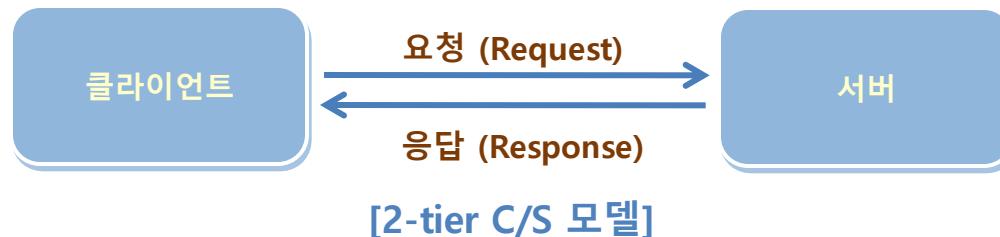


네트워킹이란?

■ 원격지의 서버를 연결하는 방식

- 2-tier C/S 모델

- 클라이언트와 서버가 일대일로 연결하는 방식



- 3-tier 모델

- 서버를 좀 더 유연하게 구성
- 응용 서버와 데이터 서버로 구성하는 경우, 데이터베이스를 분리시킴





소켓 사용하기

■ 네트워킹에 대한 이해

- TCP/IP 수준의 통신 방식을 제공하는 소켓을 이용해 서버에 연결해 보면 이해하기 쉬움
- 일반적인 프로그래밍에서는 대부분 TCP 연결 사용
- 비연결성(stateless) 특성으로 인해 실시간으로 데이터를 처리하는 애플리케이션의 경우, 응답 속도를 높이기 위해 HTTP보다 소켓 연결 선호

■ 소켓 연결 방식

- 안드로이드에서는 표준 자바의 소켓을 그대로 사용할 수 있음
- 서버쪽에는 서버소켓을 만들어 실행함 (포트 지정)
- 클라이언트쪽에서는 소켓을 만들어 서버소켓으로 연결함 (IP와 포트 지정)
- Stream 객체를 이용해 데이터를 보내거나 받을 수 있음



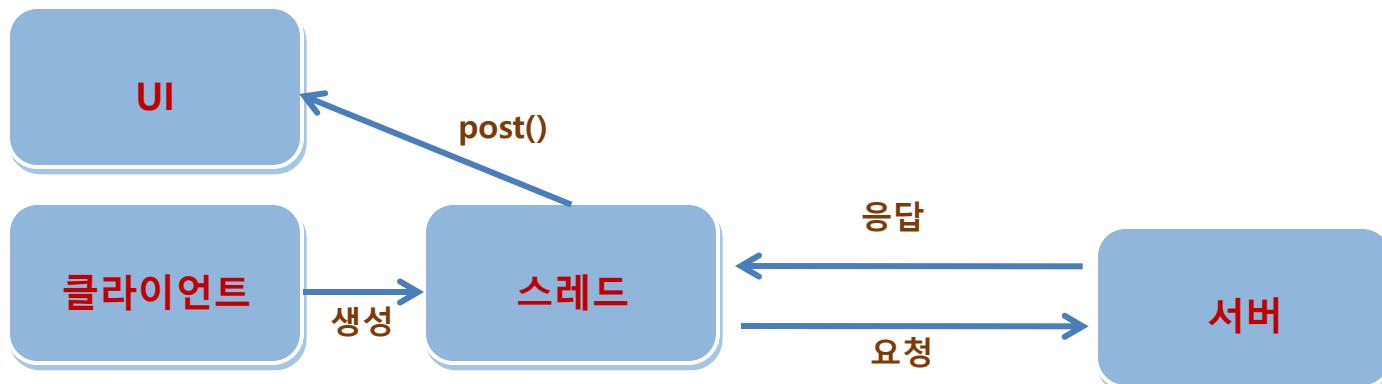
네트워킹 사용 시 주의할 점

■ 네트워킹을 사용할 때는 반드시 스레드 사용

- 최신 버전의 안드로이드에서는 네트워킹을 사용할 때는 반드시 스레드를 사용하도록 변경되었음 (이전에는 스레드 없이도 가능했음)

■ 스레드를 사용하므로 UI 업데이트를 위해서는 반드시 핸들러 사용

- 네트워킹을 위해 새로 만든 스레드 안에서 그 결과를 보여주기 위해 UI 업데이트를 하는 경우 스레드 부분에서 공부한 바와 같이 핸들러를 사용해야 함
- 가장 간단한 방법으로 post() 메소드 사용 권장



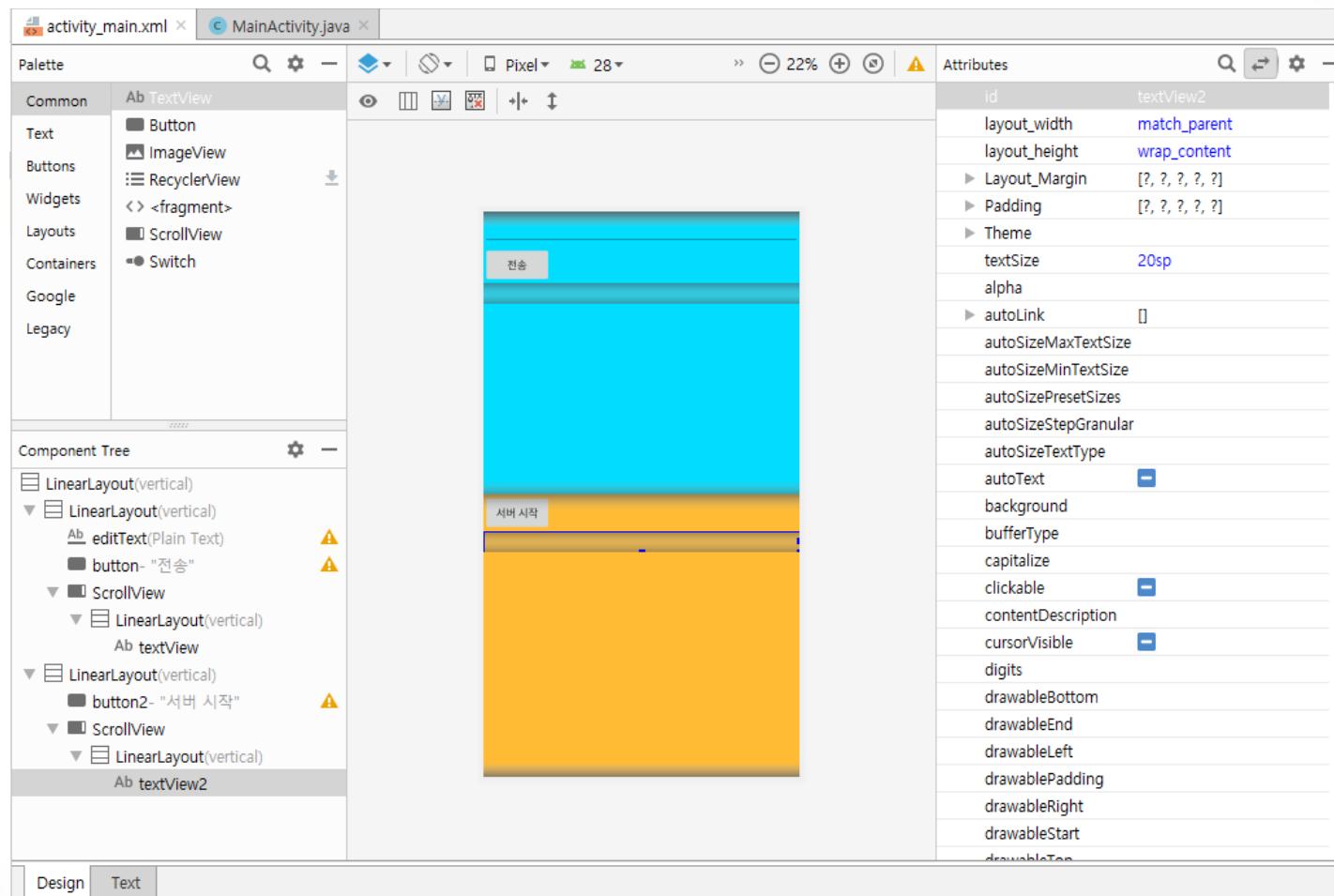


2.

소켓 사용하기



화면 레이아웃 만들기





버튼 누르면 스레드 안에서 send

```
@Override  
public void onClick(View v) {  
    final String data = editText.getText().toString();  
    new Thread(new Runnable() {  
        @Override  
        public void run() {  
            send(data);  
        }  
    }).start();  
}
```

① 스레드 안에서 send() 메서드 호출하기



버튼 누르면 스레드 안에서 startServer

```
@Override  
public void onClick(View v) {  
    new Thread(new Runnable() {  
        @Override  
        public void run() {  
            startServer();  
        }  
    }).start();  
}
```



② 스레드 안에서 startServer() 메서드 호출하기



send 메서드 정의

```
public void send(String data) {  
    try {  
        int portNumber = 5001;  
        Socket sock = new Socket("localhost", portNumber);  
        printClientLog("소켓 연결함."); } ① 소켓 객체 만들기  
  
        ObjectOutputStream outstream = new ObjectOutputStream(sock.getOutputStream());  
        outstream.writeObject(data);  
        outstream.flush();  
        printClientLog("데이터 전송함."); } ② 소켓 객체로  
        데이터 보내기  
  
        ObjectInputStream instream = new ObjectInputStream(sock.getInputStream());  
        printClientLog("서버로부터 받음: " + instream.readObject());  
        sock.close();  
    } catch (Exception ex) {  
        ex.printStackTrace();  
    }  
}
```



startServer 메서드 정의

```
public void startServer() {  
    try {  
        int portNumber = 5001;  
  
        ServerSocket server = new ServerSocket(portNumber);  
        printServerLog("서버 시작함: " + portNumber);  
    } ① 소켓 서버 객체 만들기
```

```
    while (true) {  
        Socket sock = server.accept();  
        InetAddress clientHost = sock.getLocalAddress();  
        int clientPort = sock.getPort();  
        printServerLog("클라이언트 연결됨: " + clientHost + " : " + clientPort);  
        ② 클라이언트가 접  
        속했을 때 만들어  
        지는 소켓 객체  
        참조하기  
  
        ObjectInputStream instream = new ObjectInputStream(sock.getInputStream());  
        Object obj = instream.readObject();  
        printServerLog("데이터 받음: " + obj);  
  
        ObjectOutputStream outstream = new ObjectOutputStream(sock.getOutputStream());  
        outstream.writeObject(obj + " from Server.");  
        outstream.flush();  
        printServerLog("데이터 보냄.");  
  
        sock.close();  
    }  
} catch (Exception ex) {  
    ex.printStackTrace();  
}  
}
```



INTERNET 권한 추가

참조파일 SampleSocket>/app/manifests/AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="org.techtown.socket">  
  
    <uses-permission android:name="android.permission.INTERNET"/>
```

중략...



앱 실행





3.

웹으로 요청하기



■ HTTP 연결 방식

- 예전 휴대 단말은 데이터 통신의 송수신 속도가 느려서 소켓으로 연결하거나 웹 페이지를 보기 위해서는 많이 기다려야 함
- 비연결성(stateless)인 HTTP 프로토콜은 새로 연결을 만드는 데 따른 지연 시간이 길게 발생
- 최근 스마트폰 및 무선 네트워크 환경이 좋아져서 HTTP 프로토콜을 이용한 웹의 사용이 자연스러울 뿐만 아니라 일반 웹 사이트를 보는 풀 브라우징(full browsing)도 가능함
- 자바에서 사용하던 HTTP 관련 클래스를 그대로 사용할 수 있음



URLConnection 클래스

■ URLConnection 객체

[API]

public URLConnection openConnection()

* HttpURLConnection 객체로 형변환(casting)하여 사용

■ 요청 관련 메소드

[API]

public void setRequestMethod(String method)

public void setRequestProperty(String field, String newValue)



화면 레이아웃 만들기

The screenshot shows the Android Studio Layout Editor interface. At the top, there are tabs for `activity_main.xml` and `MainActivity.java`. The main area consists of three panels: a **Palette** on the left containing various UI components like TextView, Button, ImageView, etc.; a **Component Tree** panel below it showing the hierarchical structure of the layout; and an **Attributes** panel on the right listing XML attributes for the selected component. The layout itself is a vertical `LinearLayout` containing an `editText` field and a `button` labeled "요청하기". A nested `ScrollView` contains another `LinearLayout` with a `textView` component.

Palette:

- Common
- Text
- Buttons
- Widgets
- Layouts
- Containers
- Google
- Legacy

Attributes:

<code>id</code>	<code>textView</code>
<code>layout_width</code>	<code>match_parent</code>
<code>layout_height</code>	<code>wrap_content</code>
<code>Layout_Margin</code>	[?, ?, ?, ?, ?]
<code>Padding</code>	[?, ?, ?, ?, ?]
<code>Theme</code>	
<code>textSize</code>	<code>20sp</code>
<code>alpha</code>	
<code>autoLink</code>	
<code>autoSizeMaxTextSize</code>	
<code>autoSizeMinTextSize</code>	
<code>autoSizePresetSizes</code>	
<code>autoSizeStepGranular</code>	
<code>autoSizeTextType</code>	
<code>autoText</code>	
<code>background</code>	
<code>bufferType</code>	
<code>capitalize</code>	
<code>clickable</code>	
<code>contentDescription</code>	
<code>cursorVisible</code>	
<code>digits</code>	
<code>drawableBottom</code>	
<code>drawableEnd</code>	
<code>drawableLeft</code>	
<code>drawablePadding</code>	
<code>drawableRight</code>	
<code>drawableStart</code>	
<code>drawableTop</code>	

Component Tree:

- `LinearLayout`(vertical)
 - `editText`(Plain Text)
 - `button`- "요청하기"
- `ScrollView`
 - `LinearLayout`(vertical)
 - `textView`

Design Tab



버튼 눌렀을 때 호출할 request 메서드 정의

```
public void request(String urlStr) {  
    StringBuilder output = new StringBuilder();  
    try {  
        URL url = new URL(urlStr);  
  
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();  
        if (conn != null) {  
            conn.setConnectTimeout(10000);  
            conn.setRequestMethod("GET");  
            conn.setDoInput(true);  
  
            int resCode = conn.getResponseCode();  
            BufferedReader reader = new BufferedReader(new InputStreamReader(conn.getInputStream()));  
            String line = null;  
            while (true) {  
                line = reader.readLine();  
                if (line == null) {  
                    break;  
                }  
  
                output.append(line + "\n");  
            }  
            reader.close();  
            conn.disconnect();  
        }  
    } catch (Exception e) {}  
}
```

① HttpURLConnection 객체 만들기

② 입력 데이터를 받기 위한 Reader 객체 생성하기



인터넷 권한과 usesCleartextTraffic 속성 추가

참조파일 SampleHttp>/app/manifests/AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="org.techtown.http">  
  
    <uses-permission android:name="android.permission.INTERNET"/>  
  
    <application  
        android:usesCleartextTraffic="true"  
    </application>
```

중략...



앱 실행하고 사이트 주소로 요청

```
http://www.kobis.or.kr/kobisopenapi/webservice/rest/boxoffice/searchDailyBoxOfficeList.json?key=430156241533f1d058c603178cc3ca0e&targetDt=20120101
```



HttpURLConnection으로 웹페이지 요청 화면



4.

Volley 사용하기



Volley를 이용한 요청과 응답 처리





Volley 라이브러리 추가

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

중략...

```
dependencies {
```

중략...

```
    implementation 'com.android.volley:volley:1.1.0'
```

```
}
```



인터넷 권한 추가

참조파일 SampleRequest>/app/manifests/AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="org.techtown.request">  
  
    <uses-permission android:name="android.permission.INTERNET"/>  
    <application  
        android:usesCleartextTraffic="true"  
        ...>
```

중략...



RequestQueue 객체 생성

참조파일 SampleRequest>/app/java/org.techtown.request/MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    EditText editText;
    TextView textView;

    static RequestQueue requestQueue;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        editText = findViewById(R.id.editText);
        textView = findViewById(R.id.textView);

        Button button = findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                makeRequest();
            }
        });
    }

    if (requestQueue == null) {
        requestQueue = Volley.newRequestQueue(getApplicationContext());
    }
}
```



① RequestQueue
객체 생성하기



요청 메서드 추가

```
public void makeRequest() {
    String url = editText.getText().toString();

    StringRequest request = new StringRequest(Request.Method.GET, url, [
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                println("응답-> " + response);
            }
        },
        new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                println("에러-> " + error.getMessage());
            }
        }
    ) {
        @Override
        protected Map<String, String> getParams() throws AuthFailureError {
            Map<String, String> params = new HashMap<String, String>();

            return params;
        }
    };
}
```

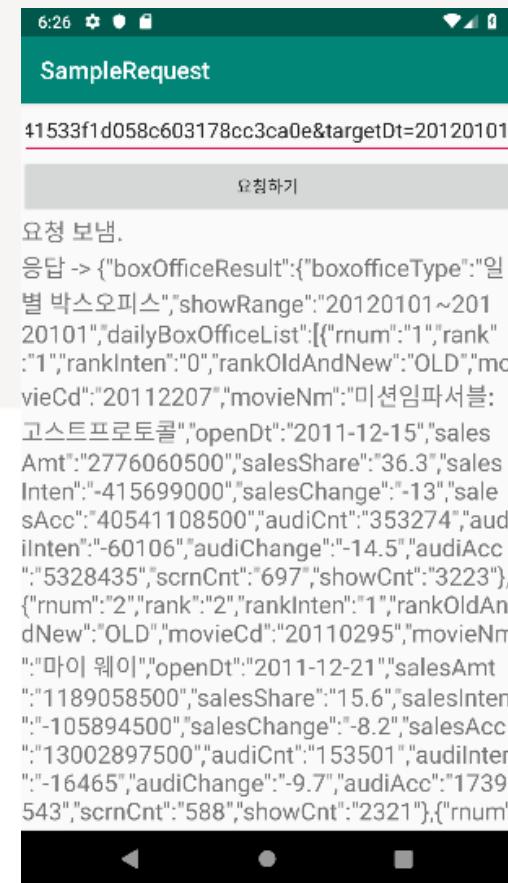
② 요청을 보내기 위한
StringRequest
객체 생성하기



요청 메서드 추가

```
request.setShouldCache(false);
requestQueue.add(request);
println("요청 보냄.");
}

public void println(String data) {
    textView.append(data + "\n");
}
```



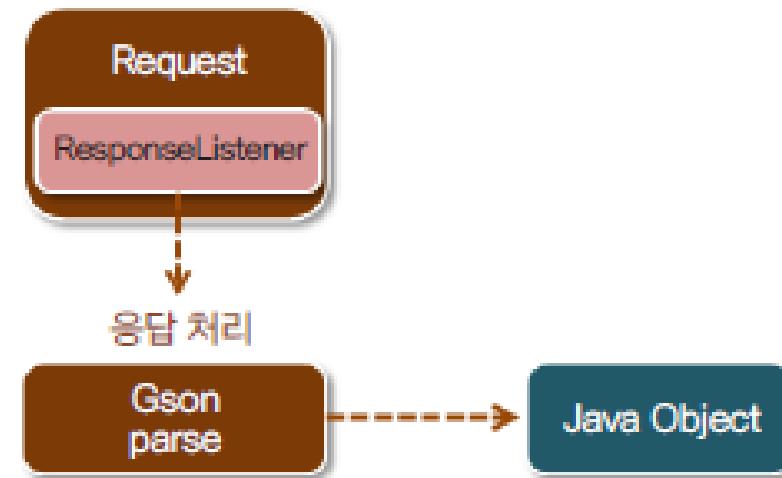


5.

JSON 데이터 다루기



Gson의 역할





Gson 라이브러리 추가

참조파일 SampleRequest2>/Gradle Scripts/build.gradle(Module:app)

중략...

```
dependencies {
```

중략...

```
    implementation 'com.android.volley:volley:1.1.0'
```

```
    implementation 'com.google.code.gson:gson:2.8.2'
```

```
}
```



응답 JSON 객체화를 위한 클래스 정의

참조파일 SampleRequest2>/app/java/org.techtown.request/MovieList.java

```
public class MovieList {  
    MovieListResult boxOfficeResult;  
}
```

참조파일 SampleRequest2>/app/java/org.techtown.request/MovieListResult.java

```
public class MovieListResult {  
  
    String boxofficeType;  
    String showRange;  
  
    ArrayList<Movie> dailyBoxOfficeList = new ArrayList<Movie>();  
}
```



응답 JSON 객체화를 위한 클래스 정의

참조파일 SampleRequest2>/app/java/org.techtown.request/Movie.java

```
public class Movie {  
  
    String rnum;  
    String rank;  
    String rankInten;  
    String rankOldAndNew;  
    String movieCd;  
    String movieNm;  
    String openDt;  
    String salesAmt;  
    String salesShare;  
    String salesInten;  
    String salesChange;  
    String salesAcc;  
    String audiCnt;  
    String audiInten;  
    String audiChange;  
    String audiAcc;  
    String scrnCnt;  
    String showCnt;  
  
}
```



Gson을 이용해 JSON 변환

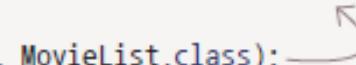
참조파일 SampleRequest2>/app/java/org.techtown.request/MainActivity.java

중략...

```
new Response.Listener<String>() {  
    @Override  
    public void onResponse(String response) {  
        println("응답-> " + response);  
  
        processResponse(response);  
    }  
  
    public void processResponse(String response) {  
        Gson gson = new Gson();  
        MovieList movieList = gson.fromJson(response, MovieList.class);  
        println("영화 정보의 수: " + movieList.boxOfficeResult.dailyBoxOfficeList.size());  
    }  
}
```

중략...

JSON 문자열을 MovieList 객체로 변환하기





6.

영화 정보 가져와 보여주기



화면 레이아웃 만들기

The screenshot shows the Android Studio interface for creating an Android application. The top navigation bar includes tabs for 'app', 'activity_main.xml', 'movie_item.xml', 'AndroidManifest.xml', 'strings.xml', 'MainActivity.java', and 'MovieAdapter.java'. The main area is divided into three panels: 'Palette' on the left, 'Design' preview in the center, and 'Attributes' on the right.

Palette: The 'Common' category is selected, showing items like Button, ImageView, RecyclerView, ScrollView, and Switch.

Design View: A vertical LinearLayout is displayed. Inside it, there is an editText with the URL 'http://www.kobis.or.kr/kobisopenapi/webservice/rs' and a button labeled '요청하기'.

Attributes Panel: The 'editText' attributes are listed:

id	editText
layout_width	match_parent
layout_height	wrap_content
Layout_Margin	[?, ?, ?, ?, ?]
Padding	[?, ?, ?, ?, ?]
Theme	
ems	10
inputType	[text]
text	http://www.kobis.or.kr/kobi
alpha	
autoLink	[]
autoText	-
background	
bufferType	
capitalize	
clickable	-
contentDescription	
cursorVisible	-
digits	
drawableBottom	
drawableEnd	
drawableLeft	
drawablePadding	
drawableRight	
drawableStart	
drawableTop	
drawingCacheQuality	
duplicateParentState	-
editable	-



리싸이클러뷰를 위한 어댑터 정의

참조파일 SampleMovie>/app/java/org.techtown.movie/MovieAdapter.java

```
public class MovieAdapter {

    static class ViewHolder extends RecyclerView.ViewHolder {
        TextView textView;
        TextView textView2;

        public ViewHolder(View itemView) {
            super(itemView);

            textView = itemView.findViewById(R.id.textView);
            textView2 = itemView.findViewById(R.id.textView2);
        }

        public void setItem(Movie item) {
            textView.setText(item.movieNm);
            textView2.setText(itemaudiCnt + " 명");
        }
    }
}
```



리싸이클러뷰의 각 아이템을 위한 XML 레이아웃 정의

참조파일 SampleMovie>/app/res/layout/movie_item.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="wrap_content"
    android:orientation="vertical">

    <android.support.v7.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="4dp"
        android:layout_marginBottom="4dp"
        app:cardBackgroundColor="#FFFFFF"
        app:cardCornerRadius="10dp"
        app:cardElevation="5dp" >

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">
```



어댑터에 코드 추가

참조파일 SampleMovie>/app/java/org.techtown.movie/MovieAdapter.java

```
public class MovieAdapter extends RecyclerView.Adapter<MovieAdapter.ViewHolder> {
    ArrayList<Movie> items = new ArrayList<Movie>();

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int viewType) {
        LayoutInflator inflater = LayoutInflator.from(viewGroup.getContext());
        View itemView = inflater.inflate(R.layout.movie_item, viewGroup, false);

        return new ViewHolder(itemView);
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolder viewHolder, int position) {
        Movie item = items.get(position);
        viewHolder.setItem(item);
    }
}
```



메인 액티비티에 코드 추가

참조파일 SampleMovie>/app/java/org.techtown.movie/MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
    EditText editText;  
    TextView textView;  
  
    RecyclerView recyclerView;  
    MovieAdapter adapter;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        editText = findViewById(R.id.editText);  
        textView = findViewById(R.id.textView);  
  
        recyclerView = findViewById(R.id.recyclerView); → ❶ XML 레이아웃에 정의한 리사이클러뷰  
                                         객체 참조하기  
        LinearLayoutManager layoutManager =  
            new LinearLayoutManager(this, LinearLayoutManager.VERTICAL, false);  
        recyclerView.setLayoutManager(layoutManager);  
  
        adapter = new MovieAdapter();  
        recyclerView.setAdapter(adapter); → ❷ 리사이클러뷰에 어댑터 설정하기  
    }  
}
```



Volley와 Gson 라이브러리 추가

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

중략...

```
dependencies {  
    중략...  
  
    implementation 'com.android.volley:volley:1.1.0'  
    implementation 'com.google.code.gson:gson:2.8.2'  
}
```



인터넷 권한 추가

참조파일 SampleMovie>/app/manifests/AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="org.techtown.movie">
```

```
    <uses-permission android:name="android.permission.INTERNET"/>
```

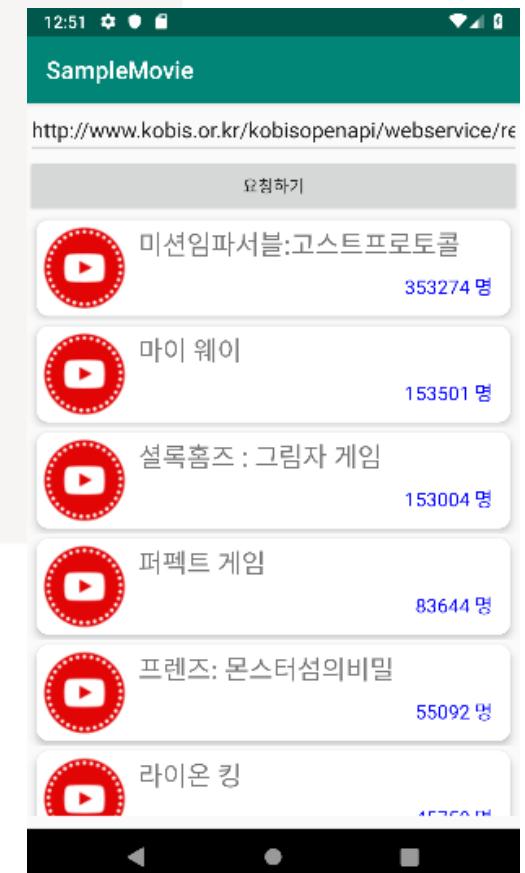
```
    <application  
        android:usesCleartextTraffic="true"
```

중략...



버튼 클릭 시 요청 보내고 응답 처리

```
public void processResponse(String response) {  
    Gson gson = new Gson();  
    MovieList movieList = gson.fromJson(response, MovieList.class); → ② 응답받은 JSON 문자열을  
                                                MovieList로 변환하기  
    println("영화 정보 수: " + movieList.boxOfficeResult.dailyBoxOfficeList.size());  
  
    for (int i = 0; i < movieList.boxOfficeResult.dailyBoxOfficeList.size(); i++) {  
        Movie movie = movieList.boxOfficeResult.dailyBoxOfficeList.get(i);  
  
        adapter.addItem(movie);  
    }  
  
    adapter.notifyDataSetChanged();  
}
```





데이터베이스에 대한 이해와 데이터 저장 및 조회 실습



- 1 모바일 데이터베이스란?
- 2 데이터베이스와 테이블 만들기
- 3 헬퍼 클래스로 업그레이드 지원하기
- 4 데이터 조회하기
- 5 내용 제공자 이해하기
- 6 앨범과 연락처 조회하기



① 스마트폰에서도 데이터베이스를 사용할 수 있나요?

- 모바일 데이터베이스



데이터베이스는 어떻게 만들면 되나요?

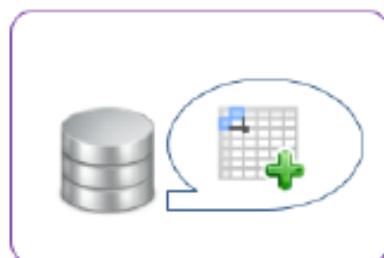
- 데이터베이스 만들기
- 테이블 만들기



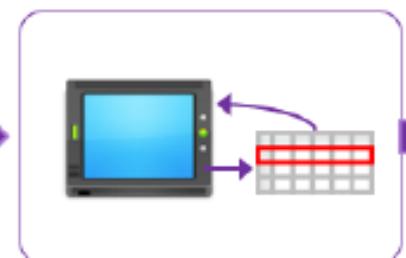
테이블의 데이터는 어떻게 가져올 수 있나요?

- 데이터 조회하기

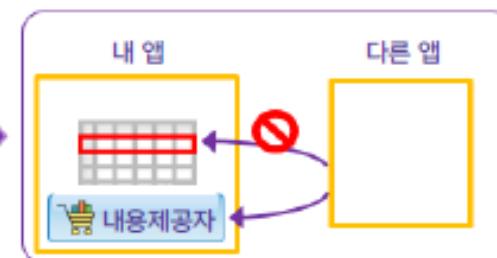
데이터베이스 만들기



데이터 조회하기



내용 제공자 이해하기





1.

모바일 데이터베이스란?



모바일 데이터베이스란?

■ 안드로이드에서 데이터를 저장하는 대표적인 방법

- 설정 정보
- 파일 사용
- **데이터베이스 → 많은 데이터를 체계적으로 관리**

■ 데이터베이스

- 여러 개의 테이블을 담고 있는 하나의 그릇 역할

■ 데이터베이스를 만드는 가장 간단한 방법

- Context 클래스에 정의된 `openOrCreateDatabase()` 메소드를 사용
- 애플리케이션에서 기본적으로 사용하는 Activity 클래스가 Context를 상속한
것이므로 액티비티 안에서 데이터베이스 생성 가능



[데이터베이스 활용 순서]



2.

데이터베이스와 테이블 만들기



데이터베이스와 테이블 만들기

■ 데이터베이스를 열거나 삭제할 수 있는 메소드

```
public abstract SQLiteDatabase openOrCreateDatabase (String name, int mode,  
                                                 SQLiteDatabase.CursorFactory factory)  
public abstract boolean deleteDatabase (String name)
```

■ SQL을 실행할 수 있는 메소드

- **create, insert, delete** 등 결과데이터가 없는 SQL문

public void execSQL(String sql) throws SQLException

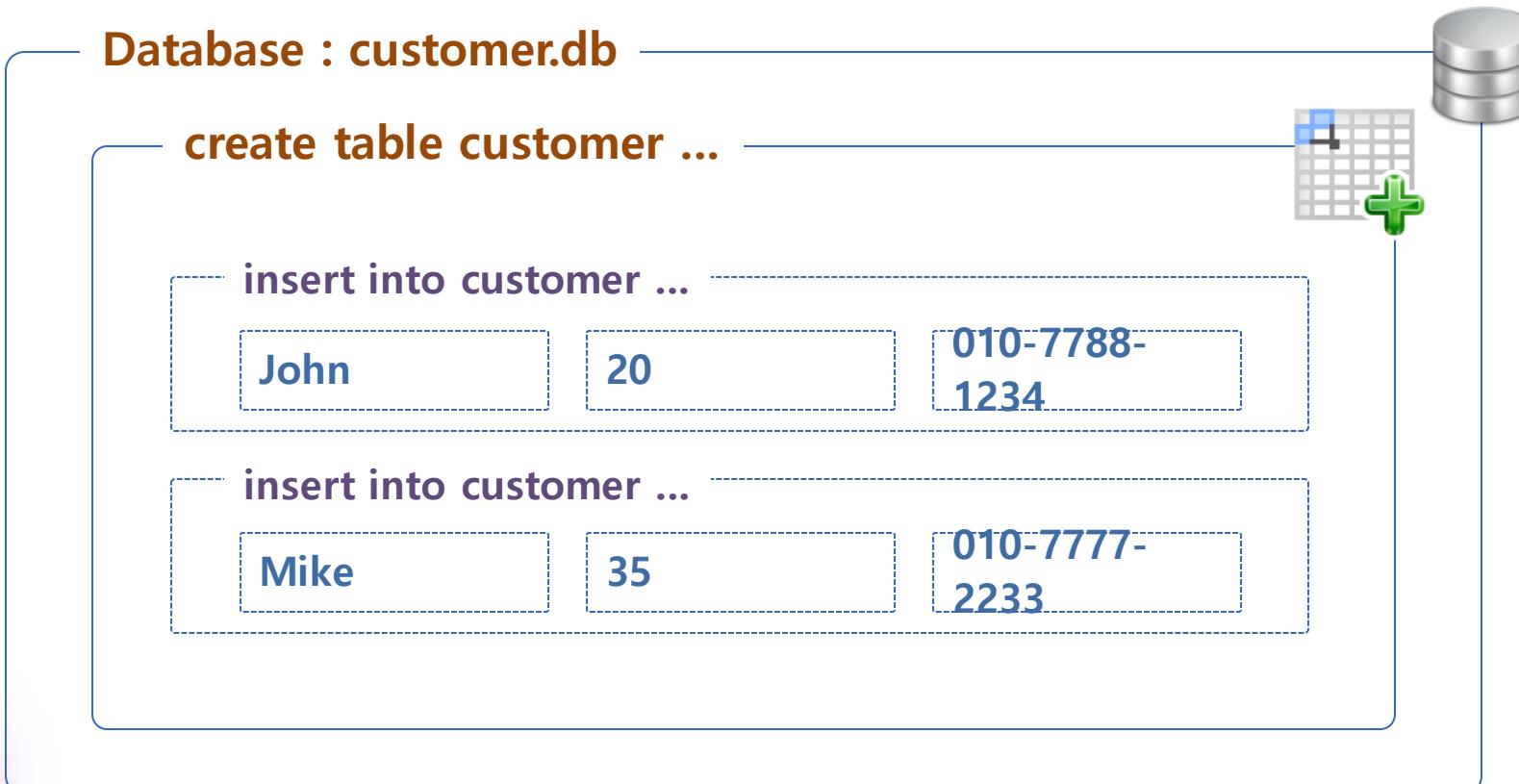
- select 와 같이 조회에 따른 결과 데이터가 있는 SQL문

public Cursor rawQuery(String sql) throws SQLException



데이터베이스 만들기 구조

- ▶ 1단계 : 데이터베이스 생성 → 2단계 : 테이블 생성 → 3단계 : 레코드 추가
- ▶ 테이블 생성과 레코드 추가는 SQL문을 만들어 실행 (create table ... & insert into ...)





XML 레이아웃 만들기

▶ SampleDatabase 프로젝트 만들고 activity_main.xml 파일 열어 화면 구성하기

The screenshot shows the Android Studio interface with the XML Layout Editor open. The title bar indicates the file is 'activity_main.xml'. The editor is divided into several sections:

- Palette:** On the left, it shows categories like Common, Text, Buttons, Widgets, Layouts, Containers, Google, and Legacy. Under 'Common', there's a section for 'Ab TextView' which includes Button, ImageView, RecyclerView, ScrollView, and Switch.
- Component Tree:** This panel on the left lists the hierarchical structure of the layout. It starts with a vertical LinearLayout, which contains two horizontal LinearLayouts. The first horizontal layout has an editText (Plain Text) and a button labeled "데이터베이스 만들기". The second horizontal layout has an editText2 (Plain Text) and a button2 labeled "테이블 만들기". Below these is a ScrollView containing a vertical LinearLayout with a textView.
- Design View:** The main central area shows the visual representation of the layout. It consists of three horizontal rows. The first row has a long input field and a button. The second row has a shorter input field and a button. The third row is a scrollable area containing a single-line text view.
- Attributes Panel:** On the right side, there is a table listing attributes for the selected textView. The table includes:

id	textView
layout_width	match_parent
layout_height	wrap_content
Layout_Margin	[?, ?, ?, ?, ?]
Padding	[?, ?, ?, ?, ?]
Theme	
textSize	20sp
alpha	
autoLink	[]
autoSizeMaxTextSize	
autoSizeMinTextSize	
autoSizePresetSizes	
autoSizeStepGranular	
autoSizeTextType	
autoText	-
background	
bufferType	
capitalize	
clickable	-
contentDescription	
cursorVisible	-
digits	
drawableBottom	
drawableEnd	
drawableLeft	
drawablePadding	
drawableRight	
drawableStart	
drawableTop	

At the bottom of the editor, there are tabs for 'Design' and 'Text', with 'Design' currently selected.



메인 액티비티 코드 만들기 (계속)

```
private void createDatabase(String name) {
    println("createDatabase 호출됨.");

    database = openOrCreateDatabase(name, MODE_PRIVATE, null); —→ ① 데이터베이스를 만들기 위한
                                                                메서드 실행하기
    println("데이터베이스 생성함: " + name);
}
```

```
private void createTable(String name) {
    println("createTable 호출됨.");

    if (database == null) {
        println("데이터베이스를 먼저 생성하세요.");
        return;
    }

    database.execSQL("create table if not exists " + name + "("
        + "_id integer PRIMARY KEY autoincrement, "
        + "name text, "
        + "age integer, "
        + "mobile text)");

    println("테이블 생성함: " + name);
}
```

② 테이블을 만들기 위한 SQL문
실행하기



메인 액티비티 코드 만들기 (계속)

```
private void insertRecord() {  
  
    println("insertRecord 호출됨.");  
    if (database == null) {  
        println("데이터베이스를 먼저 생성하세요.");  
        return;  
    }  
  
    if (tableName == null) {  
        println("테이블을 먼저 생성하세요.");  
        return;  
    }  
  
    database.execSQL("insert into " + tableName  
        + "(name, age, mobile) "  
        + " values "  
        + "( ' John ' , 20, ' 010-1000-1000 ' )");  
  
    println("레코드 추가함.");  
}
```



칼럼 참조용 데이터 타입

칼럼 타입	설명
text, varchar	문자열
smallint, integer	정수 (2바이트 또는 4바이트)
real, float, double	부동소수 (4바이트 또는 8바이트)
boolean	true 또는 false
date, time, timestamp	시간 (날짜, 시간, 날짜+시간)
blob, binary	바이너리

[표] SQLite에서 지원하는 칼럼 타입



테이블 생성과 레코드 추가를 위한 SQL 문법

■ 테이블을 만들기 위한 SQL문

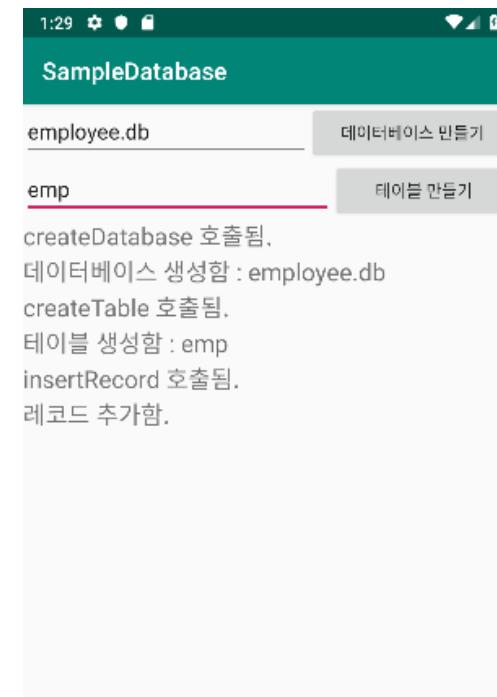
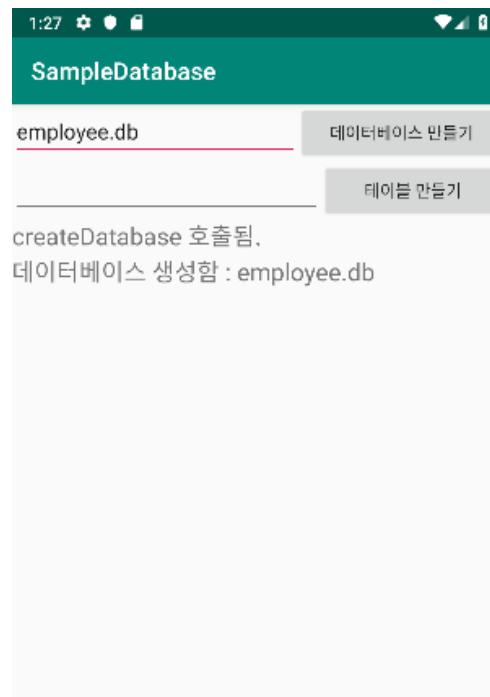
CREATE TABLE [IF NOT EXISTS] table_name(col_name column_definition, ...)
[table_option] ...

■ 레코드를 추가하기 위한 SQL문

INSERT INTO table_name<(column list)> **VALUES** (value, ...)



실행 화면

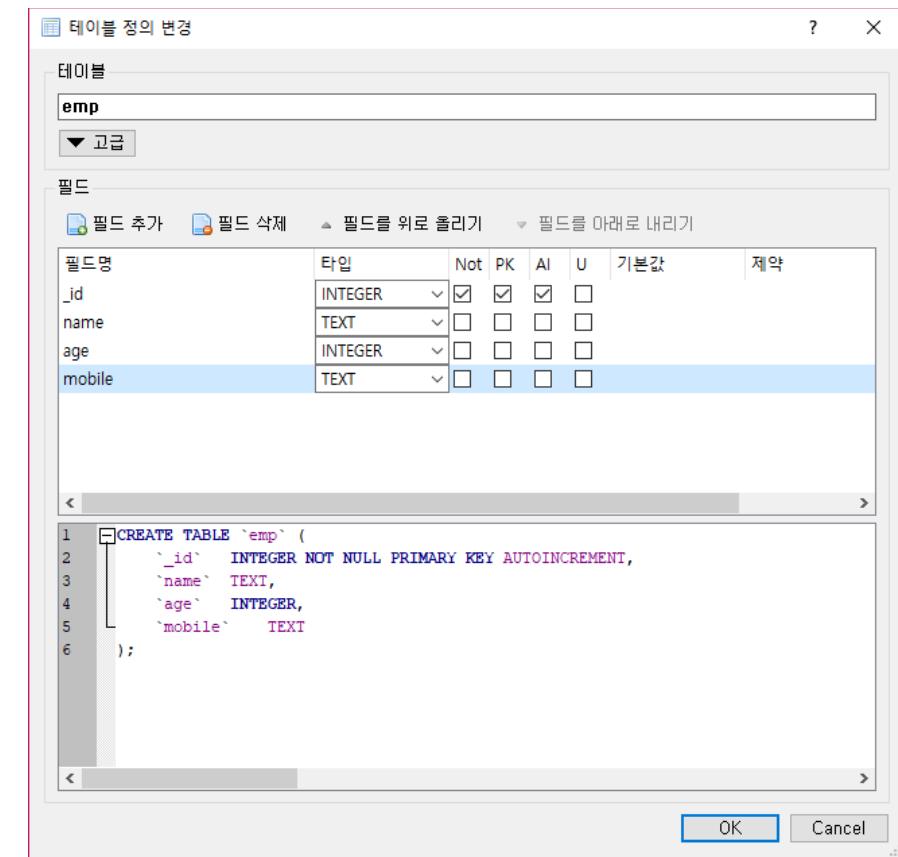
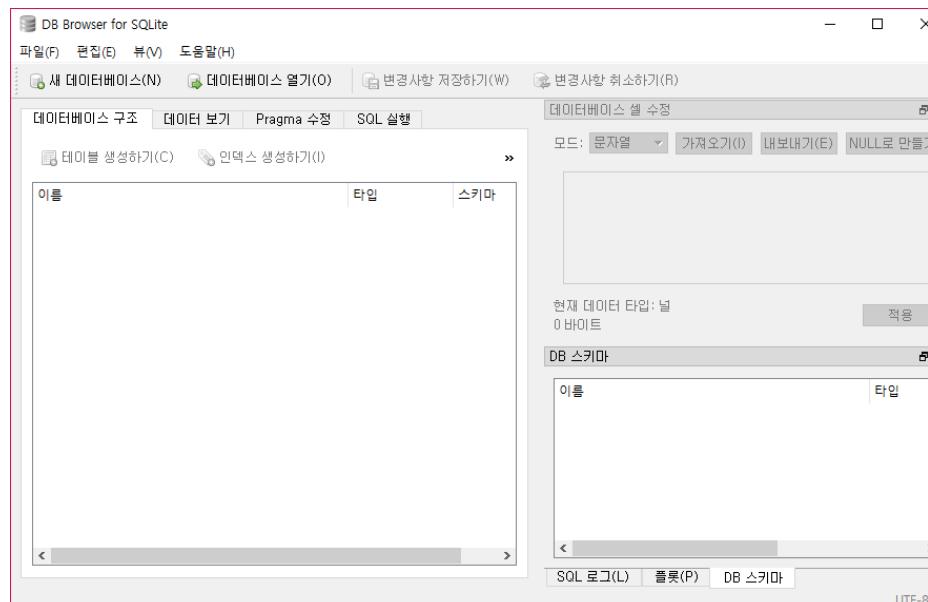


데이터베이스 생성, 테이블 생성 그리고 레코드 추가



데이터베이스 관리도구

<http://sqlitebrowser.org/> 참조





3.

헬퍼 클래스로 업그레이드 지원하기



■ SQLiteOpenHelper 클래스

- 데이터베이스를 만들거나 열기 위해 필요한 일들을 도와주는 역할을 함

■ SQLiteOpenHelper 클래스

```
public SQLiteOpenHelper (Context context, String name,  
                      SQLiteDatabase.CursorFactory factory, int version)
```

```
public abstract void onCreate (SQLiteDatabase db)
```

```
public abstract void onOpen (SQLiteDatabase db)
```

```
public abstract void onUpgrade (SQLiteDatabase db, int oldVersion, int newVersion)
```



헬퍼클래스의 구조

- ▶ 새로 만드는 CustomerDatabase 클래스는 DatabaseHelper 객체와 버전 정보 관리
- ▶ Helper 클래스를 상속한 DatabaseHelper 클래스 안에서는 처음 데이터베이스가 만들어질 때는 onCreate(), 버전이 바뀌어 업그레이드될 때는 onUpgrade() 메소드가 호출됨

CustomerDatabase

DatabaseHelper extends SQLiteOpenHelper

onCreate()

onOpen()

onUpgrade()



헬퍼클래스 만들기

참조파일 SampleDatabase2>/app/java/org.techtown.database/DatabaseHelper.java

```
public class DatabaseHelper extends SQLiteOpenHelper { → ① SQLiteOpenHelper 클래스를  
    public static String NAME = "employee.db";           상속하여 새로운 클래스 정의하기  
    public static int VERSION = 1;  
  
    public DatabaseHelper(Context context) {  
        super(context, NAME, null, VERSION);  
    }  
  
    public void onCreate(SQLiteDatabase db) {  
        println("onCreate 호출됨");  
  
        String sql = "create table if not exists emp("  
            + " _id integer PRIMARY KEY autoincrement, "  
            + " name text, "  
            + " age integer, "  
            + " mobile text)";  
  
        db.execSQL(sql); → ② onCreate() 메서드 안에서 SQL문 실행하기  
    }  
}
```



헬퍼클래스 만들기 (계속)

```
public void onOpen(SQLiteDatabase db) {
    println("onOpen 호출됨");
}

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    println("onUpgrade 호출됨: " + oldVersion + " -> " + newVersion);

    if (newVersion > 1) {
        db.execSQL("DROP TABLE IF EXISTS emp");
    }
}
```



헬퍼클래스 사용

참조파일 SampleDatabase2>/app/java/org.techtown.database/MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    EditText editText;
    EditText editText2;
    TextView textView;

    DatabaseHelper dbHelper;
    SQLiteDatabase database;
    String tableName;

    중략...

    private void createDatabase(String name) {
        println("createDatabase 호출됨.");

        dbHelper = new DatabaseHelper(this);
        database = dbHelper.getWritableDatabase(); ] DatabaseHelper 객체 생성하고
        ] SQLiteDatabase 객체 참조하기

        println("데이터베이스 생성함: " + name);
    }
}
```

중략...



4.

데이터 조회하기



데이터 조회하기

데이터 조회하기 예제

-데이터베이스에서 SQL로 데이터 조회하기

메인 액티비티의
XML 레이아웃 정의

-메인 액티비티 레이아웃 정의

메인 액티비티 코드 작성

-커서를 이용해 데이터 조회하기





메인 액티비티 만들기

```
public void executeQuery() {
    println("executeQuery 호출됨.");

    Cursor cursor = database.rawQuery("select _id, name, age, mobile from emp", null);
    int recordCount = cursor.getCount();
    println("레코드 개수: " + recordCount);

    for (int i = 0; i < recordCount; i++) {
        cursor.moveToNext(); —→ ② 다음 결과 레코드로 넘어가기
    }
}
```

① SQL 실행하고 Cursor 객체
반환받기

```
int id = cursor.getInt(0);
String name = cursor.getString(1);
int age = cursor.getInt(2);
String mobile = cursor.getString(3);

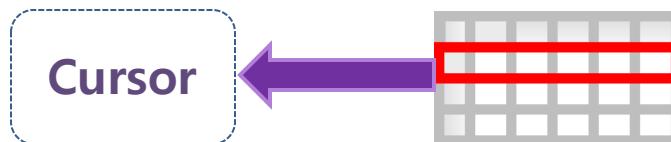
println("레코드#" + i + " : " + id + ", " + name + ", " + age + ", " + mobile);
}
cursor.close();
}
```

중략...



커서의 메소드

- ▶ 데이터베이스 조회를 위해 SELECT문 실행
- ▶ 결과값으로 Cursor 객체 리턴
- ▶ getCount() 메소드로 레코드 개수 확인
- ▶ moveToNext() 메소드로 하나씩 진행
- ▶ getXXX() 메소드로 값 확인



rawQuery("select ...")

public abstract int getColumnCount ()
public abstract int getColumnIndex (String columnName)
public abstract String getColumnName (int columnIndex)
public abstract String[] getColumnNames ()
public abstract int getCount ()
public abstract boolean moveToNext ()
public abstract boolean moveToPrevious ()
public abstract boolean moveToFirst ()
public abstract boolean moveToLast ()
public abstract boolean move (int offset)

public abstract String getString (int columnIndex)
public abstract short getShort (int columnIndex)
public abstract int getInt (int columnIndex)
public abstract long getLong (int columnIndex)
public abstract float getFloat (int columnIndex)
public abstract double getDouble (int columnIndex)
public abstract byte[] getBlob (int columnIndex)



데이터 조회 – SELECT SQL

SELECT [* | DISTINCT] column_name [,columnname2]

FROM tablename1 [,tablename2]

WHERE [condition and|or condition...]

[**GROUP BY** column-list]

[**HAVING** conditions]

[**ORDER BY** "column-list" [ASC | DESC]]



실행 화면



SQL을 이용한 테이블 레코드 조회



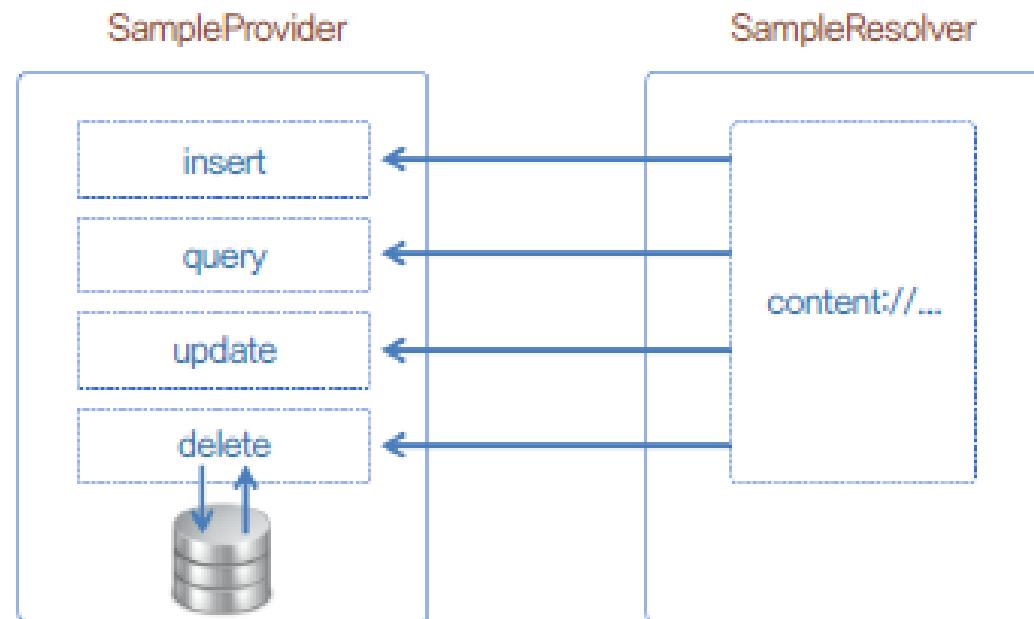
5.

내용 제공자 이해하기



내용 제공자

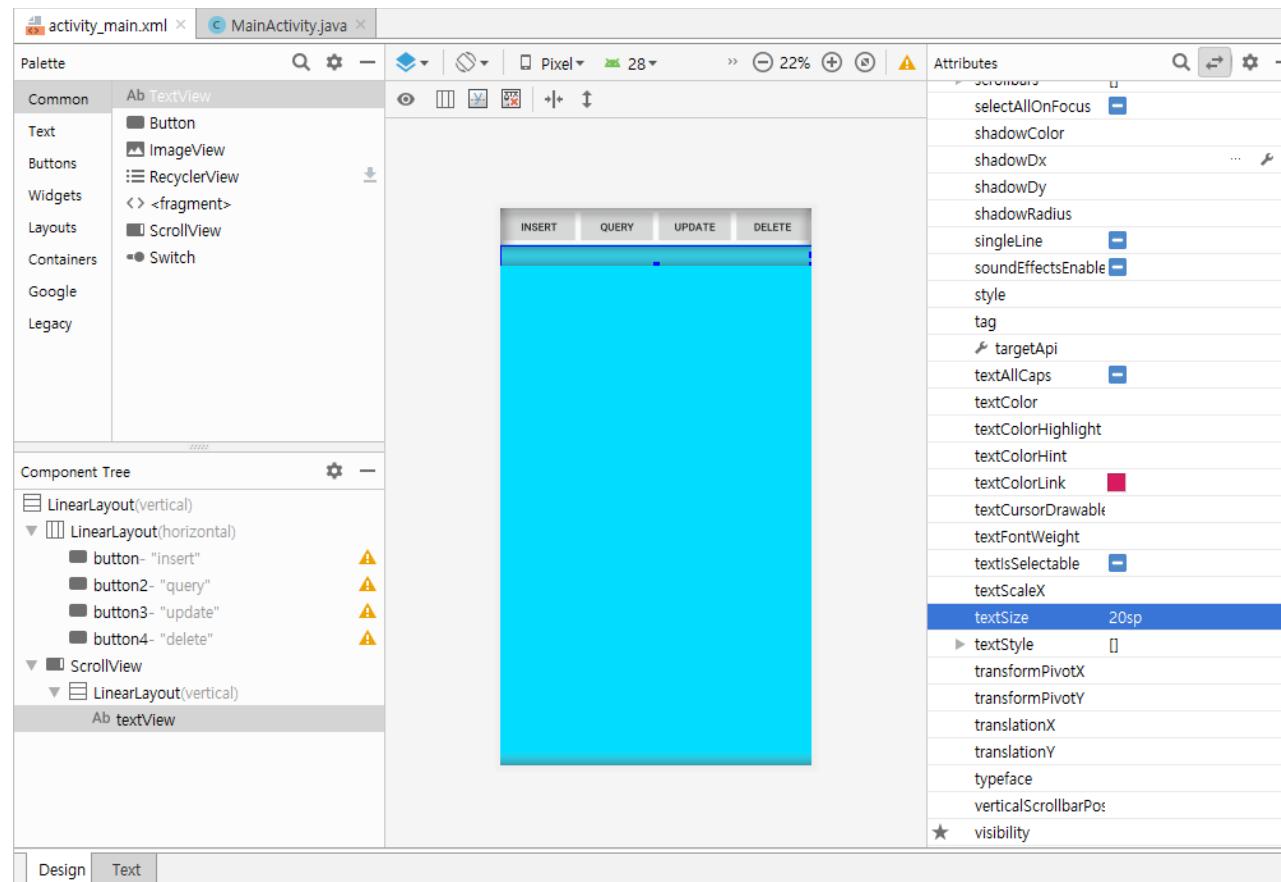
- 내용 제공자는 콘텐트 프로바이더(Content Provider)라고도 불림
- 다른 앱에서 데이터를 접근할 수 있도록 함
- Provider에서 제공하는 데이터는 Resolver를 이용해 접근함





화면 레이아웃 만들기

- 상단에 버튼 배치하고 하단에는 스크롤뷰 안에 텍스트뷰 배치





데이터베이스 코드 만들기

- SQLiteHelper를 이용해 데이터베이스를 위한 코드 추가

```
public class DatabaseHelper  
extends SQLiteOpenHelper {  
    private static final String  
DATABASE_NAME="person.db";  
    private static final int  
DATABASE_VERSION=1;  
  
    public static final String  
TABLE_NAME="person";  
    public static final String  
PERSON_ID="_id";  
    public static final String  
PERSON_NAME="name";  
    public static final String  
PERSON_AGE="age";  
    public static final String  
PERSON_MOBILE="mobile";  
  
    public static final String[]  
ALL_COLUMNS={PERSON_ID,PERSON_NAME  
,PERSON_AGE,PERSON_MOBILE};
```

```
private static final String CREATE_TABLE =  
    "CREATE TABLE " + TABLE_NAME + " (" +  
        PERSON_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +  
        PERSON_NAME + " TEXT, " +  
        PERSON_AGE + " INTEGER, " +  
        PERSON_MOBILE + " TEXT" +  
    ")";  
  
public DatabaseHelper(Context context) {  
    super(context, DATABASE_NAME, null, DATABASE_VERSION);  
}  
  
@Override  
public void onCreate(SQLiteDatabase db) {  
    db.execSQL(CREATE_TABLE);  
}  
  
@Override  
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
    db.execSQL("DROP TABLE IF EXISTS "+ TABLE_NAME);  
    onCreate(db);  
}
```



Provider 클래스 만들기

- ContentProvider를 상속해 새로운 Provider 클래스 정의

참조파일 SampleProvider>/app/java/org.techtown.provider/PersonProvider.java

```
public class PersonProvider extends ContentProvider {

    private static final String AUTHORITY = "org.techtown.provider";
    private static final String BASE_PATH = "person";
    public static final Uri CONTENT_URI = Uri.parse("content://" + AUTHORITY + "/" + BASE_PATH );

    private static final int PERSONS = 1;
    private static final int PERSON_ID = 2;
    private static final UriMatcher uriMatcher = new UriMatcher(UriMatcher.NO_MATCH);
    static {
        uriMatcher.addURI(AUTHORITY, BASE_PATH, PERSONS);
        uriMatcher.addURI(AUTHORITY, BASE_PATH + "/#", PERSON_ID);
    }

    private SQLiteDatabase database;

    @Override
    public boolean onCreate() {
        DatabaseHelper helper = new DatabaseHelper(getContext());
        database = helper.getWritableDatabase();

        return true;
    }
}
```



Provider 클래스 만들기

- 조회를 위한 query 메서드 정의
- UriMatcher를 사용하고 커서를 생성해 반환

```
@Nullable  
@Override  
public Cursor query(Uri uri, String[] strings, String s, String[] strings1, String s1) {  
    Cursor cursor;  
    switch (uriMatcher.match(uri)) {  
        case PERSONS:  
            cursor = database.query(DatabaseHelper.TABLE_NAME,  
                DatabaseHelper.ALL_COLUMNS,  
                s,null,null,null,DatabaseHelper.PERSON_NAME +" ASC");  
            break;  
        default:  
            throw new IllegalArgumentException("알 수 없는 URI " + uri);  
    }  
    cursor.setNotificationUri(getContext().getContentResolver(), uri);  
  
    return cursor;  
}
```



Provider 클래스 만들기

- 추가를 위한 insert 등의 메서드 추가

```
@Nullable  
@Override  
public Uri insert(Uri uri, ContentValues contentValues) {  
    long id = database.insert(DatabaseHelper.TABLE_NAME, null, contentValues);  
  
    if (id > 0) {  
        Uri _uri = ContentUris.withAppendedId(CONTENT_URI, id);  
        getContext().getContentResolver().notifyChange(_uri, null);  
        return _uri;  
    }  
  
    throw new SQLException("추가 실패-> URI :" + uri);  
}
```

```
public String getType(@NonNull Uri uri) {  
    switch(uriMatcher.match(uri)) {  
        case PERSONS:  
            return "vnd.android.cursor.dir/persons";  
        default:  
            throw new IllegalArgumentException("알 수 없는 URI : "+uri);  
    }  
}
```



Provider 클래스 만들기

```
public int update(@NonNull Uri uri, @Nullable ContentValues values, @Nullable String selection,
@Nullable String[] selectionArgs) {
    int count = 0;
    switch (uriMatcher.match(uri)) {
        case PERSONS:
            count = database.update(DatabaseHelper.TABLE_NAME,values,selection, selectionArgs);
            break;
        default:
            throw new IllegalArgumentException("알수없는 URI: "+uri);
    }
    getContext().getContentResolver().notifyChange(uri, null);
    return count;
}

public int delete(@NonNull Uri uri, @Nullable String selection, @Nullable String[] selectionArgs) {
    int count = 0;
    switch (uriMatcher.match(uri)) {
        case PERSONS:
            count = database.delete(DatabaseHelper.TABLE_NAME,selection, selectionArgs);
            break;
        default:
            throw new IllegalArgumentException("알수없는 URI: "+uri);
    }
    getContext().getContentResolver().notifyChange(uri, null);
    return count;
}
```



Provider 클래스 만들기

Content URI 정의 형식

```
content://org.techtown.provider/person/1
```

content:// → 내용 제공자에 의해 제어되는 데이터라는 의미로 항상 content://로 시작함

Authority → org.techtown.provider 부분을 가리키며 특정 내용 제공자를 구분하는 고유한 값

Base Path → person 부분을 가리키며 요청할 데이터의 자료형을 결정함 (여기에서는 테이블 이름)

ID → 맨 뒤의 1과 같은 숫자를 가리키며 요청할 데이터 레코드를 지정함

- query 메서드

```
Cursor query (  
    Uri uri,  
    String[] projection,  
    String selection,  
    String[] selectionArgs,  
    String sortOrder  
)
```



메인 액티비티에서 Provider의 메서드 호출

- **insertPerson** 메서드 정의

```
public void insertPerson() {  
    println("insertPerson 호출됨");  
  
    String uriString = "content://org.techtown.provider/person";  
    Uri uri = new Uri.Builder().build().parse(uriString);  
  
    Cursor cursor = getContentResolver().query(uri, null, null, null, null);  
    String[] columns = cursor.getColumnNames();  
    println("columns count -> " + columns.length);  
  
    for (int i = 0; i < columns.length; i++) {  
        println("#" + i + " : " + columns[i]);  
    }  
  
    ContentValues values = new ContentValues();  
    values.put("name", "john");  
    values.put("age", 20);  
    values.put("mobile", "010-1000-1000");  
  
    uri = getContentResolver().insert(uri, values);  
    println("insert 결과-> " + uri.toString());  
}
```

중략…



메인 액티비티에서 Provider의 메서드 호출

- **queryPerson** 메서드 정의

```
public void queryPerson() {
    try {
        String uriString = "content://org.techtown.provider/person";
        Uri uri = new Uri.Builder().build().parse(uriString);

        String[] columns = new String[] {"name", "age", "mobile"};
        Cursor cursor = getContentResolver().query(uri, columns, null, null, "name ASC");
        println("query 결과: " + cursor.getCount());

        int index = 0;
        while(cursor.moveToNext()) {
            String name = cursor.getString(cursor.getColumnIndex(columns[0]));
            int age = cursor.getInt(cursor.getColumnIndex(columns[1]));
            String mobile = cursor.getString(cursor.getColumnIndex(columns[2]));

            println("#" + index + " -> " + name + ", " + age + ", " + mobile);
            index += 1;
        }
    }
}
```



메인 액티비티에서 Provider의 메서드 호출

- **queryPerson** 메서드 정의

```
public void insertPerson() {  
    println("insert 호출됨");  
  
    String uriStr = "content://com.example.myprovider/person";  
    Uri uri = new Uri.Builder().build().parse(uriStr);  
  
    Cursor cursor = getContentResolver().query(uri, null, null, null, null);  
    String[] columns = cursor.getColumnNames();  
    println("columns count->" + columns.length);  
  
    for(int i = 0; i < columns.length; i++) {  
        println("#" + i + ":" + columns[i]);  
    }  
  
    ContentValues values = new ContentValues();  
    values.put("name", "kim");  
    values.put("age", 20);  
    values.put("mobile", "010-1000-1000");  
  
    uri = getContentResolver().insert(uri, values);  
    println("insert 결과-> " + uri.toString());  
}
```



메인 액티비티에서 Provider의 메서드 호출

- queryPerson 메서드 정의

```
public void updatePerson() {  
    String uriStr = "content://com.example.myprovider/person";  
    Uri uri = new Uri.Builder().build().parse(uriStr);  
  
    String selection = "mobile = ?";  
    String[] selectionArgs = new String[]{"010-1000-1000"};  
    ContentValues updateValue = new ContentValues();  
    updateValue.put("mobile", "010-2000-2000");  
    int count = getContentResolver().update(uri, updateValue, selection, selectionArgs);  
    println("update 결과: "+count);  
}  
  
public void deletePerson() {  
    String uriStr = "content://com.example.myprovider/person";  
    Uri uri = new Uri.Builder().build().parse(uriStr);  
  
    String selection = "name = ?";  
    String[] selectionArgs = new String[]{"kim"};  
  
    int count = getContentResolver().delete(uri, selection, selectionArgs);  
    println("delete 결과: "+count);  
}
```



매니페스트에 내용 제공자 등록

- SD 카드 접근 권한과 provider 태그 추가

참조파일 SampleProvider>/app/manifests/AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="org.techtown.provider">

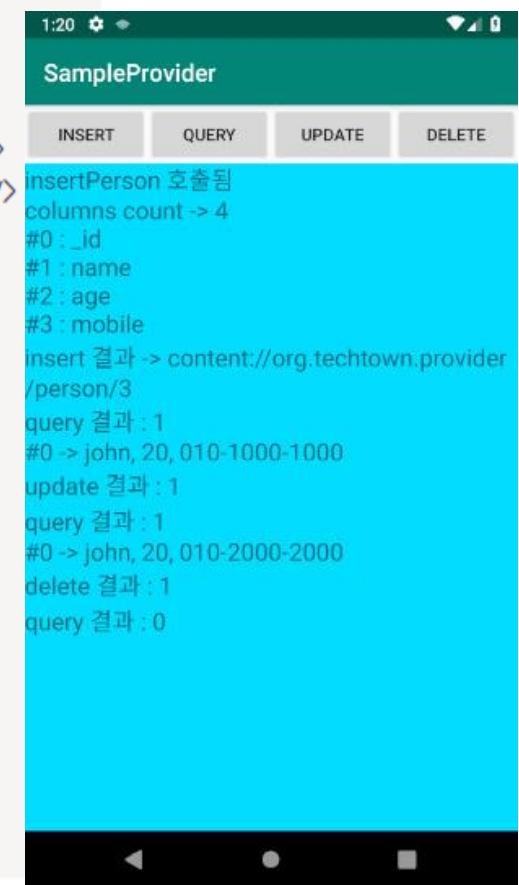
    <permission android:name="org.techtown.provider.READ_DATABASE" android:protectionLevel="normal" />
    <permission android:name="org.techtown.provider.WRITE_DATABASE" android:protectionLevel="normal" />

    <application

        중략...

        <provider
            android:authorities="org.techtown.provider"
            android:name=".PersonProvider"
            android:exported="true"
            android:readPermission="org.techtown.provider.READ_DATABASE"
            android:writePermission="org.techtown.provider.WRITE_DATABASE"
        />

    </application>
</manifest>
```





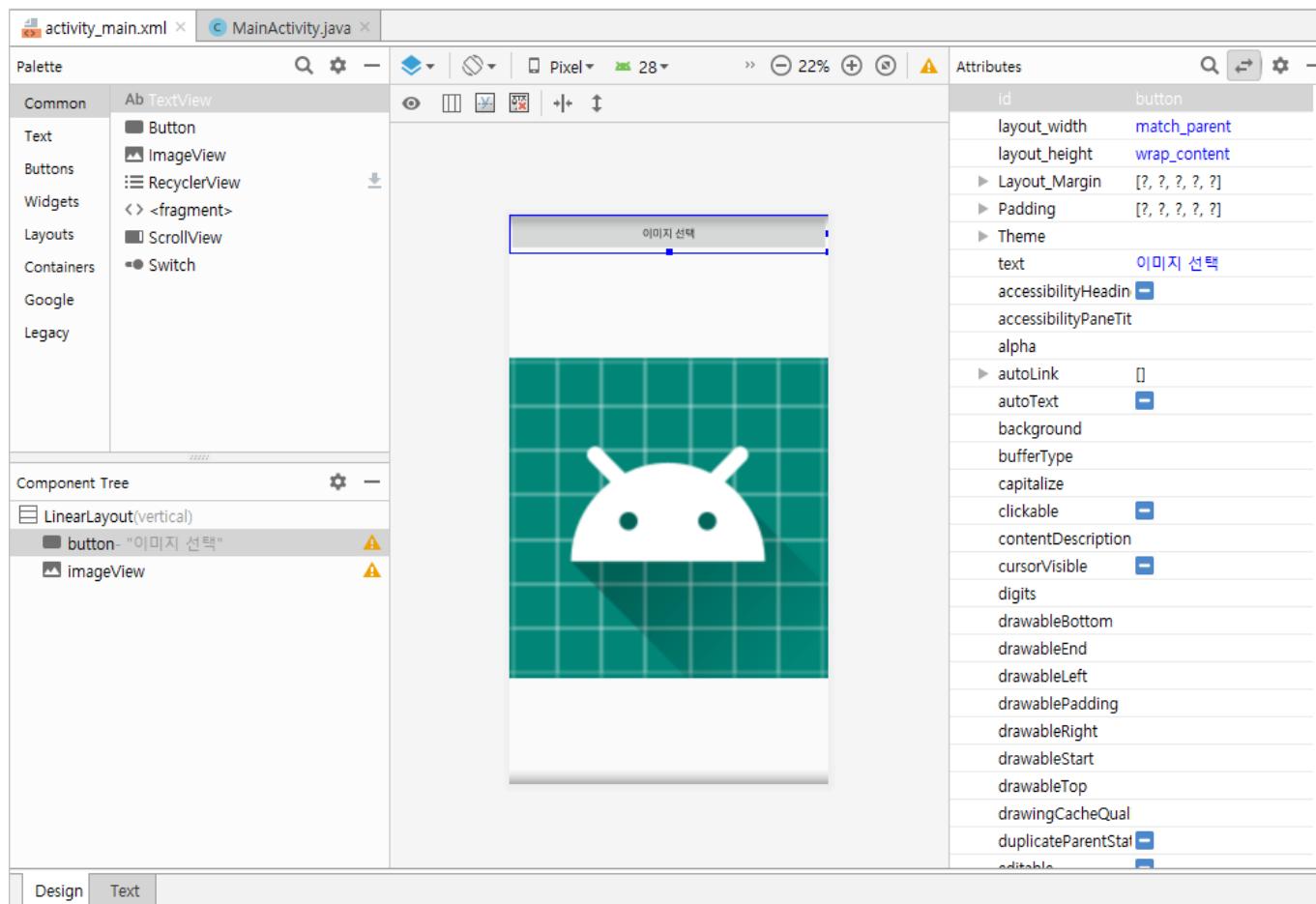
6.

앨범과 연락처 조회하기



화면 레이아웃 만들기

- 상단에 버튼 배치하고 하단에는 이미지뷰 배치





메인 액티비티에서 앨범 띄우기

- 인텐트를 이용해 앨범 화면 띄우기

```
public void openGallery() {  
    Intent intent = new Intent();  
    intent.setType("image/*");  
    intent.setAction(Intent.ACTION_GET_CONTENT);  
  
    startActivityForResult(intent, 101);  
}
```



선택된 사진 보여주기

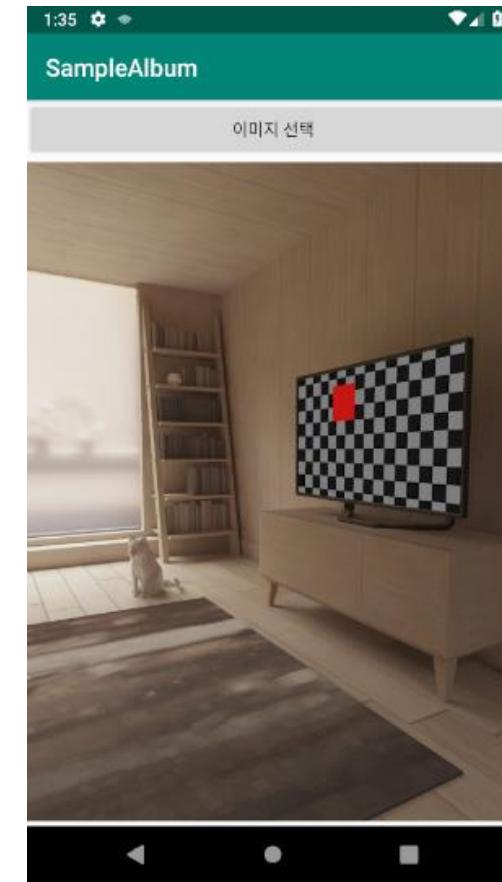
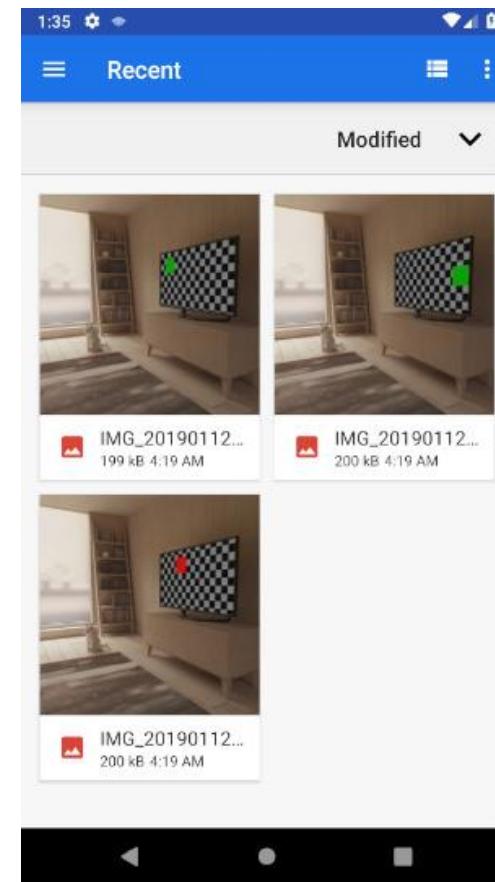
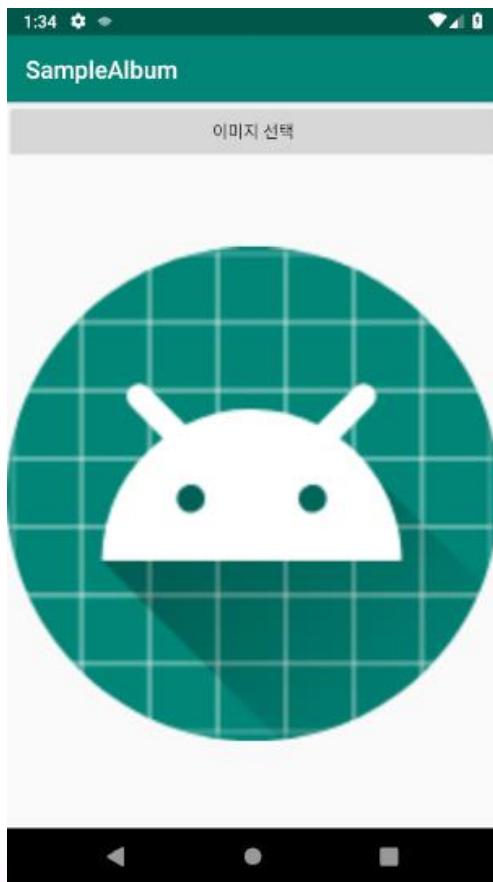
- **onActivityResult** 메서드 안에서 선택된 사진 보여주기

```
@Override  
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    if(requestCode == 101) {  
        if(resultCode == RESULT_OK) {  
            Uri fileUri = data.getData();  
  
            ContentResolver resolver = getContentResolver(); → ❶ ContentResolver 객체 참조하기  
  
            try {  
  
                InputStream instream = resolver.openInputStream(fileUri); → ❷ ContentResolver  
                Bitmap imgBitmap = BitmapFactory.decodeStream(instream);  
                imageView.setImageBitmap(imgBitmap);  
                instream.close();  
            } catch(Exception e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

객체의 openInputStream() 메서드로
 파일 읽어 들이기



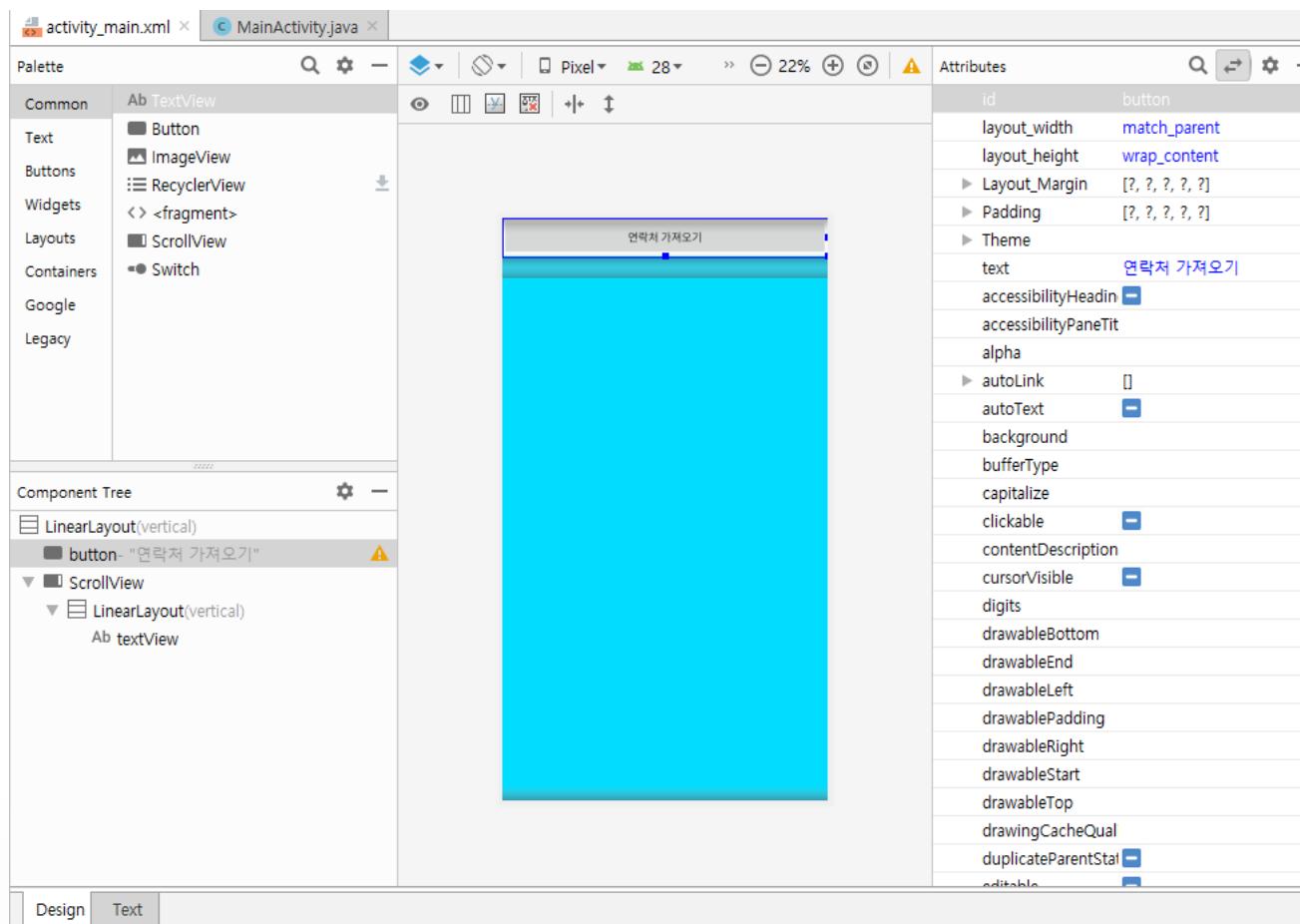
앱 실행





연락처 선택을 위한 화면 레이아웃 만들기

- 상단에 버튼 배치하고 하단에는 스크롤뷰 안에 텍스트뷰 배치하기





메인 액티비티에서 연락처 화면 띄우기

- 인텐트로 연락처 선택 화면 띄우고 결과 받아 표시하기

```
public void chooseContacts() {  
    Intent contactPickerIntent = new Intent(Intent.ACTION_PICK,  
    ContactsContract.Contacts.CONTENT_URI); → ❶ 연락처 화면을 띄우기 위한 인텐트 만들기  
    startActivityForResult(contactPickerIntent, 101);  
}  
  
@Override  
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    if (resultCode == RESULT_OK) {  
        if (requestCode == 101) {  
            try {  
                Uri contactsUri = data.getData();  
                String id = contactsUri.getLastPathSegment(); → ❷ 선택한 연락처의 id 값 확인하기  
  
                getContacts(id);  
            } catch (Exception e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```



메인 액티비티에서 연락처 화면 띄우기

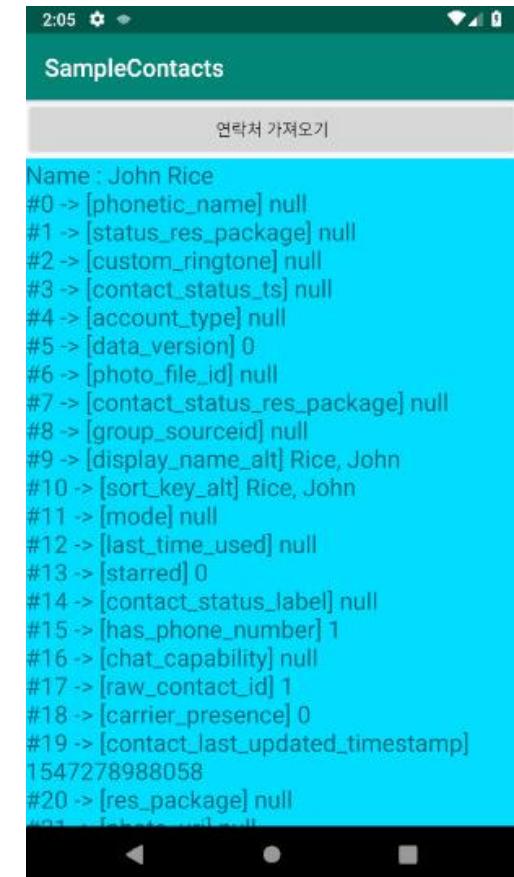
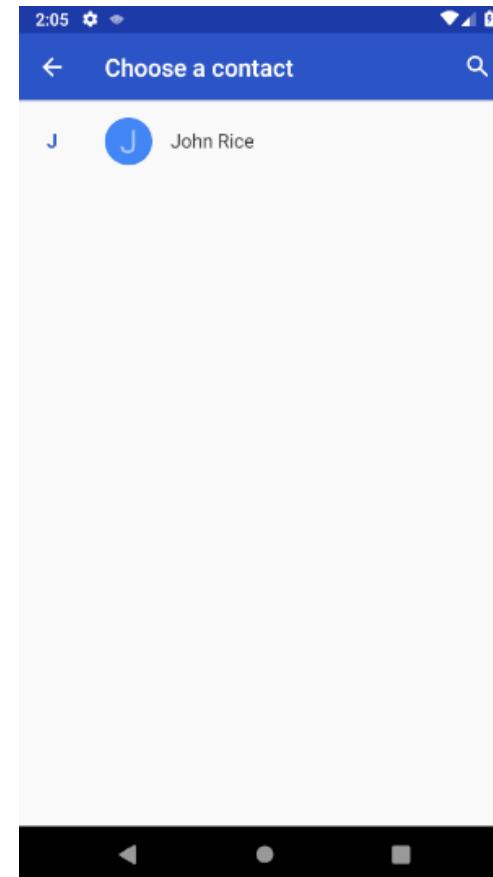
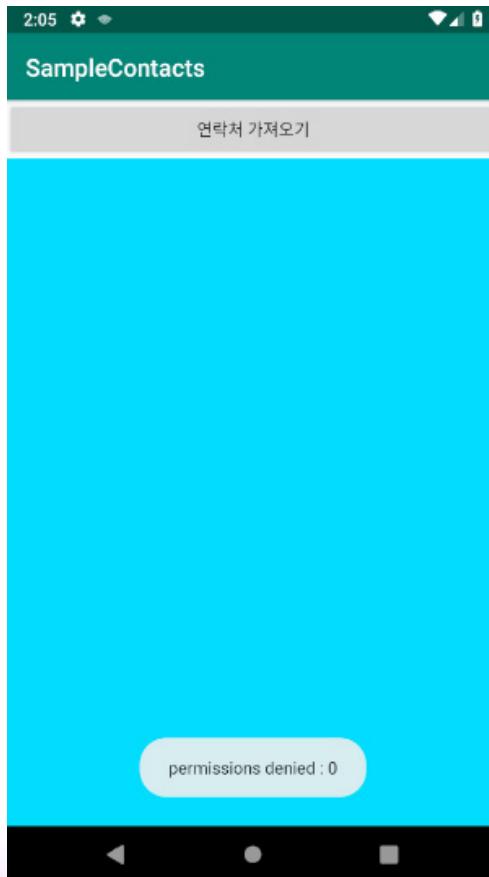
- `getContacts` 메서드에서 연락처 정보 접근

```
public void getContacts(String id) {  
    Cursor cursor = null;  
    String name = "";  
  
    try {  
        cursor = getContentResolver().query(ContactsContract.Data.CONTENT_URI,  
                                            null,  
                                            ContactsContract.Data.CONTACT_ID + "=?",  
                                            new String[] { id },  
                                            null);  
  
        if (cursor.moveToFirst()) {  
            name = cursor.getString(cursor.getColumnIndex(ContactsContract.Data.DISPLAY_NAME));  
            println("Name : " + name);  
  
            String columns[] = cursor.getColumnNames();  
            for (String column : columns) {  
                int index = cursor.getColumnIndex(column);  
                String columnOutput = ("#" + index + " -> [" + column + "] " + cursor.getString(index));  
                println(columnOutput);  
            }  
            cursor.close();  
        }  
    }  
}
```

ContentResolver 객체의
query() 메서드 호출하기



앱 실행





뷰에 그래픽을 그리는 방법에 대한 이해



1 뷰에 그래픽 그리기

2 드로어블 객체로 만들어 그리기

3 비트맵 이미지 사용하기

4 페인트보드 만들기

5 멀티터치 이미지 뷰어 만들기

6 머티리얼 디자인의 개념 알아두기



뷰 위에 직접 그릴 수 있나요?

- 뷰에 그래픽 그리기
- 드로어블 객체로 만들어 그리기



비트맵 이미지는 어떻게 사용하나요?

- 비트맵 이미지 사용하기



사인 받는 페드를 만들고 싶어요

- 페인트보드 만들기



두 손가락으로 이미지를 확대할 수 있나요?

- 멀티터치 이미지뷰어 만들기



머티리얼 디자인이 적용되어 있다고 들었어요

- 머티리얼 디자인이란?





1.

뷰에 그래픽 그리기



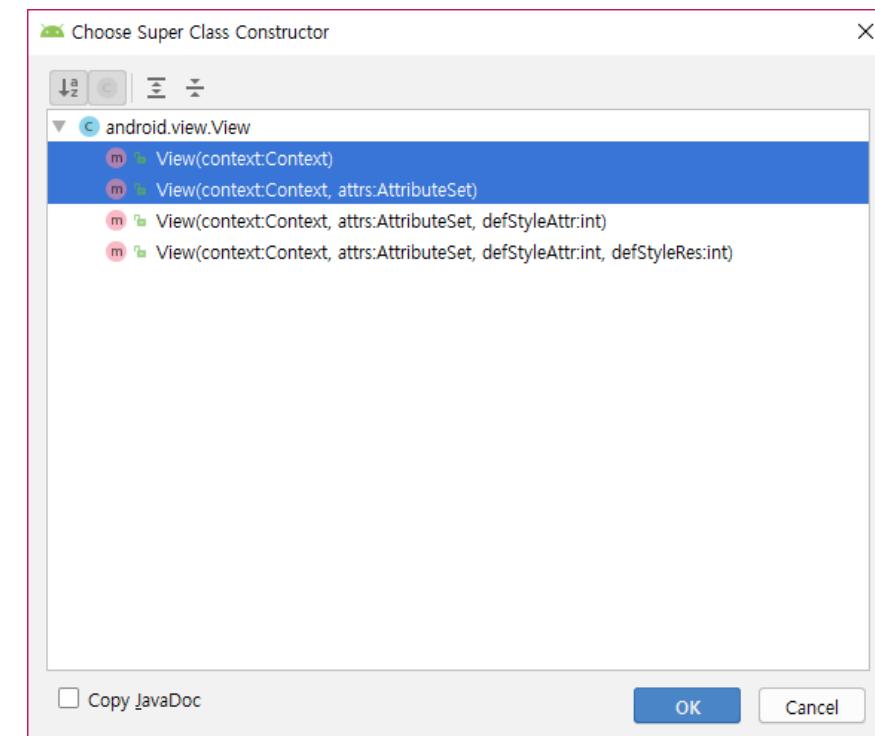
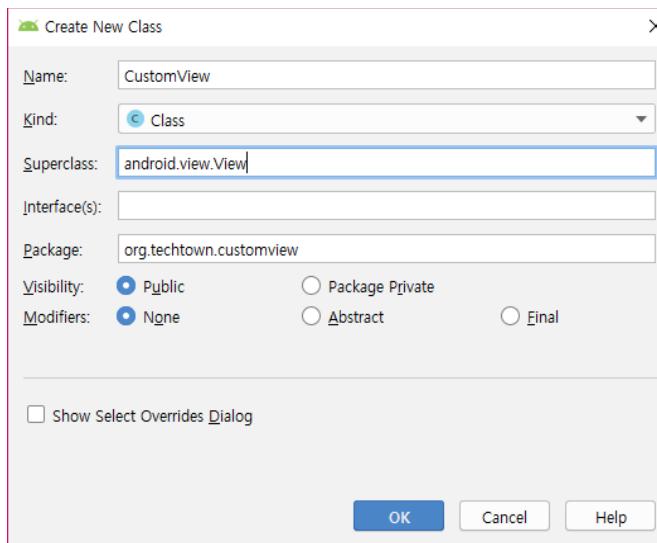
뷰 위에 그래픽을 그리는 순서

구분	설명
1단계	새로운 클래스를 만들고 뷰를 상속받습니다.
2단계	페인트 객체를 초기화하고 필요한 속성을 설정합니다.
3단계	onDraw() 메서드 내에 사각형을 그리는 메서드를 호출합니다.
4단계	onTouchEvent() 메서드 내에 터치 이벤트를 처리하는 코드를 넣습니다.
5단계	새로 만든 뷰를 메인 액티비티에 추가합니다.



뷰 상속하기

- 새로운 뷰 클래스 정의
- 생성자 2개 추가





뷰 상속하기

- **init** 메서드 안에서 초기화

참조파일 SampleCustomView>/app/java/org.techtown.graphics.custom/CustomView.java

```
public class CustomView extends View {

    public CustomView(Context context) {
        super(context);

        init(context);
    }

    public CustomView(Context context, AttributeSet attrs) {
        super(context, attrs);

        init(context);
    }

    private void init(Context context) {
    }
}
```



뷰 상속하기

- Paint 객체 생성

참조파일 SampleCustomView>/app/java/org.techtown.graphics.custom/CustomView.java

```
public class CustomView extends View {  
  
    private Paint paint;  
  
    중략...  
  
    private void init(Context context) {  
        paint = new Paint();  
        paint.setColor(Color.RED);  
    }  
}
```

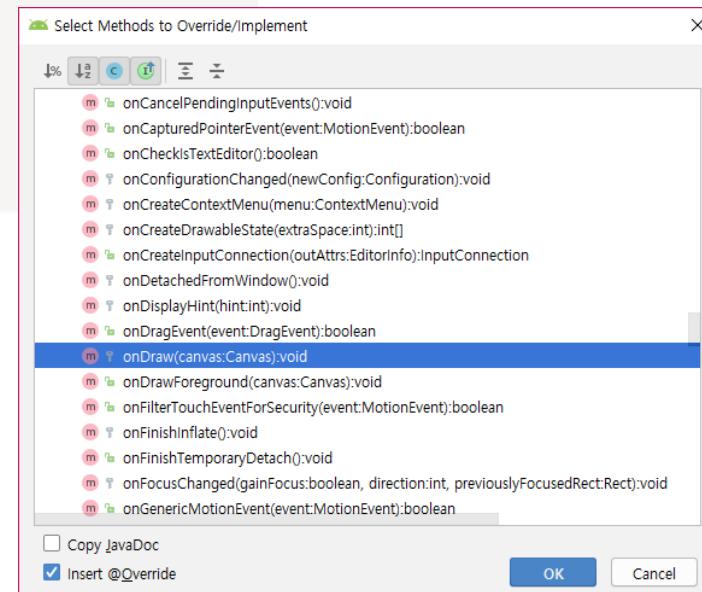


onDraw 메서드 구현

- onDraw 메서드 재정의

참조파일 SampleCustomView>/app/java/org.techtown.graphics.custom/CustomView.java

```
public class CustomView extends View {  
  
    중략...  
  
    @Override  
    protected void onDraw(Canvas canvas) {  
        super.onDraw(canvas);  
        canvas.drawRect(100, 100, 200, 200, paint);  
    }  
}
```





onTouchEvent 메서드 구현

- **onTouchEvent** 메서드 재정의

참조파일 SampleCustomView>/app/java/org.techtown.graphics.custom/CustomView.java

```
public class CustomView extends View {  
  
    중략...  
  
    @Override  
    public boolean onTouchEvent(MotionEvent event) {  
        if (event.getAction() == MotionEvent.ACTION_DOWN) {  
            Toast.makeText(super.getContext(), "MotionEvent.ACTION_DOWN : " +  
                event.getX() + ", " + event.getY(), Toast.LENGTH_LONG).show();  
        }  
  
        return super.onTouchEvent(event);  
    }  
}
```



메인 액티비티에 새로 만든 뷰 추가

- XML 레이아웃 또는 코드에서 새로 만든 뷰 추가

참조파일 SampleCustomView>/app/java/org.techtown.graphics.custom/MainActivity.java

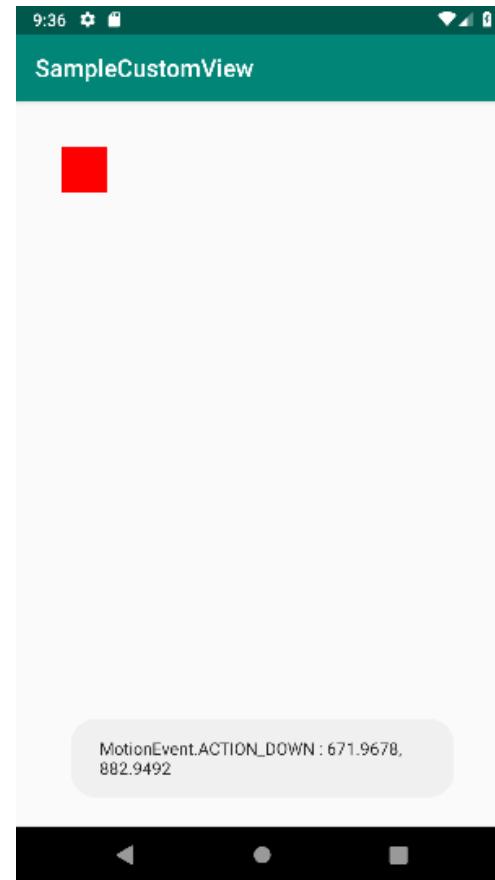
```
public class MainActivity extends AppCompatActivity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        CustomView view = new CustomView(this);
        setContentView(view);
    }
}
```



실행 화면





스타일 적용해보기

- View를 상속한 새로운 뷰를 정의하고 Paint 객체의 속성 수정

```
paint.setStyle(Paint.Style.FILL); —→ ① 첫 번째 사각형을 Fill 스타일로 설정
```

```
paint.setColor(Color.RED);
```

```
canvas.drawRect(10, 10, 100, 100, paint);
```

```
paint.setStyle(Paint.Style.STROKE); —→ ② 첫 번째 사각형을 Stroke 스타일로 설정
```

```
paint.setStrokeWidth(2.0F);
```

```
paint.setColor(Color.GREEN);
```

```
canvas.drawRect(10, 10, 100, 100, paint);
```

```
paint.setStyle(Paint.Style.FILL); —→ ③ 두 번째 사각형을 Fill 스타일로 설정
```

```
paint.setARGB(128, 0, 0, 255);
```

```
canvas.drawRect(120, 10, 210, 100, paint);
```

```
DashPathEffect dashEffect = new DashPathEffect(new float[]{5,5}, 1);
```

```
paint.setStyle(Paint.Style.STROKE);
```

```
paint.setStrokeWidth(3.0F);
```

```
paint.setPathEffect(dashEffect);
```

```
paint.setColor(Color.GREEN);
```

```
canvas.drawRect(120, 10, 210, 100, paint);
```

④ 두 번째 사각형을 Stroke 스타일로
설정하고 PathEffect 적용



스타일 적용해보기

- View를 상속한 새로운 뷰를 정의하고 Paint 객체의 속성 수정

```
paint = new Paint();

paint.setColor(Color.MAGENTA); → ⑤ 첫 번째 원에 색상 적용
canvas.drawCircle(50, 160, 40, paint);

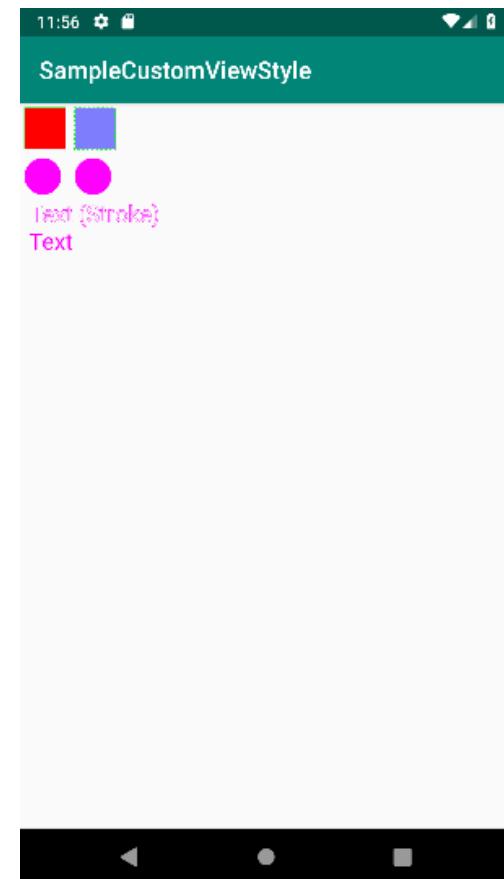
paint.setAntiAlias(true); → ⑥ 두 번째 원에 AntiAlias 설정
canvas.drawCircle(160, 160, 40, paint);

paint.setStyle(Paint.Style.STROKE); → ⑦ 첫 번째 텍스트를 Stroke 스타일로 설정
paint.setStrokeWidth(1);
paint.setColor(Color.MAGENTA);
paint.setTextSize(30);
canvas.drawText("Text (Stroke)", 20, 260, paint); → ⑧ 두 번째 텍스트를 Fill 스타일로 설정

paint.setStyle(Paint.Style.FILL);
paint.setTextSize(30);
canvas.drawText("Text", 20, 320, paint);
```



앱 실행





2.

드로어블 객체로 만들어 그리기



드로어블 객체를 사용하는 세 가지 방법

구분	설명
리소스 파일의 사용	프로젝트 리소스에 이미지와 같은 파일을 포함시킨 후 읽어 들여 사용합니다.
XML로 정의하여 사용	그리기 객체의 속성을 정의한 XML 파일을 정의하여 사용합니다.
소스 코드에서 객체를 만들어 사용	소스 코드에서 new 연산자를 이용하여 그리기 객체를 만든 후 사용합니다. 비트맵 이미지를 주어진 좌표 값에 그립니다.



드로어블을 위한 새로운 뷰 정의

참조파일 SampleCustomViewDrawable>/app/java/org.techtown.graphics.custom.drawable/CustomViewDrawable.java

```
public class CustomViewDrawable extends View {

    private ShapeDrawable upperDrawable;
    private ShapeDrawable lowerDrawable;

    public CustomViewDrawable(Context context) {
        super(context);

        init(context);
    }

    public CustomViewDrawable(Context context, AttributeSet attrs) {
        super(context, attrs);
    }
}
```



드로어블을 위한 새로운 뷰 정의

```
    init(context);
}

private void init(Context context) {
    WindowManager manager = (WindowManager)
        context.getSystemService(Context.WINDOW_SERVICE);
    Display display = manager.getDefaultDisplay();
    int width = display.getWidth();
    int height = display.getHeight();

    Resources curRes = getResources();
    int blackColor = curRes.getColor(R.color.color01);
    int grayColor = curRes.getColor(R.color.color02);
    int darkGrayColor = curRes.getColor(R.color.color03);

    upperDrawable = new ShapeDrawable(); —> ③ Drawable 객체 생성

    RectShape rectangle = new RectShape();
    rectangle.resize(width, height*2/3);
    upperDrawable.setShape(rectangle);
    upperDrawable.setBounds(0, 0, width, height*2/3);

    LinearGradient gradient = new LinearGradient(0, 0, 0, height*2/3, —> ④ LinearGradient
        grayColor, blackColor, Shader.TileMode.CLAMP);           객체 생성
```

① 윈도우 매니저를 이용해 뷰의 폭과 높이 확인

② 리소스에 정의된 색상 값을 변수에 설정

③ Drawable 객체 생성

④ LinearGradient 객체 생성



드로어블을 위한 새로운 뷰 정의

```
Paint paint = upperDrawable.getPaint();

paint.setShader(gradient); —→ ⑤ Paint 객체에 새로 생성한 LinearGradient 객체를 Shader로 설정

lowerDrawable = new ShapeDrawable();

RectShape rectangle2 = new RectShape();
rectangle2.resize(width, height*1/3);
lowerDrawable.setShape(rectangle2);
lowerDrawable.setBounds(0, height*2/3, width, height);

LinearGradient gradient2 = new LinearGradient(0, 0, 0, height*1/3,
                                              blackColor, darkGrayColor, Shader.TileMode.CLAMP);

Paint paint2 = lowerDrawable.getPaint();
paint2.setShader(gradient2);

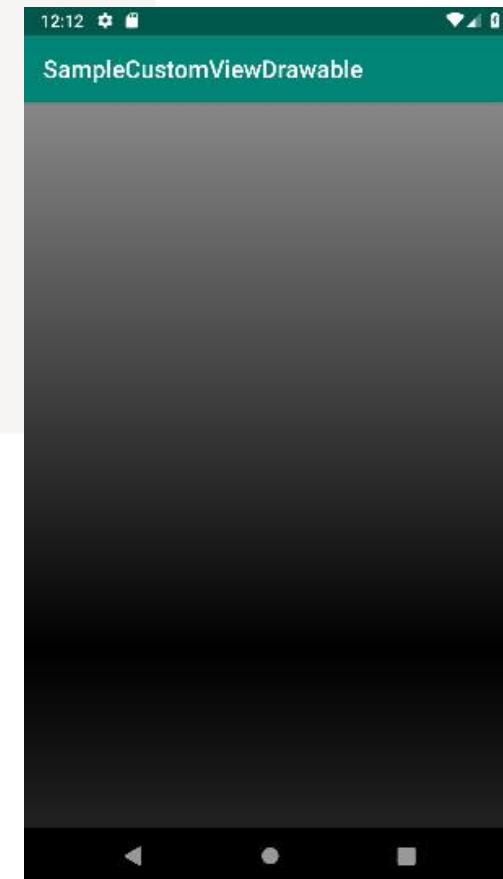
}
```



메인 액티비티에서 뷰 추가

참조파일 SampleCustomViewDrawable>/app/java/org.techtown.graphics.custom.drawable/MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        CustomViewDrawable view = new CustomViewDrawable(this);  
        setContentView(view);  
    }  
}
```





선 그리기를 위한 속성

구분	설명
setStrokeWidth	Stroke의 폭을 설정합니다.
setStrokeCap	Stroke의 시작과 끝 부분의 모양을 설정합니다. 설정할 수 있는 값은 Cap.BUTT, Cap.ROUND, Cap.SQUARE입니다. 디폴트 값은 Cap.BUTT입니다.
setStrokeJoin	Stroke의 꼭짓점 부분에 사용되는 연결 모양을 설정합니다. 설정할 수 있는 값은 Join.MITER, Join.ROUND, Join.BEVEL입니다. 디폴트 값은 Join.MITER입니다.
setStrokeMiter	Stroke 폭의 절반에 대한 Miter 길이의 비율 제한을 설정합니다.



onDraw 메서드 안에서 Path 사용

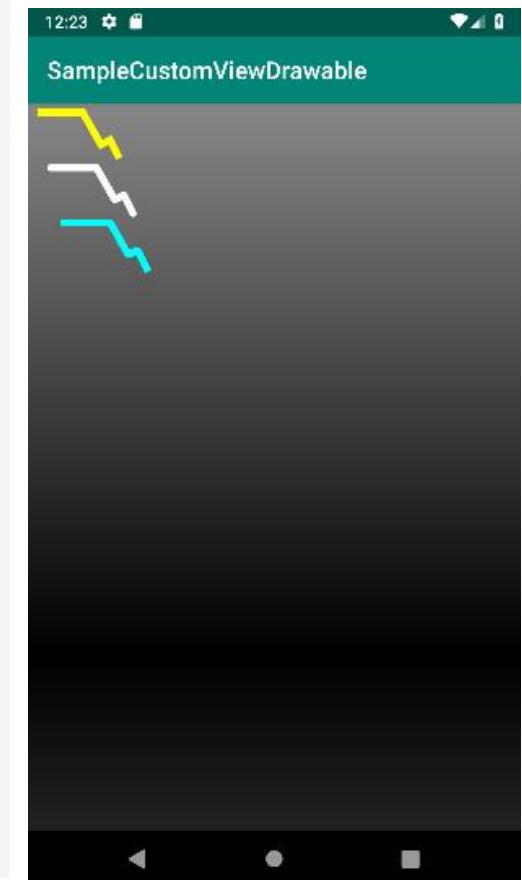
참조파일 SampleCustomViewDrawable>/app/java/org.techtown.graphics.custom.drawable/CustomViewDrawable.java

중략...

```
protected void onDraw(Canvas canvas) {  
    super.onDraw(canvas);  
  
    upperDrawable.draw(canvas);  
    lowerDrawable.draw(canvas);  
  
    Paint pathPaint = new Paint();  
    pathPaint.setAntiAlias(true);  
    pathPaint.setColor(Color.YELLOW);  
    pathPaint.setStyle(Paint.Style.STROKE);  
    pathPaint.setStrokeWidth(16.0F);  
    pathPaint.setStrokeCap(Paint.Cap.BUTT);  
    pathPaint.setStrokeJoin(Paint.Join.MITER);  
  
    Path path = new Path();  
    path.moveTo(20, 20);  
    path.lineTo(120, 20);  
    path.lineTo(160, 90);  
    path.lineTo(180, 80);  
    path.lineTo(200, 120);  
  
    canvas.drawPath(path, pathPaint);→③ Path 객체 그리기
```

① Cap.BUTT와 Join.MITER를 페인트 객체에
적용

② Path 객체 생성





3.

비트맵 이미지 사용하기



더블 버퍼링

- ▶ 뷰 위에 그래픽을 그린 후 다시 그리기를 할 때 깜박임 현상 발생
- ▶ 좌표값을 이용해 그래픽을 그리는 시간을 줄이기 위해 메모리 상의 비트맵 객체에 미리 그린 후 비트맵 객체를 뷰에 그려주는 방식



drawBitmap()
←





새로운 뷰 클래스 정의하기

참조파일 SampleCustomViewImage>/app/java/org.techtown.graphics.custom.image/CustomViewImage.java

```
public class CustomViewImage extends View { → ① View를 상속하여 새로운 뷰 정의

    private Bitmap cacheBitmap; → ② 메모리에 만들어질 Bitmap 객체 선언
    private Canvas cacheCanvas; → ③ 메모리에 만들어질 Bitmap 객체에 그리기 위한 Canvas 객체 선언
    private Paint mPaint;

    public CustomViewImage(Context context) {
        super(context);

        init(context);
    }

    public CustomViewImage(Context context, AttributeSet attrs) {
        super(context, attrs);

        init(context);
    }

    private void init(Context context) {
        mPaint = new Paint();
    }
}
```



메모리의 비트맵 객체에 그리기

```
protected void onSizeChanged(int w, int h, int oldw, int oldh) {  
    createCacheBitmap(w, h);  
    testDrawing();  
}  
  
private void createCacheBitmap(int w, int h) {  
    cacheBitmap = Bitmap.createBitmap(w, h, Bitmap.Config.ARGB_8888);  
    cacheCanvas = new Canvas();  
    cacheCanvas.setBitmap(cacheBitmap);  
}  
  
private void testDrawing() {  
    cacheCanvas.drawColor(Color.WHITE);  
    mPaint.setColor(Color.RED);  
    cacheCanvas.drawRect(100, 100, 200, 200, mPaint);  
}  
  
protected void onDraw(Canvas canvas) {  
    if (cacheBitmap != null) {  
        canvas.drawBitmap(cacheBitmap, 0, 0, null);  
    }  
}
```

④ 뷰가 화면에 보이기 전에
Bitmap 객체 만들고
그 위에 그리기

⑤ 메모리에 Bitmap
객체를 만들고
Canvas 객체 설정

⑥ 빨간 사각형 그리기

⑦ 메모리의 Bitmap을 이용해 화면에 그리기



메인 액티비티에 뷰 추가

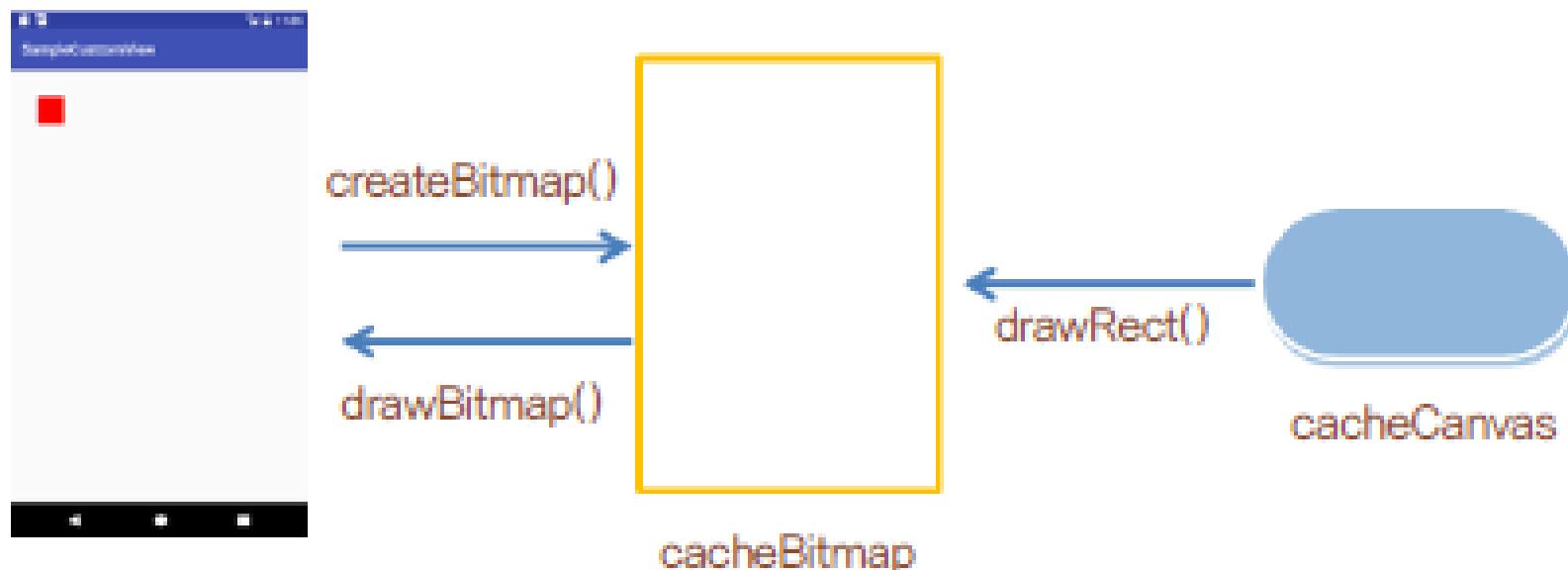
참조파일 SampleCustomViewImage>/app/java/org.techtown.graphics.custom.image/MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        CustomViewImage view = new CustomViewImage(this);  
        setContentView(view);  
    }  
}
```





더블버퍼링 그리기 개념도





이미지를 비트맵 객체로 만드는 방법

파일에서 읽기	파일 패스를 지정하면 해당 위치의 이미지 파일을 읽어옵니다. → <code>public static Bitmap decodeFile(String pathName)</code>
리소스에서 읽기	리소스에 저장한 이미지 파일을 읽어옵니다. → <code>public static Bitmap decodeResource(Resources res, int id)</code>
바이트 배열에서 읽기	바이트 배열로 되어 있는 이미지 파일을 읽어옵니다. → <code>public static Bitmap decodeByteArray(byte[] data, int offset, int length)</code>
스트림에서 읽기	입력 스트림에서 이미지 파일을 읽어옵니다. → <code>public static Bitmap decodeStream(InputStream is)</code>



리소스에서 로딩하기

참조파일 SampleCustomViewImage>/app/java/org.techtown.graphics.custom.image/CustomViewImage.java

중략...

```
private void testDrawing() {
    cacheCanvas.drawColor(Color.WHITE);

    mPaint.setColor(Color.RED);
    cacheCanvas.drawRect(100, 100, 200, 200, mPaint);

    Bitmap srcImg = BitmapFactory.decodeResource(getResources(), R.drawable.waterdrop); ①
    cacheCanvas.drawBitmap(srcImg, 30, 30, mPaint);

    Matrix horInverseMatrix = new Matrix();
    horInverseMatrix.setScale(-1, 1);
    Bitmap horInverseImg = Bitmap.createBitmap(srcImg, 0, 0,
        srcImg.getWidth(), srcImg.getHeight(), horInverseMatrix, false); ②
    cacheCanvas.drawBitmap(horInverseImg, 30, 130, mPaint);

    Matrix verInverseMatrix = new Matrix();
    verInverseMatrix.setScale(1, -1);
    Bitmap verInverseImg = Bitmap.createBitmap(srcImg, 0, 0,
        srcImg.getWidth(), srcImg.getHeight(), verInverseMatrix, false); ③
    cacheCanvas.drawBitmap(verInverseImg, 30, 230, mPaint);
}
```



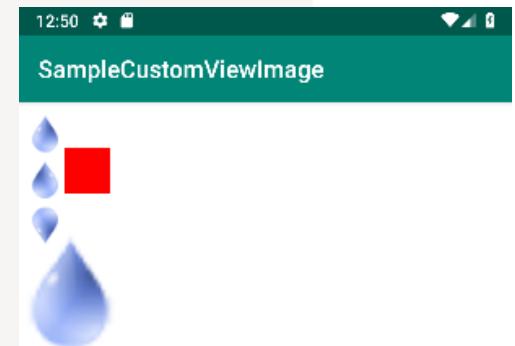
메모리의 비트맵 객체에 그리기

참조파일 SampleCustomViewImage>/app/java/org.techtown.graphics.custom.image/CustomViewImage.java

중략...

```
mPaint.setMaskFilter(new BlurMaskFilter(10, BlurMaskFilter.Blur.NORMAL));
Bitmap scaledImg = Bitmap.createScaledBitmap(srcImg,
                                             srcImg.getWidth()*3, srcImg.getHeight()*3, false);
cacheCanvas.drawBitmap(scaledImg, 30, 300, mPaint);
```

중략...





4.

페인트보드 만들기



뷰를 상속해 PaintBoard 클래스 정의

참조파일 SamplePaint>/app/java/org.techtown.paint/PaintBoard.java

```
public class PaintBoard extends View {

    Canvas mCanvas;
    Bitmap mBitmap;
    Paint mPaint;

    int lastX;
    int lastY;

    public PaintBoard(Context context) {
        super(context);

        init(context);
    }

    public PaintBoard(Context context, AttributeSet attrs) {
        super(context, attrs);

        init(context);
    }
}
```



onTouchEvent() 메소드 정의 (계속)

```
public boolean onTouchEvent(MotionEvent event) {  
    int action = event.getAction();  
  
    int X = (int) event.getX();  
    int Y = (int) event.getY();  
  
    switch (action) {  
        case MotionEvent.ACTION_UP:  
            lastX = -1;  
            lastY = -1;  
  
            break;  
  
        case MotionEvent.ACTION_DOWN:  
            if (lastX != -1) {  
                if (X != lastX || Y != lastY) {  
                    mCanvas.drawLine(lastX, lastY, X, Y, mPaint);  
                }  
            }  
  
            lastX = X;  
            lastY = Y;  
  
            break;  
    }  
}
```



onTouchEvent() 메소드 정의 (계속)

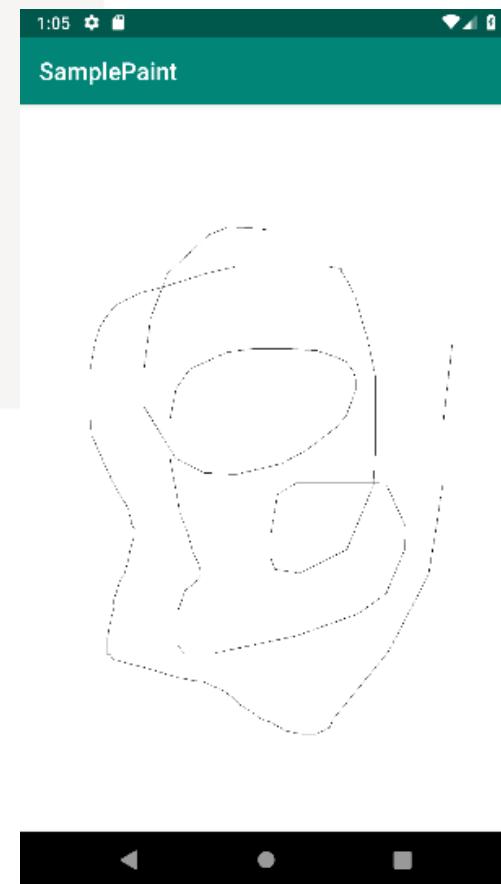
```
        case MotionEvent.ACTION_MOVE:  
            if (lastX != -1) {  
                mCanvas.drawLine(lastX, lastY, X, Y, mPaint);  
            }  
  
            lastX = X;  
            lastY = Y;  
  
            break;  
        }  
  
        invalidate();  
  
        return true;  
    }  
}
```



메인 액티비티에 새로운 뷰 추가

참조파일 SamplePaint>/app/java/org.techtown.paint/MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        PaintBoard view = new PaintBoard(this);  
        setContentView(view);  
    }  
}
```





좀 더 부드러운 그리기

```
private void init(Context context) {  
    mPaint = new Paint();  
    mPaint.setAntiAlias(true);  
    mPaint.setColor(Color.BLACK);  
    mPaint.setStyle(Paint.Style.STROKE);  
    mPaint.setStrokeJoin(Paint.Join.ROUND);  
    mPaint.setStrokeCap(Paint.Cap.ROUND);  
    mPaint.setStrokeWidth(3.0F);  
  
    this.lastX = -1;  
    this.lastY = -1;  
}  
  
protected void onSizeChanged(int w, int h, int oldw, int oldh) {  
    Bitmap img = Bitmap.createBitmap(w, h, Bitmap.Config.ARGB_8888);  
    Canvas canvas = new Canvas();  
    canvas.setBitmap(img);  
    canvas.drawColor(Color.WHITE);  
  
    mBitmap = img;  
    mCanvas = canvas;  
}
```



좀 더 부드러운 그리기

```
case MotionEvent.ACTION_DOWN:  
    rect = touchDown(event);  
    if (rect != null) {  
        invalidate(rect);  
    }  
  
    return true;  
case MotionEvent.ACTION_MOVE:  
    rect = touchMove(event);  
    if (rect != null) {  
        invalidate(rect);  
    }  
}
```



좀 더 부드러운 그리기

```
private Rect touchDown(MotionEvent event) {  
    float x = event.getX();  
    float y = event.getY();  
  
    lastX = x;  
    lastY = y;  
  
    Rect mInvalidRect = new Rect();  
    mPath.moveTo(x, y);  
  
    final int border = mInvalidateExtraBorder;  
    mInvalidRect.set((int) x - border, (int) y - border, (int) x + border, (int) y + border);  
    mCurveEndX = x;  
    mCurveEndY = y;  
  
    mCanvas.drawPath(mPath, mPaint);  
  
    return mInvalidRect;  
}
```



좀 더 부드러운 그리기

```
private Rect processMove(MotionEvent event) {
    final float x = event.getX();
    final float y = event.getY();

    final float dx = Math.abs(x - lastX);
    final float dy = Math.abs(y - lastY);

    Rect mInvalidRect = new Rect();
    if (dx >= TOUCH_TOLERANCE || dy >= TOUCH_TOLERANCE) {
        final int border = mInvalidateExtraBorder;
        mInvalidRect.set((int) mCurveEndX - border, (int) mCurveEndY - border,
                         (int) mCurveEndX + border, (int) mCurveEndY + border);

        float cX = mCurveEndX = (x + lastX) / 2;
        float cY = mCurveEndY = (y + lastY) / 2;

        mPath.quadTo(lastX, lastY, cX, cY);

        mInvalidRect.union((int) lastX - border, (int) lastY - border,
                           (int) lastX + border, (int) lastY + border);

        mInvalidRect.union((int) cX - border, (int) cY - border,
                           (int) cX + border, (int) cY + border);

        lastX = x;
        lastY = y;
    }

    mCanvas.drawPath(mPath, mPaint);
}
```





5.

멀티터치 이미지 뷰어 만들기



화면 레이아웃 만들기

참조파일 SampleMultiTouch>/app/res/layout/activity_main.xml

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

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="두 손가락을 이용해 터치해 보세요." />

    <LinearLayout
        android:id="@+id/container"
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >
        </LinearLayout>

    </LinearLayout>
```



이미지를 보여줄 뷰 정의

참조파일 SampleMultiTouch>/app/java/org.techtown.multitouch/ImageDisplayView.java

```
public class ImageDisplayView extends View implements View.OnTouchListener {  
    private static final String TAG = "ImageDisplayView";  
  
    Context mContext;  
    Canvas mCanvas;  
    Bitmap mBitmap;  
    Paint mPaint;  
  
    int lastX;  
    int lastY;  
  
    Bitmap sourceBitmap;  
  
    Matrix mMMatrix;  
  
    float sourceWidth = 0.0F;  
    float sourceHeight = 0.0F;  
  
    float bitmapCenterX;  
    float bitmapCenterY;
```

① 뷰를 상속하면서 OnTouchListener
인터페이스를 구현하는 클래스 정의



터치 이벤트 처리

```
case MotionEvent.ACTION_MOVE: → ⑦ 손가락으로 움직일 때의 기능 추가
    if (pointerCount == 1) {
        if (isScrolling) {
            return true;
        }

        float curX = ev.getX();
        float curY = ev.getY();

        if (startX == 0.0F) {
            startX = curX;
            startY = curY;

            return true;
        }
    }
}
```



터치 이벤트 처리

```
float offsetX = startX - curX;
float offsetY = startY - curY;

if (oldPointerCount == 2) {

} else {
    Log.d(TAG, "ACTION_MOVE : " + offsetX + ", " + offsetY);
    if (totalScaleRatio > 1.0F) {
        moveImage(-offsetX, -offsetY); —→ ⑧ 한 손가락으로 움직이고 있을 때는
    }                                moveImage() 메서드 호출

    startX = curX;
    startY = curY;
}

} else if (pointerCount == 2) {

    중략...

    scaleImage(outScaleRatio); —→ ⑨ 두 손가락으로 움직이고 있을 때는
    scaleImage() 메서드 호출

    중략...

}

oldPointerCount = pointerCount;
```



확대/이동을 위한 메서드 정의

```
private void scaleImage(float inScaleRatio) {  
    Log.d(TAG, "scaleImage() called : " + inScaleRatio);  
  
    mMatrix.postScale(inScaleRatio, inScaleRatio, bitmapCenterX, bitmapCenterY);  
    mMatrix.postRotate(0);  
  
    totalScaleRatio = totalScaleRatio * inScaleRatio;  
  
    redraw();  
}
```

⑪ 매트릭스 객체를 사용해 이미지 크기 변경

```
private void moveImage(float offsetX, float offsetY) {  
    Log.d(TAG, "moveImage() called : " + offsetX + ", " + offsetY);  
  
    mMatrix.postTranslate(offsetX, offsetY);  
  
    redraw();  
}
```

⑫ 매트릭스 객체를 사용해 이미지 이동



메인 액티비티에 추가

참조파일 SampleMultiTouch>/app/java/org.techtown.multitouch/MainActivity.java

```
public class MainActivity extends AppCompatActivity {

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

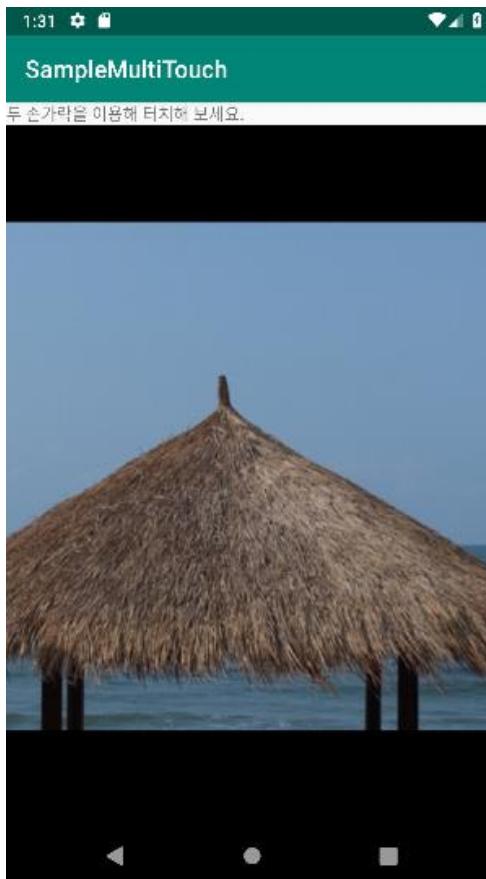
        LinearLayout container = findViewById(R.id.container);
        Resources res = getResources();
        Bitmap bitmap = BitmapFactory.decodeResource(res, R.drawable.beach);

        ImageDisplayView view = new ImageDisplayView(this); → ❶ ImageDisplayView 객체 생성하기
        view.setImageData(bitmap);
        LinearLayout.LayoutParams params = new LinearLayout.LayoutParams(
            LinearLayout.LayoutParams.MATCH_PARENT,
            LinearLayout.LayoutParams.MATCH_PARENT);

        container.addView(view, params); → ❷ XML 레이아웃에 들어있는 LinearLayout 안에
    }                                         ImageDisplayView 객체 추가하기
}
```



앱 실행



Logcat

```
Emulator Nexus_5X_API_28 Andri org.techtown.multitouch (14084) Verbose Q+ 
2018-12-05 22:32:52.084 14084-14084/org.techtown.multitouch D/ImageView: scaleImage() called : 0.9508
2018-12-05 22:32:52.156 14084-14084/org.techtown.multitouch D/ImageView: Pointer Count : 2
2018-12-05 22:32:52.156 14084-14084/org.techtown.multitouch D/ImageView: Distance : 202.46715, ScaleR
2018-12-05 22:32:52.156 14084-14084/org.techtown.multitouch D/ImageView: scaleImage() called : 0.9552
2018-12-05 22:32:52.214 14084-14084/org.techtown.multitouch D/ImageView: Pointer Count : 2
2018-12-05 22:32:52.214 14084-14084/org.techtown.multitouch D/ImageView: Distance : 183.04942, ScaleR
2018-12-05 22:32:52.214 14084-14084/org.techtown.multitouch D/ImageView: scaleImage() called : 0.9547
2018-12-05 22:32:52.214 14084-14084/org.techtown.multitouch D/ImageView: Pointer Count : 2
```

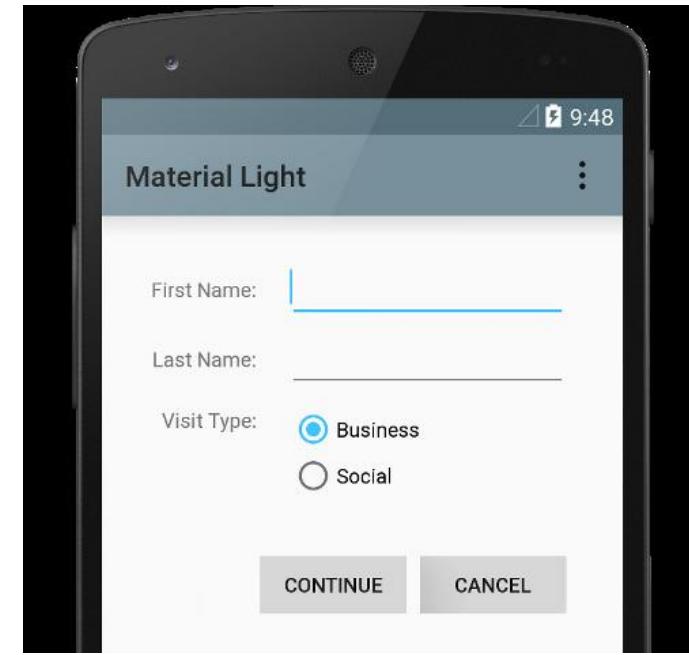
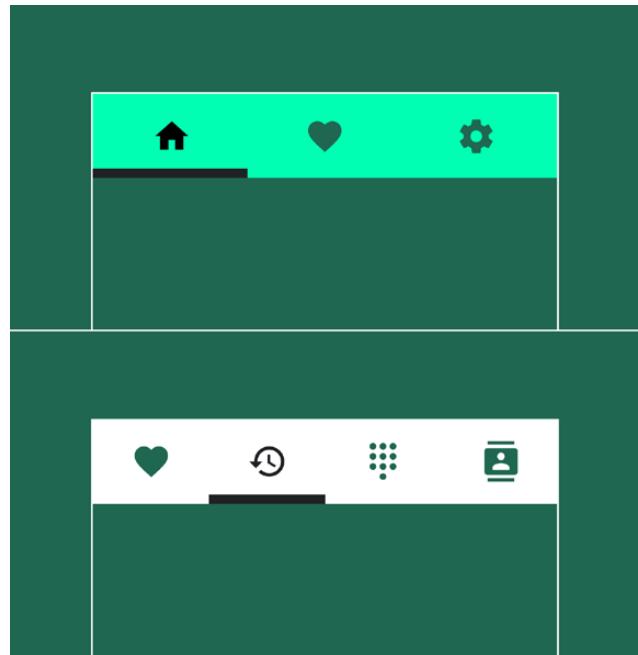


6.

머티리얼 디자인의 개념 알아두기

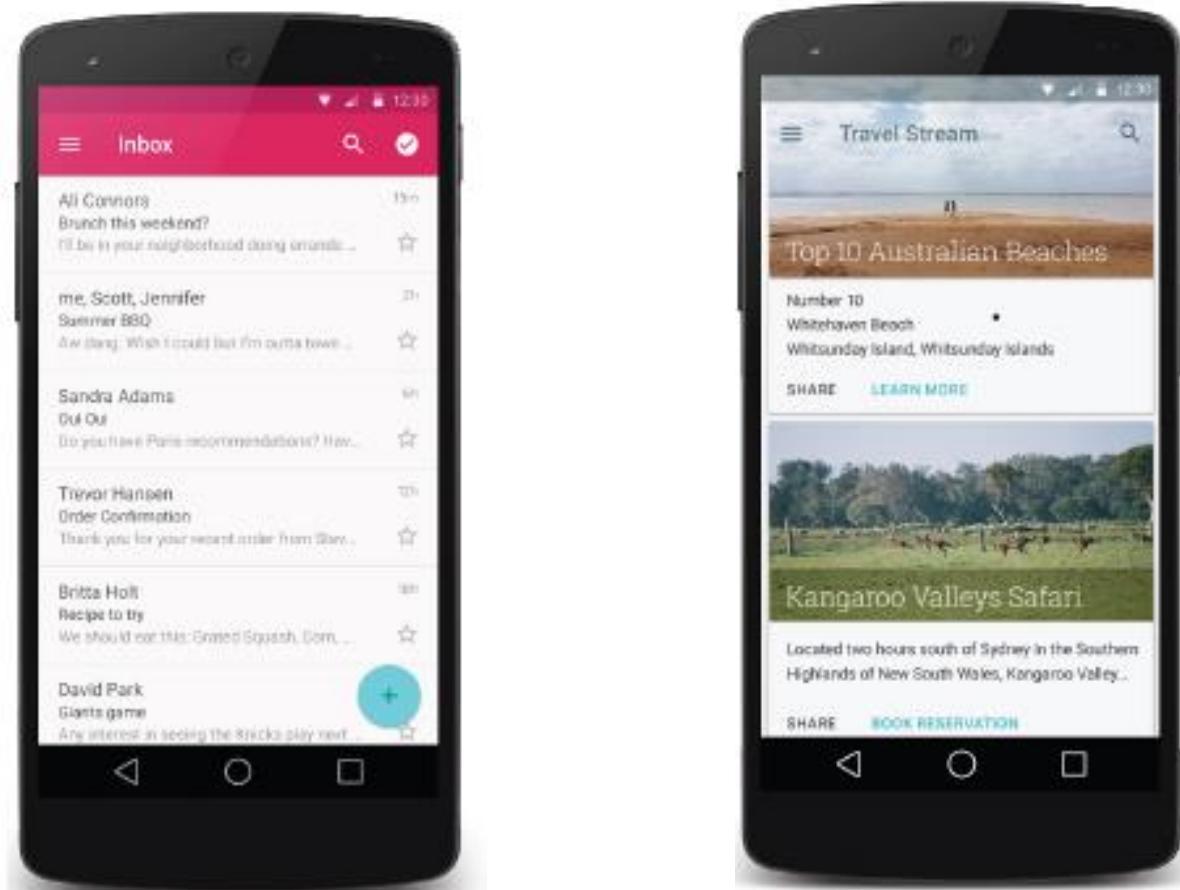


직관적인 디자인 트렌드





리스트와 카드의 모양





멀티미디어 기능에 대한 이해



- 1 카메라로 사진 찍어 저장하기
- 2 화면에 카메라 미리보기 넣기
- 3 음악 파일 재생하기
- 4 동영상 재생하기
- 5 오디오 녹음하여 저장하기
- 6 동영상 녹화하기
- 7 유튜브 영상 재생하기



카메라로 사진을 찍고 싶어요

- 카메라로 사진 찍어 저장하기
- 화면에 카메라 미리보기 넣기



음악 파일이나 동영상은 어떻게 재생하나요?

- 음악 파일 재생하기
- 동영상 재생하기



음성을 직접 녹음하고 싶은데요?

- 오디오 녹음하여 저장하기
- 동영상 녹화하기



유튜브 영상을 재생할 수 있나요?

- 유튜브 영상 재생하기

사진 찍기



음악 파일과 동영상 재생/녹화



유튜브 영상 재생하기





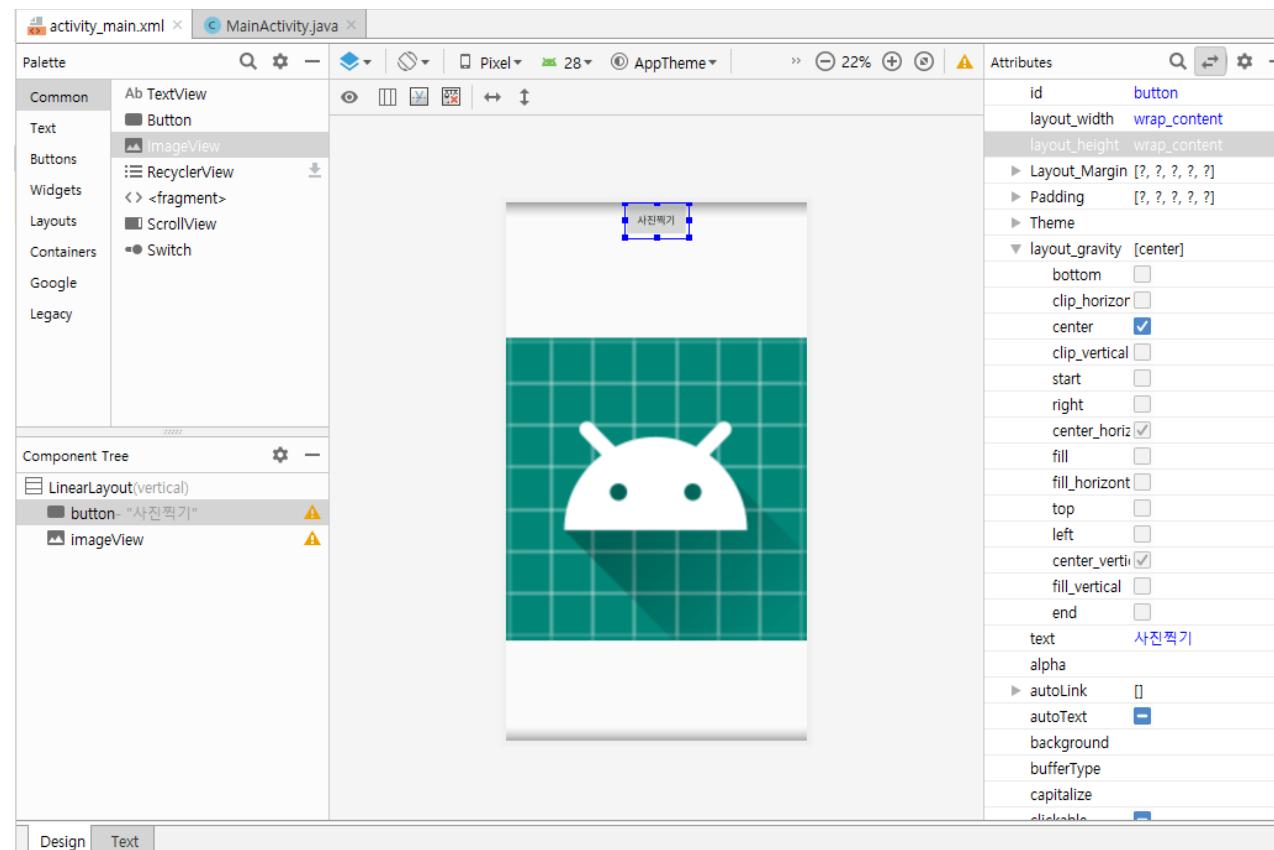
1.

카메라로 사진 찍어 저장하기



카메라로 사진찍기

- 단말의 카메라 앱을 띄워 사진을 찍은 후 그 결과물을 가져와 처리함
- 화면 레이아웃에는 버튼과 이미지뷰 배치





카메라로 사진찍기

- 버튼 클릭했을 때 호출되는 메서드에서는 인텐트를 이용해 카메라 앱 띄움

```
public void takePicture() {  
    if (file == null) {  
        file = createFile();  
    }  
  
    Uri fileUri = FileProvider.getUriForFile(this, "org.techtown.capture.intent.fileprovider", file);  
    Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);  
    intent.putExtra(MediaStore.EXTRA_OUTPUT, fileUri);  
    if (intent.resolveActivity(getPackageManager()) != null) {  
        startActivityForResult(intent, 101); → ② 사진 찍기 화면 띄우기  
    }  
}
```

① File 객체로부터
Uri 객체 만들기



카메라로 사진찍기

- 사진을 찍으면 파일에 저장되도록 해야 함
- /app/res/xml/external.xml 파일 생성

참조파일 SampleCaptureIntent>/app/res/xml/external.xml

```
<?xml version="1.0" encoding="utf-8"?>
<paths xmlns:android="http://schemas.android.com/apk/res/android">
    <external-path name="sdcard" path="/" />
</paths>
```



카메라로 사진찍기

- 매니페스트 안에 파일 접근을 위한 Provider 태그 추가

참조파일 SampleCaptureIntent>/app/manifests/AndroidManifest.xml

|중략...

```
<application>  
    <activity>  
        <provider  
            android:name="android.support.v4.content.FileProvider"  
            android:authorities="org.techtown.capture.intent.fileprovider"  
            android:exported="false"  
            android:grantUriPermissions="true">  
            <meta-data  
                android:name="android.support.FILE_PROVIDER_PATHS"  
                android:resource="@xml/external" />  
        </provider>  
    </application>
```



카메라로 사진찍기

- 인텐트에 부가 데이터로 사진 파일의 위치 정보를 넣어 줌

참조파일 SampleCaptureIntent>/app/java/org.techtown.capture.intent/MainActivity.java

중략...

```
Uri fileUri = FileProvider.getUriForFile(this, "org.techtown.capture.intent.fileprovider", file);
Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
intent.putExtra(MediaStore.EXTRA_OUTPUT, fileUri);
```

중략...



카메라로 사진찍기

- 사진 찍은 결과를 응답으로 받아 처리

참조파일 SampleCaptureIntent>/app/java/org.techtown.capture.intent/MainActivity.java

중략...

```
@Override  
public void onClick(View v) {  
    takePicture();  
}  
});  
}  
  
  
@Override  
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  
    super.onActivityResult(requestCode, resultCode, data);  
  
    if (requestCode == 101 && resultCode == RESULT_OK) {  
        BitmapFactory.Options options = new BitmapFactory.Options();→ ① 이미지 파일을 Bitmap  
        options.inSampleSize = 8;                                객체로 만들기  
        Bitmap bitmap = BitmapFactory.decodeFile(file.getAbsolutePath(), options);  
  
        imageView.setImageBitmap(bitmap);→ ② 이미지뷰에 Bitmap 설정하기  
    }  
}
```



카메라로 사진찍기

- 매니페스트에 권한 추가

참조파일 SampleCaptureIntent>/app/manifests/AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="org.techtown.capture.intent">

    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

    <uses-feature android:name="android.hardware.camera"
        android:required="true" />
```

중략...



카메라로 사진찍기

- 위험 권한 부여를 위한 라이브러리 추가

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

중략...

```
allprojects {  
    repositories {  
        maven { url 'https://jitpack.io' }  
    }  
}  
  
dependencies {  
    중략...  
  
    implementation 'com.github.pedroS694:AutoPermissions:1.0.3'  
}
```

중략...



카메라로 사진찍기

- 메인 액티비티에 위험 권한 부여를 위한 코드 추가

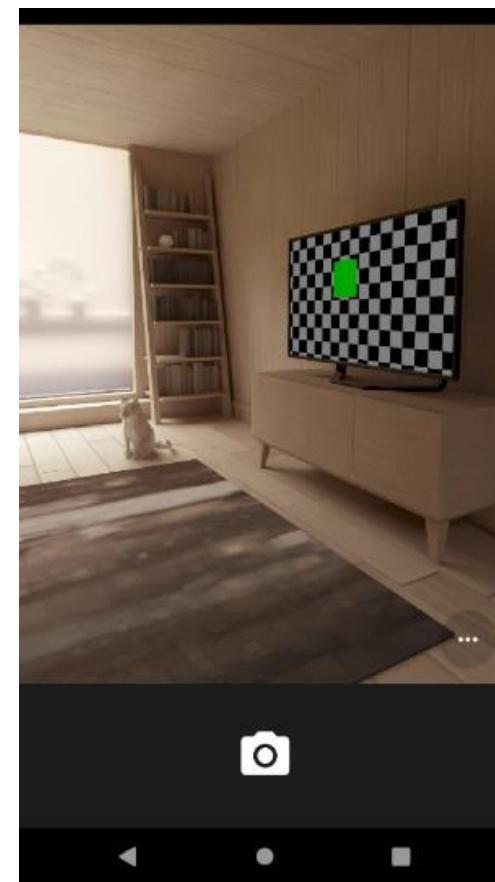
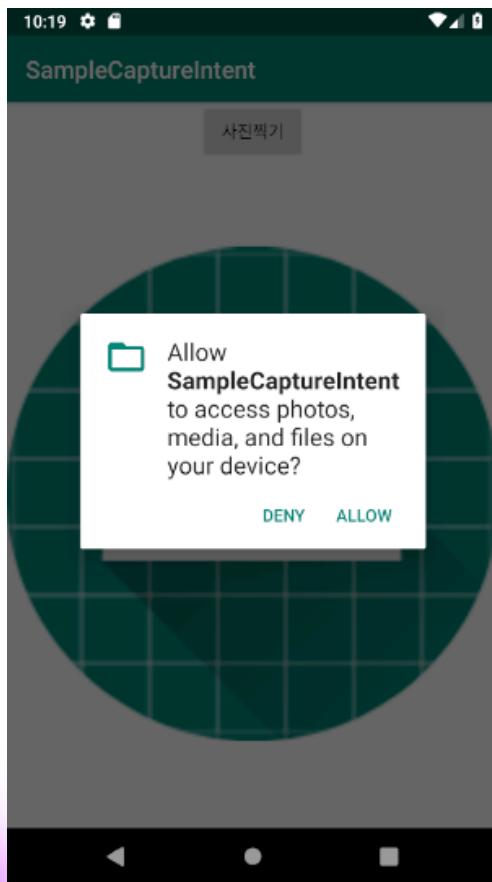
참조파일 SampleCaptureIntent>/app/java/org.techtown.capture.intent/MainActivity.java

```
public class MainActivity extends AppCompatActivity implements AutoPermissionsListener {  
  
    중략...  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        중략...  
  
        AutoPermissions.Companion.loadAllPermissions(this, 101);  
    }  
  
    중략...  
  
    @Override  
    public void onRequestPermissionsResult(int requestCode, String permissions[],  
                                         int[] grantResults) {  
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
        AutoPermissions.Companion.parsePermissions(this, requestCode, permissions, this);  
    }  
}
```



카메라로 사진찍기

- 앱 실행





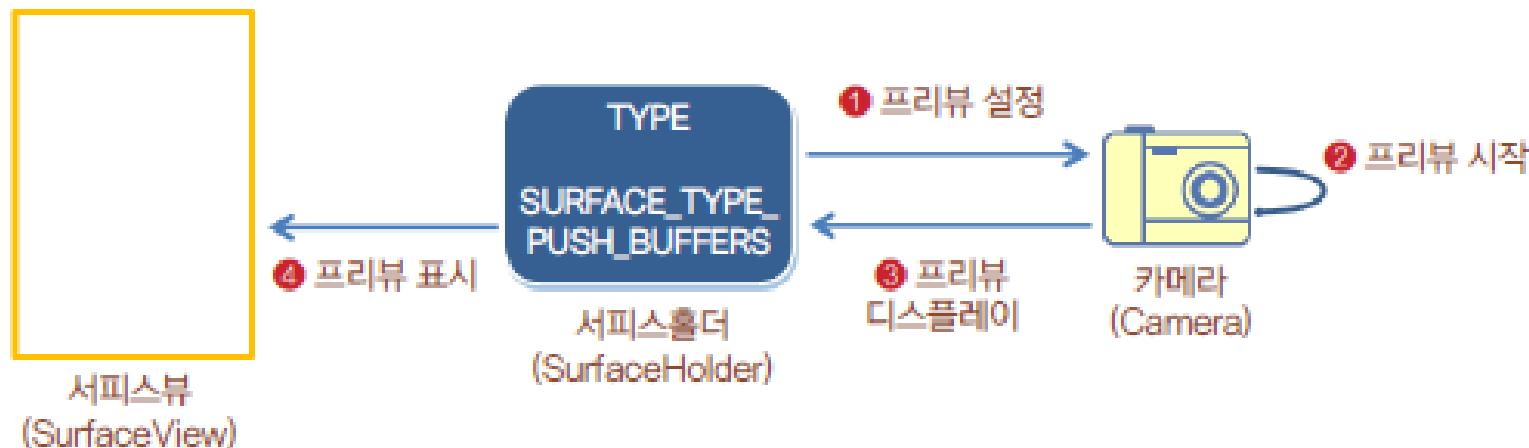
2.

화면에 카메라 미리보기 넣기



카메라 미리보기를 위한 서피스뷰

- 서피스뷰는 뷰 체계와 상관없이 만들어진 별개의 뷰
- 서피스뷰를 이용해 카메라 미리보기 표시





화면 레이아웃 구성

- 서피스뷰가 들어갈 공간 확보

참조파일 SampleCapture>/app/res/layout/activity_main.xml

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

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="사진 찍기" />

    <FrameLayout → 카메라 미리보기를 추가할 공간을 확보하기 위해 FrameLayout 태그 넣기
        android:id="@+id/previewFrame"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

    </FrameLayout>
</LinearLayout>
```



코드 추가

- 메인 액티비티에서 새로 만든 뷰를 화면에 추가
- 버튼 클릭 시 사진찍기 위한 takePicture 메서드 호출

참조파일 SampleCapture>/app/java/org.techtown.capture/MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    CameraSurfaceView cameraView;

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

        FrameLayout previewFrame = findViewById(R.id.previewFrame);
        cameraView = new CameraSurfaceView(this);
        previewFrame.addView(cameraView);

        Button button = findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                takePicture();
            }
        });
    }
}
```



코드 추가

- **takePicture** 메서드에서 화면 캡처를 위한 코드 실행

```
public void takePicture() {  
    cameraView.capture(new Camera.PictureCallback() { → ❶ CameraSurfaceView의  
        public void onPictureTaken(byte[] data, Camera camera) { capture() 메서드 호출하기  
            try {  
                Bitmap bitmap = BitmapFactory.decodeByteArray(data, 0, data.length); → ❷ 전달받은 바이트  
                String outUriStr = MediaStore.Images.Media.insertImage( 배열을 Bitmap  
                    getContentResolver(), 객체로 만들기  
                    bitmap,  
                    "Captured Image",  
                    "Captured Image using Camera.");  
  
                if (outUriStr == null) {  
                    Log.d("SampleCapture", "Image insert failed.");  
                    return;  
                } else {  
                    Uri outUri = Uri.parse(outUriStr);  
                    sendBroadcast(new Intent(Intent.ACTION_MEDIA_SCANNER_SCAN_FILE, outUri));  
                }  
            }  
        }  
    }  
}
```



CameraSurfaceView 정의

- 서피스뷰를 상속하여 새로운 클래스 정의

참조파일 SampleCapture>/app/java/org.techtown.capture/MainActivity.java

중략...

```
class CameraSurfaceView extends SurfaceView implements SurfaceHolder.Callback {  
    private SurfaceHolder mHolder;  
    private Camera camera = null;  
  
    public CameraSurfaceView(Context context) {  
        super(context);  
  
        mHolder = getHolder();  
        mHolder.addCallback(this);  
    }  
  
    public void surfaceCreated(SurfaceHolder holder) {  
        camera = Camera.open();  
  
        try {  
            camera.setPreviewDisplay(mHolder);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
  
    public void surfaceChanged(SurfaceHolder holder, int format, int width, int height) {  
        camera.startPreview();  
    }  
}
```



CameraSurfaceView 정의

- 기본적인 콘트롤은 Holder 객체에서 담당

```
public void surfaceChanged(SurfaceHolder holder, int format, int width, int height) {  
    camera.startPreview();  
}  
  
public void surfaceDestroyed(SurfaceHolder holder) {  
    camera.stopPreview();  
    camera.release();  
    camera = null;  
}  
  
public boolean capture(Camera.PictureCallback handler) {  
    if (camera != null) {  
        camera.takePicture(null, null, handler);  
        return true;  
    } else {  
        return false;  
    }  
}
```



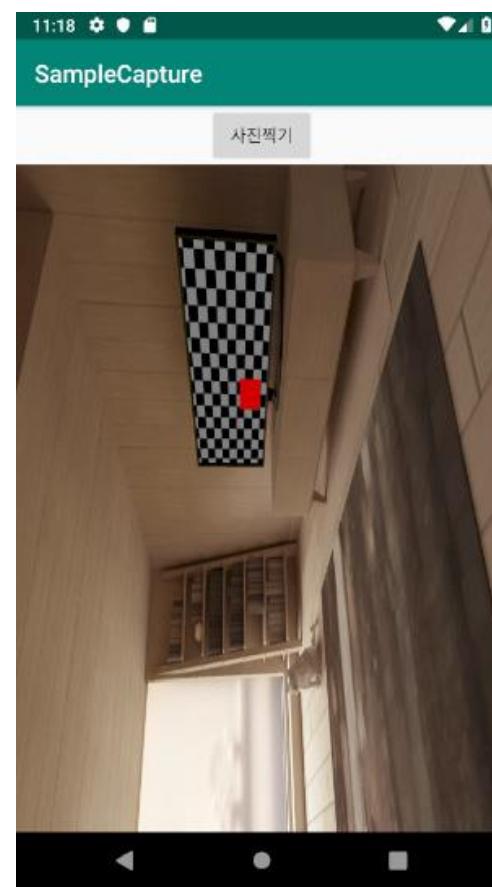
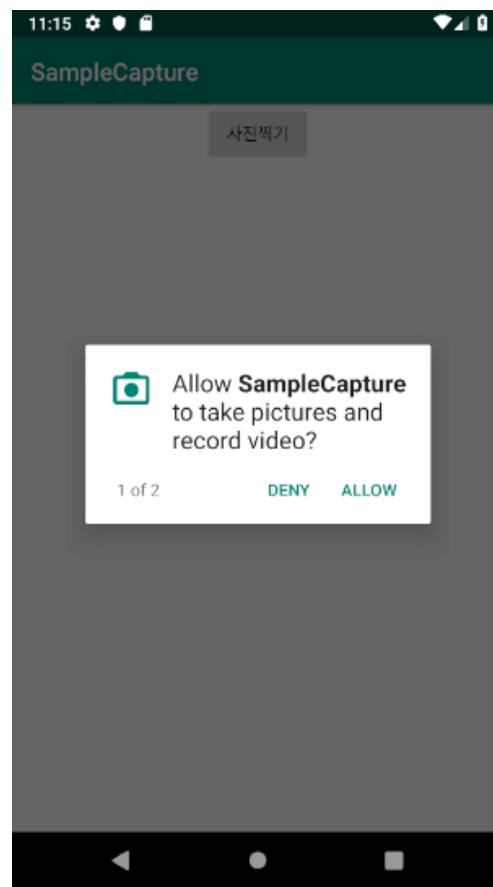
The diagram illustrates the scope of the code blocks using curly braces and red numbers:

- Brace 4 covers the entire `surfaceChanged` method.
- Brace 5 covers the entire `surfaceDestroyed` method.
- Brace 6 covers the entire `capture` method.



권한 부여 후 실행

- 위험 권한 부여 후에 앱 실행





화면 회전

- 화면 회전을 위한 코드 추가

```
Camera.CameraInfo info = new Camera.CameraInfo();
Camera.getCameraInfo(0, info);

WindowManager manager = (WindowManager) getSystemService(Context.WINDOW_SERVICE);
int rotation = manager.getDefaultDisplay().getRotation(); → ❶ 회전에 대한 정보 확인하기

int degrees = 0;
switch (rotation) {
    case Surface.ROTATION_0: degrees = 0; break;
    case Surface.ROTATION_90: degrees = 90; break;
    case Surface.ROTATION_180: degrees = 180; break;
    case Surface.ROTATION_270: degrees = 270; break;
}

int result;
if (info.facing == Camera.CameraInfo.CAMERA_FACING_FRONT) {
    result = (info.orientation + degrees) % 360;
    result = (360 - result) % 360;
} else {
    result = (info.orientation - degrees + 360) % 360;
}

camera.setDisplayOrientation(result); → ❷ 카메라 객체의 setDisplayOrientation() 메서드
}                                     호출하기
```



3.

음악 파일 재생하기



오디오 재생하기

오디오 재생을 위해 MediaPlayer 클래스 사용

MediaPlayer의 데이터 소스 지정 방법

(1) 웹 상의 위치 지정

- 미디어가 있는 웹 상의 위치를 URL로 지정

(2) 프로젝트 파일에 포함한 후 위치 지정

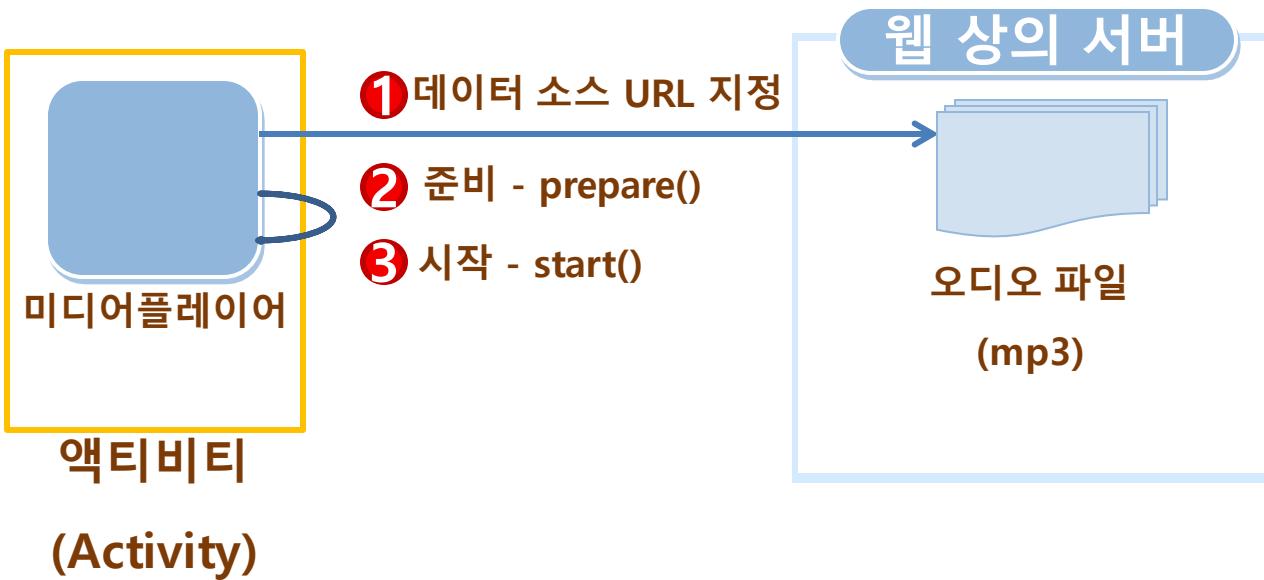
- 애플리케이션을 개발하여 배포하는 과정에서 프로젝트의 리소스 또는 애셋 (assets) 폴더에 넣은 후 그 위치를 지정

(3) 단말 SD카드에 넣은 후 위치 지정

- 단말에 넣어 둔 SD카드에 파일을 넣은 후 그 위치를 지정



음악파일 재생 과정

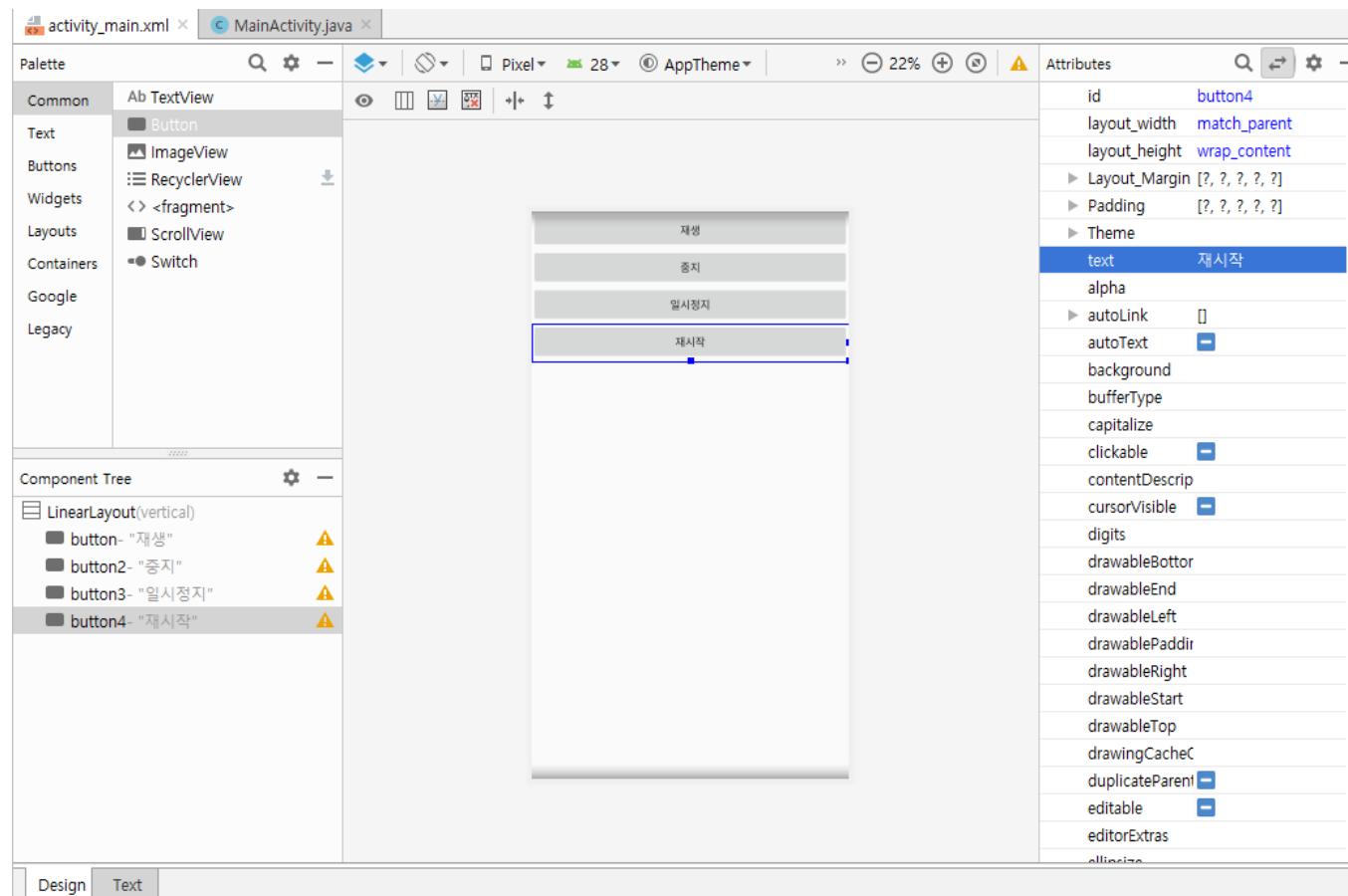


미디어플레이어를 이용한 음악파일 재생 과정



화면 레이아웃 구성

- 재생 기능 확인을 위한 버튼들 배치





메인 액티비티 만들기

- 미디어 플레이어 객체 생성
- 버튼 클릭 시 playAudio 메서드 호출

참조파일 SampleMediaPlayer>/app/java/org.techtown.audio.player/MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    public static final String AUDIO_URL = "https://sites.google.com/site/ubiaccessmobile/sample_audio.mp3";

    MediaPlayer mediaPlayer;
    int position = 0;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button button = findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
                playAudio(AUDIO_URL);
                Toast.makeText(getApplicationContext(),"음악 파일 재생 시작됨.", Toast.LENGTH_LONG).show();
            }
        });
    }
}
```



메인 액티비티 만들기

- playAudio 메서드에서는 미디어플레이어를 이용해 재생

```
private void playAudio(String url) {  
    killMediaPlayer();  
    try {  
        mediaPlayer = new MediaPlayer(); } ① MediaPlayer 객체 만들어 시작하기  
        mediaPlayer.setDataSource(url);  
        mediaPlayer.prepare();  
        mediaPlayer.start();  
    } catch(Exception e) {  
        e.printStackTrace();  
    }  
}
```



권한 추가 및 실행

- 인터넷 권한 추가 후 앱 실행

참조파일 SampleAudioPlayer>/app/manifests/AndroidManifest.xml

```
<uses-permission android:name="android.permission.INTERNET" />
```





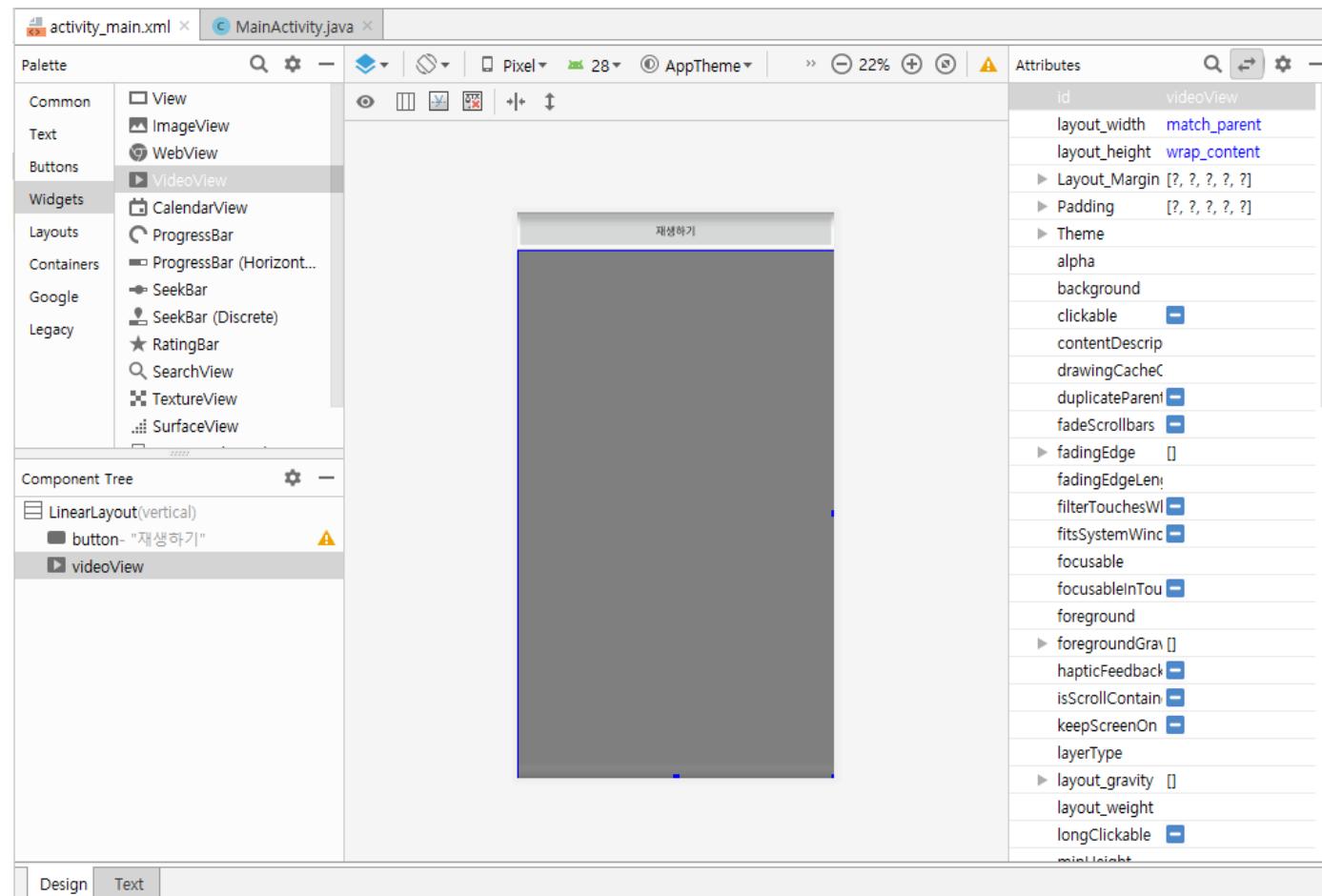
4.

동영상 재생하기



화면 레이아웃 만들기

- VideoView 위젯 추가





메인 액티비티에 코드 추가

- 버튼 누르면 VideoView의 start 메서드 호출

참조파일 SampleVideoPlayer>/app/java/org.techtown.video.player/MainActivity.java

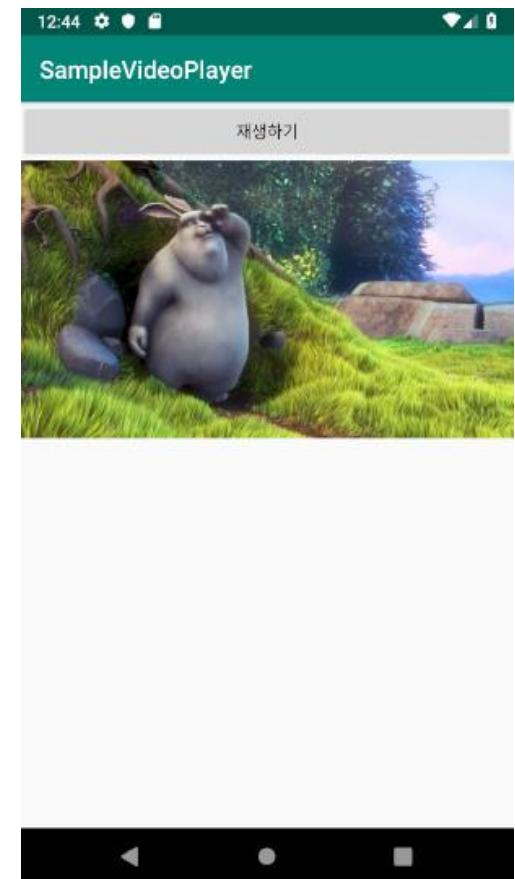
```
public class MainActivity extends AppCompatActivity {
    public static final String VIDEO_URL = "https://sites.google.com/site/ubiaccessmobile/sample_video.mp4";
    VideoView videoView;

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

        videoView = findViewById(R.id.videoView);

        MediaController mc = new MediaController(this);
        videoView.setMediaController(mc);→ ① VideoView에 MediaController 설정하기

        Button button = findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                videoView.setVideoURI(Uri.parse(VIDEO_URL));→ ② VideoView에 재생할 대상 설정하고
                videoView.requestFocus();                           재생 시작하기
                videoView.start();
            }
        });
    }
}
```





5.

오디오 녹음하여 저장하기



오디오 녹음하여 저장하기

오디오 녹음을 위한 과정

(1) 미디어리코더 객체 생성

- 오디오 녹음을 위해 미디어리코더 객체를 new 연산자를 이용하여 만듬

(2) 오디오 입력 및 출력 형식 설정

- 오디오 정보를 입력받을 데이터 소스와 함께 출력 형식을 설정함

(3) 오디오 인코더와 파일 지정

- 오디오 파일을 만들 때 필요한 인코더(Encoder)와 함께 파일 이름을 지정함

(4) 녹음 시작

- 녹음을 시작하면 오디오 파일이 만들어지고 인코딩된 바이트 스트림이 저장됨

(5) 매니페스트에 권한 설정

- 애플리케이션에서 녹음을 하려면 RECORD_AUDIO 권한이 있어야 하므로 매니페스트에 추가



화면 레이아웃 구성

The screenshot shows the Android Studio interface with the Design tab selected. The main area displays a vertical `LinearLayout` containing four buttons. The buttons are labeled "녹음시작", "녹음중지", "재생시작", and "재생중지". The "녹음시작" button is currently selected, highlighted with a blue border. The Attributes panel on the right lists various properties for the selected button, such as `id`, `layout_width`, `layout_height`, and `text`. The Component Tree panel on the left shows the structure of the layout.

id	button
layout_width	<code>match_parent</code>
layout_height	<code>wrap_content</code>
Layout_Margin	[?, ?, ?, ?, ?]
Padding	[?, ?, ?, ?, ?]
Theme	
text	녹음시작
alpha	
autoLink	
autoText	
background	
bufferType	
capitalize	
clickable	
contentDescrip	
cursorVisible	
digits	
drawableBottor	
drawableEnd	
drawableLeft	
drawablePaddir	
drawableRight	
drawableStart	
drawableTop	
drawingCacheC	
duplicateParent	
editable	
editorExtras	
ellipsize	



메인 액티비티 만들기

참조파일 SampleAudioRecorder>/app/java/org.techtown.audio.recorder/MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    MediaRecorder recorder;
    MediaPlayer player;

    String filename;

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

        Button button = findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startRecording();
            }
        });
    }

    private void startRecording() {
        if (recorder == null) {
            recorder = new MediaRecorder();
            recorder.setAudioSource(MediaRecorder.AudioSource.MIC);
            recorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
            recorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
            recorder.setOutputFile(filename);
        }
        recorder.start();
    }

    private void stopRecording() {
        if (recorder != null) {
            recorder.stop();
            recorder.release();
            recorder = null;
        }
    }

    private void playRecordedFile() {
        if (player == null) {
            player = new MediaPlayer();
            player.setDataSource(filename);
            player.prepare();
            player.start();
        }
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        stopRecording();
    }
}
```



메인 액티비티 만들기

- MediaRecorder 를 이용해 녹음

```
public void startRecording() {  
    if (recorder == null) {  
        recorder = new MediaRecorder();  
    }  
  
    recorder.setAudioSource(MediaRecorder.AudioSource.MIC);  
    recorder.setOutputFormat(MediaRecorder.OutputFormat.MPEG_4);  
    recorder.setAudioEncoder(MediaRecorder.AudioEncoder.DEFAULT);  
    recorder.setOutputFile(filename); }  
  
    try {  
        recorder.prepare(); } ② MediaRecorder 시작시키기  
        recorder.start(); } catch (Exception e) {  
            e.printStackTrace(); }  
    }  
  
    중략...
```

① MediaRecorder 설정하기

② MediaRecorder 시작시키기



메인 액티비티 만들기

- 녹음을 중지했을 때 저장

```
recorder.stop();
recorder.release();
recorder = null;

ContentValues values = new ContentValues(10);

values.put(MediaStore.MediaColumns.TITLE, "Recorded");
values.put(MediaStore.Audio.Media.ALBUM, "Audio Album");
values.put(MediaStore.Audio.Media.ARTIST, "Mike");
values.put(MediaStore.Audio.Media.DISPLAY_NAME, "Recorded Audio");
values.put(MediaStore.Audio.Media.IS_RINGTONE, 1);
values.put(MediaStore.Audio.Media.IS_MUSIC, 1);
values.put(MediaStore.MediaColumns.DATE_ADDED, System.currentTimeMillis()/1000);
values.put(MediaStore.MediaColumns.MIME_TYPE, "audio/mp4");
values.put(MediaStore.Audio.Media.DATA, filename);

Uri audioUri = getContentResolver().insert(MediaStore.Audio.Media.EXTERNAL_CONTENT_URI, values);
```



권한 추가

- SD 카드 접근과 오디오 녹음을 위한 권한 추가

참조파일 SampleAudioRecorder>/app/manifests/AndroidManifest.xml

중략...

```
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

중략...



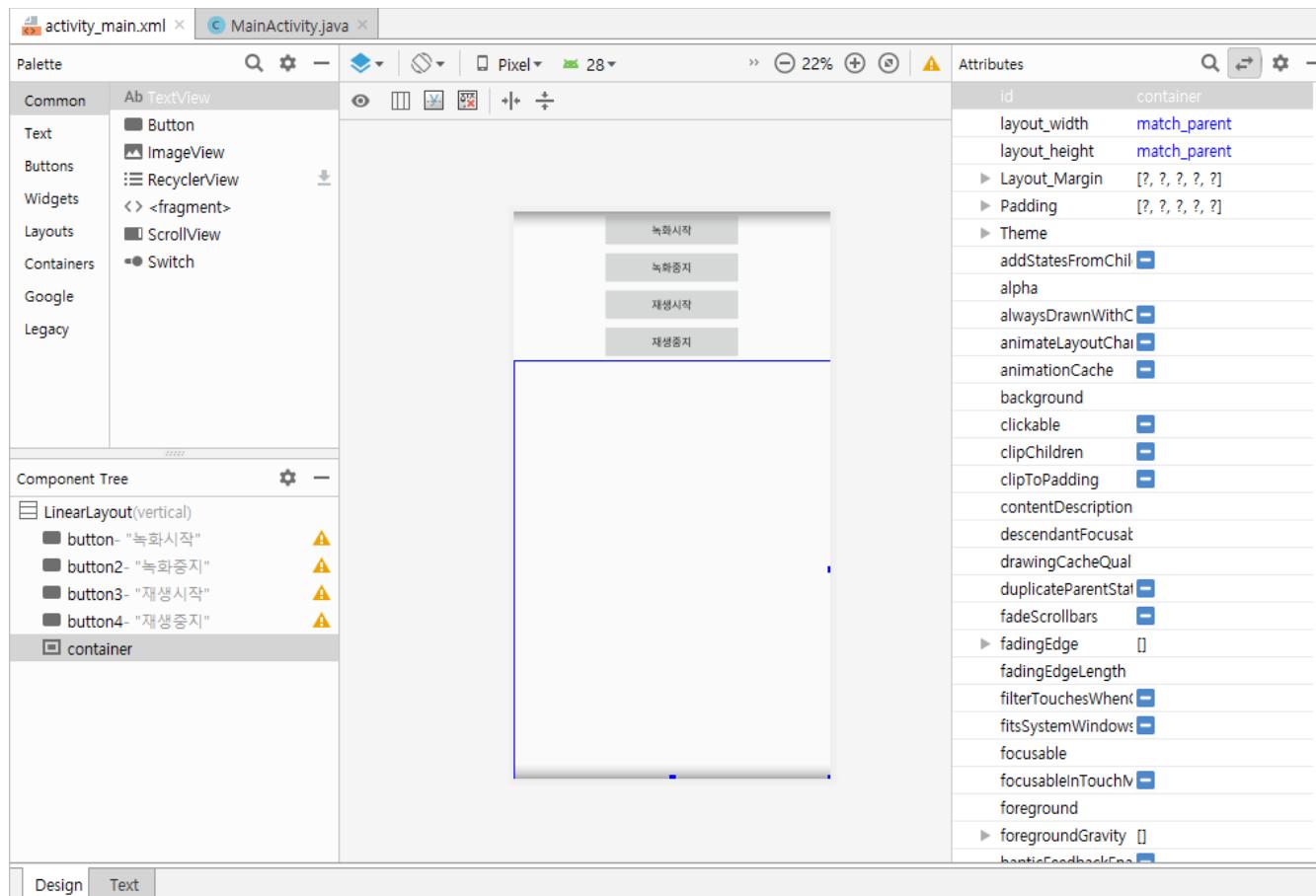
6.

동영상 녹화하기



화면 레이아웃 만들기

- 화면에 카메라 미리보기를 위한 공간 확보





메인 액티비티 작성

- 서피스뷰를 위한 코드 추가

참조파일 SampleVideoRecorder>/app/java/org.techtown.video.recorder/MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    MediaPlayer player;
    MediaRecorder recorder;

    String filename;

    SurfaceHolder holder;

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

        SurfaceView surface = new SurfaceView(this); → ❶ SurfaceView 객체 만들기
        holder = surface.getHolder();
        holder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);

        FrameLayout frame = findViewById(R.id.container);
        frame.addView(surface);
```



메인 액티비티 작성

- 버튼 클릭 시 녹화 시작

```
public void startRecording() {  
    if (recorder == null) {  
        recorder = new MediaRecorder();  
    }  
  
    recorder.setAudioSource(MediaRecorder.AudioSource.MIC);  
    recorder.setVideoSource(MediaRecorder.VideoSource.CAMERA);  
    recorder.setOutputFormat(MediaRecorder.OutputFormat.MPEG_4);  
    recorder.setAudioEncoder(MediaRecorder.AudioEncoder.DEFAULT);  
    recorder.setVideoEncoder(MediaRecorder.VideoEncoder.DEFAULT);  
    recorder.setOutputFile(filename);  
  
    recorder.setPreviewDisplay(holder.getSurface()); —→ ② MediaRecorder에 미리보기 화면을  
                                                보여줄 객체 설정하기  
    try {  
        recorder.prepare();  
        recorder.start();  
    } catch (Exception e) {  
        e.printStackTrace();  
  
        recorder.release();  
        recorder = null;  
    }  
}
```



메인 액티비티 작성

- 미디어 플레이어 객체에 서피스홀더 객체 설정

참조파일 SampleVideoRecorder>/app/java/org.techtown.video.recorder/MainActivity.java

중략...

```
public void startPlay() {
    if (player == null) {
        player = new MediaPlayer();
    }

    try {
        player.setDataSource(filename);
        player.setDisplay(holder);
        player.prepare();
        player.start();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

중략...



권한 추가 후 실행

- 카메라 등 권한 추가 및 위험 권한 부여를 위한 코드 추가

참조파일 SampleVideoRecorder>/app/manifests/AndroidManifest.xml

중략...

```
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

중략...



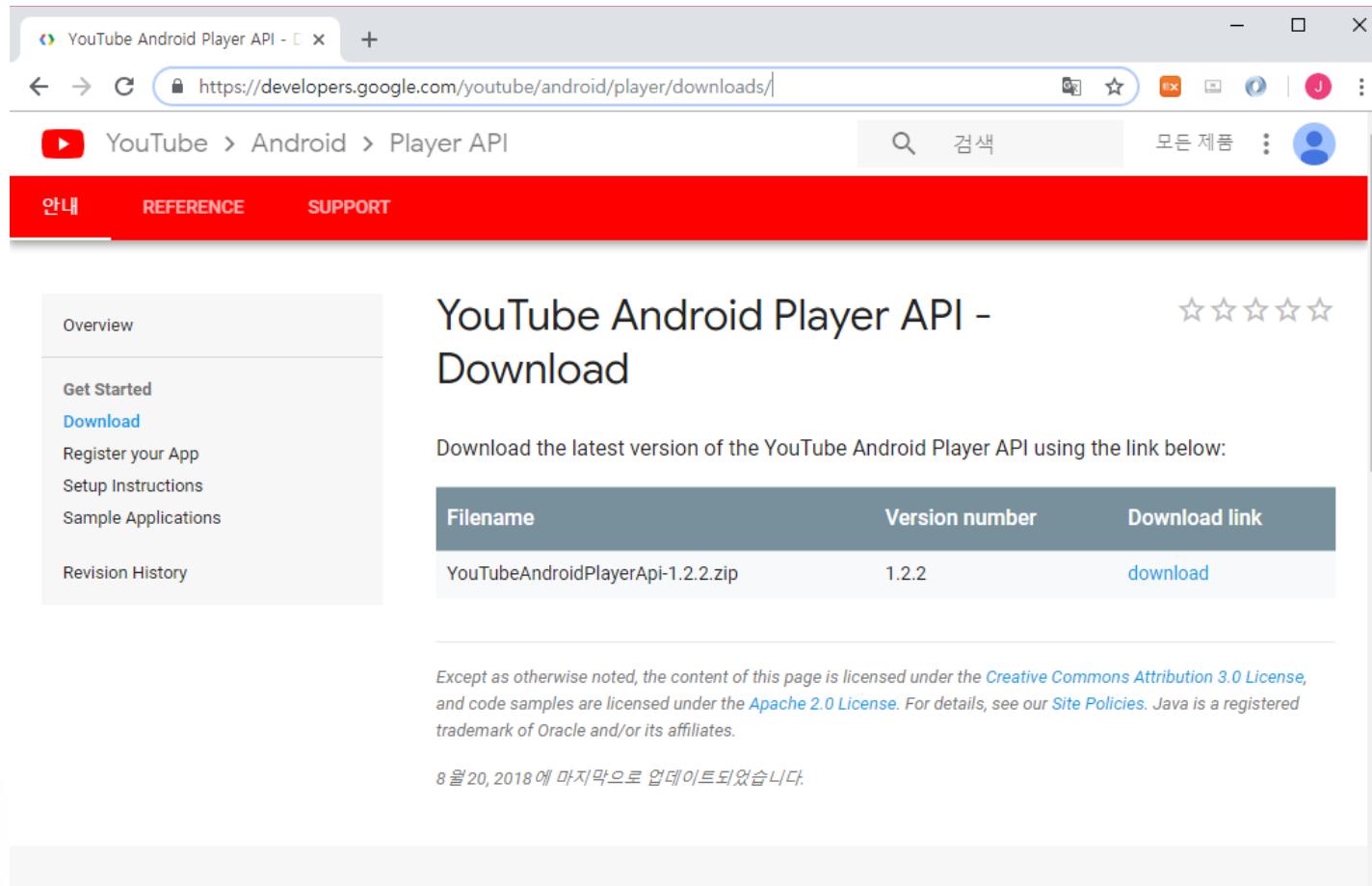
7.

YouTube 영상 재생하기



YouTube API 다운로드

- 다운로드 사이트
- <https://developers.google.com/youtube/android/player/downloads/>



The screenshot shows a web browser window with the URL <https://developers.google.com/youtube/android/player/downloads/>. The page title is "YouTube Android Player API - Download". The left sidebar has links for Overview, Get Started, Download (which is highlighted in blue), Register your App, Setup Instructions, Sample Applications, and Revision History. The main content area features a table with one row showing the file "YouTubeAndroidPlayerApi-1.2.2.zip" with version number 1.2.2 and a "download" link. A note at the bottom states that the content is licensed under Creative Commons Attribution 3.0 License and Apache 2.0 License.

Filename	Version number	Download link
YouTubeAndroidPlayerApi-1.2.2.zip	1.2.2	download

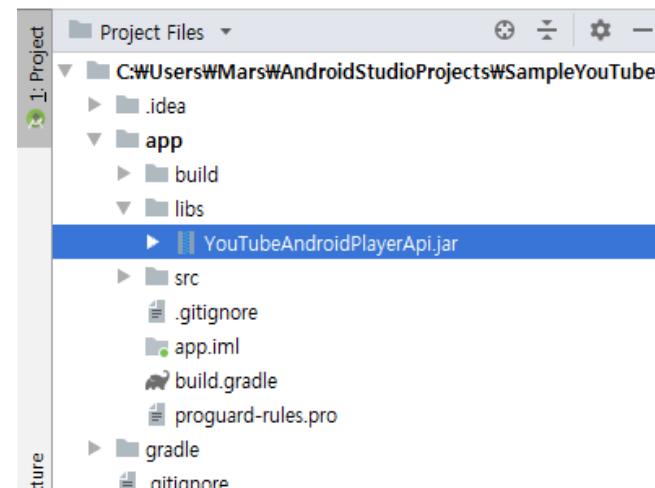
Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

8월 20, 2018에 마지막으로 업데이트되었습니다.



YouTube API 다운로드

- 다운로드한 파일을 /app/libs 폴더 안에 복사
- Sync... 메뉴 클릭하여 변경 사항 반영





화면 레이아웃 구성

- YouTubePlayerView 추가

```
<com.google.android.youtube.player.YouTubePlayerView → YouTubePlayerView 태그 추가하기  
    android:id="@+id/playerView"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent" />
```



메인 액티비티 코드 추가

- YouTubePlayerView 참조 및 초기화

참조파일 SampleYouTube>/app/java/org.techtown.youtube/MainActivity.java

```
public class MainActivity extends YouTubeBaseActivity {  
    YouTubePlayerView playerView;  
    YouTubePlayer player;  
  
    private static String API_KEY = "AIzaSyDWXmWuB7xjPaN5gwW2h1NtHr9nETESN7E";  
    private static String videoId = "A22oy8dFjqc";  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        initPlayer();  
  
        Button button = findViewById(R.id.button);  
        button.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                playVideo();  
            }  
        });  
    }  
}
```



메인 액티비티 코드 추가

- YouTubePlayerView 참조 및 초기화

```
playerView.initialize(API_KEY, new YouTubePlayer.OnInitializedListener() { → ❶ YouTube
    @Override
    public void onInitializationSuccess(YouTubePlayer.Provider provider,
                                       YouTubePlayer youTubePlayer, boolean b) {
        player = youTubePlayer;

        player.setPlayerStateChangeListener(new YouTubePlayer.PlayerStateChangeListener() {
            @Override
            public void onLoading() {}

            @Override
            public void onLoaded(String id) {
                Log.d("PlayerView", "onLoaded 호출됨: " + id);

                player.play(); → ❷ 동영상이 로딩되었으면 재생하기
            }
        });
    }
}
```



메인 액티비티 코드 추가

- playVideo 메서드에서 play

```
public void playVideo() {  
    if (player != null) {  
        if (player.isPlaying()) {  
            player.pause();  
        }  
  
        player.cueVideo(videoId);  
    }  
}
```



재생할 동영상 지정 및 권한 추가

- 재생할 동영상의 id 확인

Queen - Live at LIVE AID 1985/07/13 [Best Version]
조회수 129,113,057회

다음 동영상

- Queen Perform Live at LIVE AI...
24:37
- 관련 재생목록 - Queen - Live at LIV...
YouTube
- 보헤미안 랩소디 OST 전곡 |...
DY.CINE.MUSIC
조회수 204만회
- 에드 · 셔란 메드レー ▾The...
Pop Music
조회수 428만회
- 광고 없는 아이유 IU 노래 모음...



참조파일 SampleYouTube>/app/manifests/AndroidManifest.xml

중략...

```
<uses-permission android:name="android.permission.INTERNET"/>
```

중략...



GPS를 이용한 위치확인 및 지도표시 방법의 이해



목 차

- 1 GPS로 나의 위치 확인하기
- 2 현재 위치의 지도 보여주기
- 3 지도에 아이콘 추가하기
- 4 앱위젯 만들기



내가 있는 위치를 GPS로 확인하려면 어떻게 하나요?

- GPS를 이용해 나의 위치 확인하기
- 그래픽 그리기



내 위치를 지도 위에 보여주고 아이콘으로 표시하고 싶어요!

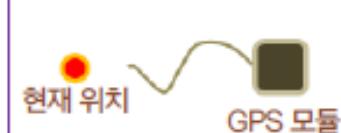
- 현재 위치의 지도 보여주기
- 지도에 아이콘 추가하기



홈 화면에 위젯을 띄워주고 싶어요!

- 앱 위젯 만들기

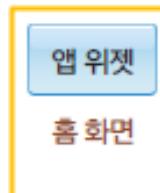
내 위치 확인하기



내 위치 지도에 표시하기



앱 위젯 만들기





1.

GPS로 나의 위치 확인하기



안드로이드의 주요 클래스

- **android.location 패키지**

- **수신 가능한 정보**

위치는 위도(latitude) 와 경도(longitude)로 표시되며 시간은 UTC timestamp로 표시되고 그 외에 선택적으로 고도(altitude), 속도(speed) 그리고 bearing 정보 표시

- **Location 객체**

위도와 경도 좌표는 각각 getLatitude()와 getLongitude() 메소드를 사용하며 그 값은 double 타입

- **LocationManager 객체**

시스템의 위치기반 서비스 접근 지원

애플리케이션에 제공하는 기능

- 주기적으로 바뀌는 단말의 위치정보 수신

- 단말이 특정 위치에 가까이 가면 애플리케이션의 특정 인텐트 실행

사용 방법

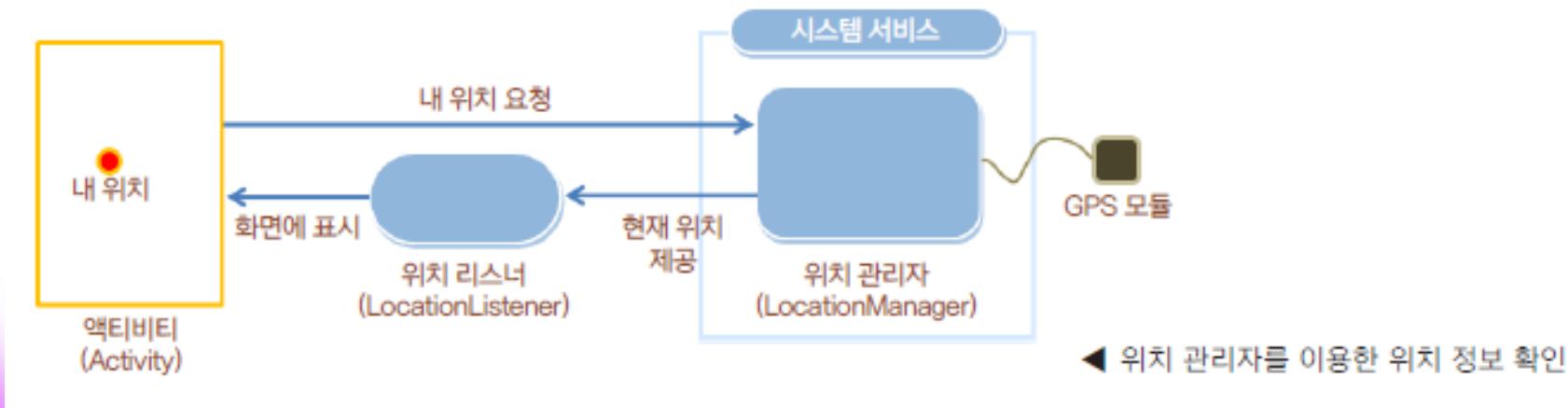
- 시스템 서비스로 제공되므로 아래의 메소드 사용하여 참조

Context.getSystemService(Context.LOCATION_SERVICE)



GPS를 이용해 나의 위치 확인하기

- 1단계 : 위치관리자 객체 참조하기
- 2단계 : 위치리스너 구현하기
- 3단계 : 위치정보 업데이트 요청하기
- 4단계 : 매니페스트에 권한 추가하기





위치관리자 객체 참조하기

(1) 1단계 - 위치관리자 객체 참조하기

```
public void startLocationService() {  
    LocationManager manager = (LocationManager) getSystemService(Context.LOCATION_SERVICE); ① LocationManager 객체 참조하기  
    try {  
        Location location = manager.getLastKnownLocation(LocationManager.GPS_PROVIDER); ② 이전에 확인했던 위치 정보 가져오기  
        if (location != null) {  
            double latitude = location.getLatitude();  
            double longitude = location.getLongitude();  
            String message = "최근 위치-> Latitude : " + latitude + "WnLongitude:" + longitude;  
  
            textView.setText(message);  
        }  
    }  
}
```



위치리스너 구현하기

(2) 2단계 - 위치리스너 구현하기

참조파일 SampleLocation>/app/java/org.techtown.location/MainActivity.java

중략…

```
class GPSListener implements LocationListener {  
    public void onLocationChanged(Location location) { —→ 위치가 확인되었을 때 자동으로 호출되는  
        Double latitude = location.getLatitude();           onLocationChanged() 메서드  
        Double longitude = location.getLongitude();  
        String message = "내 위치-> Latitude : "+ latitude + "WnLongitude:" + longitude;  
        textView.setText(message);  
    }  
  
    public void onProviderDisabled(String provider) {}  
  
    public void onProviderEnabled(String provider) {}  
  
    public void onStatusChanged(String provider, int status, Bundle extras) {}  
}
```



위치정보 업데이트 요청하기

(3) 3단계 - 위치정보 업데이트 요청하기

```
GPSListener gpsListener = new GPSListener(); → ① 리스너 객체 생성  
long minTime = 10000;  
float minDistance = 0;  
  
manager.requestLocationUpdates(LocationManager.GPS_PROVIDER, minTime, minDistance, gpsListener); → ② 위치 요청하기  
Toast.makeText(getApplicationContext(), "내 위치확인 요청함", Toast.LENGTH_SHORT).show();  
  
} catch(SecurityException e) {  
    e.printStackTrace();  
}  
}
```

중략...



매니페스트에 권한 추가하기

(4) 4단계 - 매니페이스에 권한 추가하기

위험 권한 부여 코드도 추가해야 함

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION">
</uses-permission>
```

참조파일 SampleLocation>/app/java/org.techtown.location/MainActivity.java

```
public class MainActivity extends AppCompatActivity implements AutoPermissionsListener {
    TextView textView;

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

        중략...

        AutoPermissions.Companion.loadAllPermissions(this, 101);
    }
}
```



실행 화면

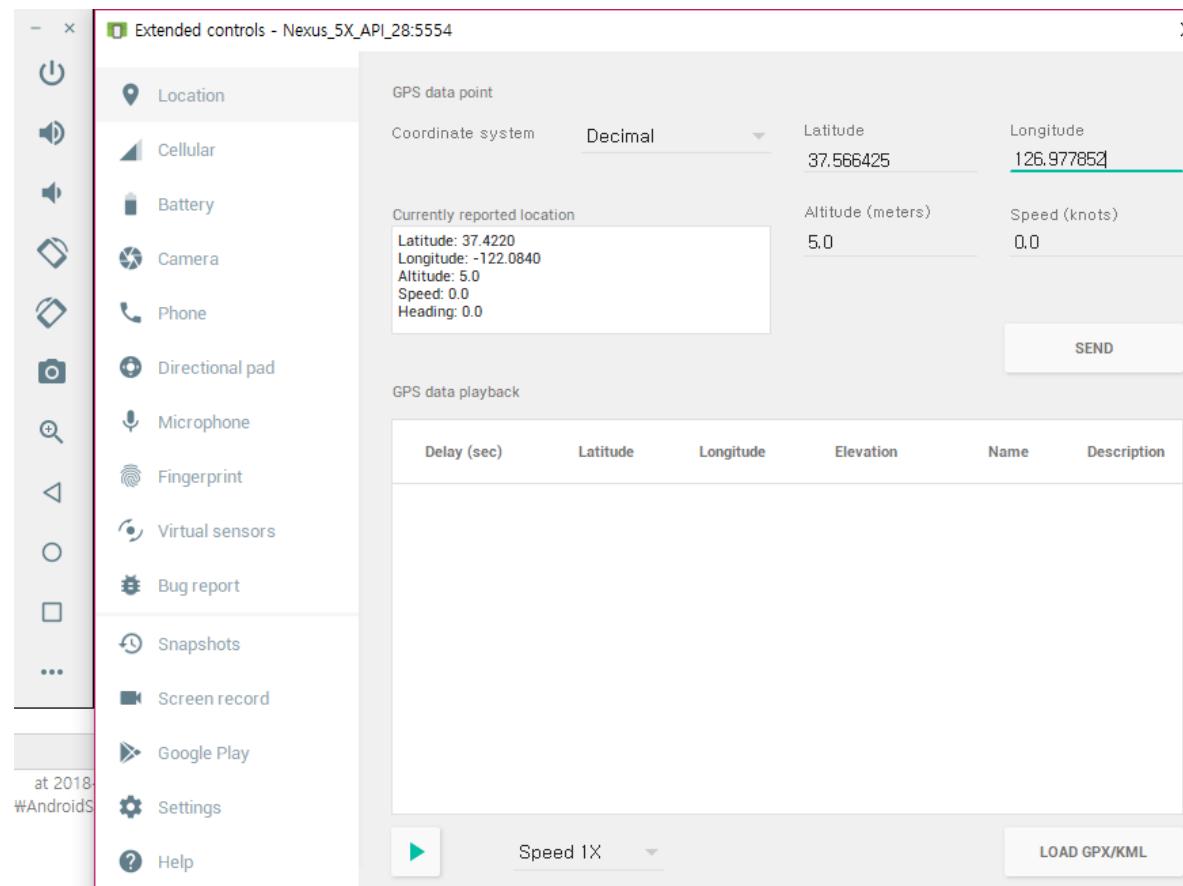


위치관리자 서비스를 이용한 위치 확인



가상 위치정보 전송하기

(5) 에뮬레이터로 가상 위치정보 전송하기



에뮬레이터로 가상 위치정보 전송



2.

현재 위치의 지도 보여주기



현재 위치의 지도 보여주기

- 구글맵 API v2에서는 Google Play Service 모듈을 사용
 - Google Play Service 모듈을 안드로이드 스튜디오의 Dependencies에 추가합니다.
- 맵 프래그먼트 (Map Fragment)
 - 지도를 보여주기 위해서는 프래그먼트를 사용합니다.
- 지도 API 키 발급받아 사용
 - 지도 API키는 구글 개발자 콘솔에서 발급받을 수 있습니다.
이 키는 매니페스트에 등록합니다.
- 권한 및 기타 정보 등록
 - 지도를 추가하게 되면 여러 가지 권한 (인터넷 접속 권한, GPS 위치정보 확인 권한) 과 기타 정보를 매니페스트에 등록해야 합니다.



Google Play Service 모듈 설치 여부 확인

(1) SDK 매니저에서 Google Play Service 모듈 설치했는지 확인

The screenshot shows the 'SDK Manager' window under 'Appearance & Behavior > System Settings > Android SDK'. The 'SDK Platforms' tab is selected. A table lists various Android developer tools and their status:

Name	Version	Status
Android SDK Build-Tools		Installed
GPU Debugging tools		Not Installed
LLDB		Not Installed
CMake		Not Installed
Android Auto API Simulators	1	Not installed
Android Auto Desktop Head Unit emulator	1.1	Not installed
Android Emulator	28.0.22	Installed
Android SDK Platform-Tools	28.0.1	Installed
Android SDK Tools	26.1.1	Installed
Documentation for Android SDK	1	Not installed
Google Play APK Expansion library	1	Not installed
Google Play Instant Development SDK	1.6.0	Not installed
Google Play Licensing Library	1	Not installed
Google Play services	49	Installed
Google USB Driver	11	Not installed
Google Web Driver	2	Not installed
Intel x86 Emulator Accelerator (HAXM installer)	7.3.2	Installed
NDK	19.0.5160253 rc2	Not installed
Support Repository		
ConstraintLayout for Android		Installed
Solver for ConstraintLayout		Installed
Android Support Repository	47.0.0	Installed
Google Repository	58	Installed

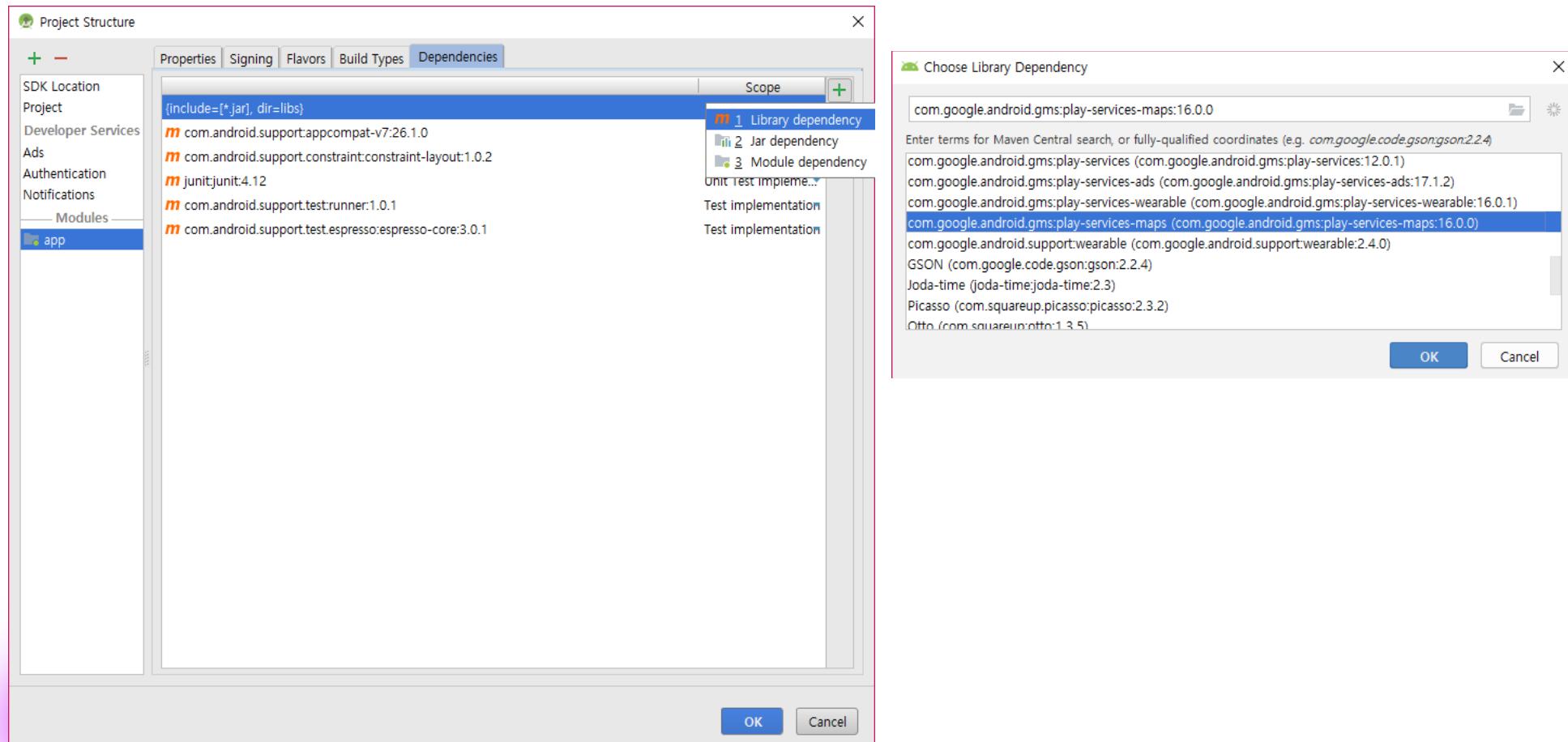
At the bottom, there are checkboxes for 'Hide Obsolete Packages' and 'Show Package Details', and buttons for 'OK', 'Cancel', 'Apply', and 'Help'.



Google Play Service 모듈 로딩

(2) 라이브러리 모듈 추가

● File > Project Structure





레이아웃에 맵 프래그먼트 추가하기

(3) 레이아웃에 맵 프래그먼트 추가

<fragment → 지도를 보여주기 위한 프래그먼트 태그 추가하기

```
    android:id="@+id/map"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    class="com.google.android.gms.maps.SupportMapFragment" />
```



액티비티 정의하기

(4) 액티비티 정의하기

참조파일 SampleLocationMap>/app/java/org.techtown.location/MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
  
    SupportMapFragment mapFragment;  
    GoogleMap map;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        mapFragment = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.map);  
        mapFragment.getMapAsync(new OnMapReadyCallback() {  
            @Override  
            public void onMapReady(GoogleMap googleMap) {  
                Log.d("Map", "지도 준비됨.");  
                map = googleMap;  
            }  
        });  
    }  
}
```

XML 레이아웃에 추가했던 프래그먼트 객체 참조하고 getMapAsync() 메서드 호출하기



액티비티 정의하기 (계속)

```
        }
    });

    try {
        MapsInitializer.initialize(this);
    } catch (Exception e) {
        e.printStackTrace();
    }

    Button button = findViewById(R.id.button);
    button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            startLocationService();
        }
    });
}
```



위치 확인 시 현재 위치 보여주기

참조파일 SampleLocationMap>/app/java/org.techtown.location/MainActivity.java

중략...

```
class GPSListener implements LocationListener {  
    public void onLocationChanged(Location location) {
```

중략...

```
        showCurrentLocation(latitude, longitude);  
    }  
    중략...
```

}

```
private void showCurrentLocation(Double latitude, Double longitude) {  
    LatLng curPoint = new LatLng(latitude, longitude); —→ ① 현재 위치의 좌표로 LatLng 객체 생성하기  
    map.animateCamera(CameraUpdateFactory.newLatLngZoom(curPoint, 15)); —→ ② 지정한 위치의  
}
```

지도 영역 보여주기

중략...



매니페스트에 정보 등록하기

(5) 매니페스트에 정보 등록하기

참조파일 SampleLocationMap>/app/manifests/AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="org.techtown.location"

    <permission android:name="org.techtown.location.permission.MAPS_RECEIVE"
        android:protectionLevel="signature" />

    <uses-permission android:name="org.techtown.location.permission.MAPS_RECEIVE" />

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />

    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />

    <uses-feature android:glEsVersion="0x00020000"
        android:required="true" />
```



매니페스트에 정보 등록하기

(5) 매니페스트에 정보 등록하기

```
<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"
    android:usesCleartextTraffic="true">

    <uses-library android:name="com.google.android.maps" />
    <uses-library
        android:name="org.apache.http.legacy"
        android:required="false" />

    <meta-data —→ API 키를 설정하는 부분
        android:name="com.google.android.maps.v2.API_KEY"
        android:value="AIzaSyAMMxTYNG0pesLG6tq1E55_DSnAFYTtw8bo" />

    <meta-data
        android:name="com.google.android.gms.version"
        android:value="@integer/google_play_services_version" />
```



그래들 설정

```
allprojects {  
    repositories {  
        maven { url 'https://jitpack.io' }  
    }  
}  
  
dependencies {  
    implementation fileTree(include: ['*.jar'], dir: 'libs')  
    implementation 'com.android.support:appcompat-v7:28.0.0'  
    implementation 'com.android.support:animated-vector-drawable:28.0.0'  
    implementation 'com.android.support:support-media-compat:28.0.0'  
    implementation 'com.android.support:support-v4:28.0.0'  
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'  
    testImplementation 'junit:junit:4.12'  
    androidTestImplementation 'com.android.support.test:runner:1.0.2'  
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'  
    implementation 'com.google.android.gms:play-services-maps:16.0.0'  
  
    implementation 'com.github.pedroSG94:AutoPermissions:1.0.3'  
}
```



구글맵 API 키 발급받기

(6) 구글 콘솔 사이트에서 구글맵 API 키 발급받기

▶ <http://console.developers.google.com>

The screenshot shows the Google Cloud Platform API Library interface. On the left, there's a sidebar with project management options like '새 프로젝트', '프로젝트 이름' (set to 'MyMap'), and '고급 옵션 보기...'. Below that are buttons for '만들기' and '취소'. The main content area displays the 'Maps SDK for Android' entry under the 'Google' category. It includes a circular icon with an Android logo, a brief description 'Maps for your native Android app.', and a '관리' button with a checkmark indicating 'API 사용 설정됨'. At the bottom, there are sections for '유형' (API 및 서비스), '개요' (description), and '최종 업데이트' (last update). The '개요' section contains the following text: 'Add maps based on Google Maps data to your Android application with the Maps SDK for Android. The SDK automatically handles access to Google Maps servers, map display and response to user gestures such as clicks and drags.' A callout box at the bottom right highlights the XML code for adding the API key to an Android manifest file:

```
<meta-data  
    android:name="com.google.android.maps.v2.API_KEY"  
    android:value="00000000000000000000000000000000" />
```



실행 화면



현재 위치가 맵 프래그먼트에서 지도로 보여지는 화면

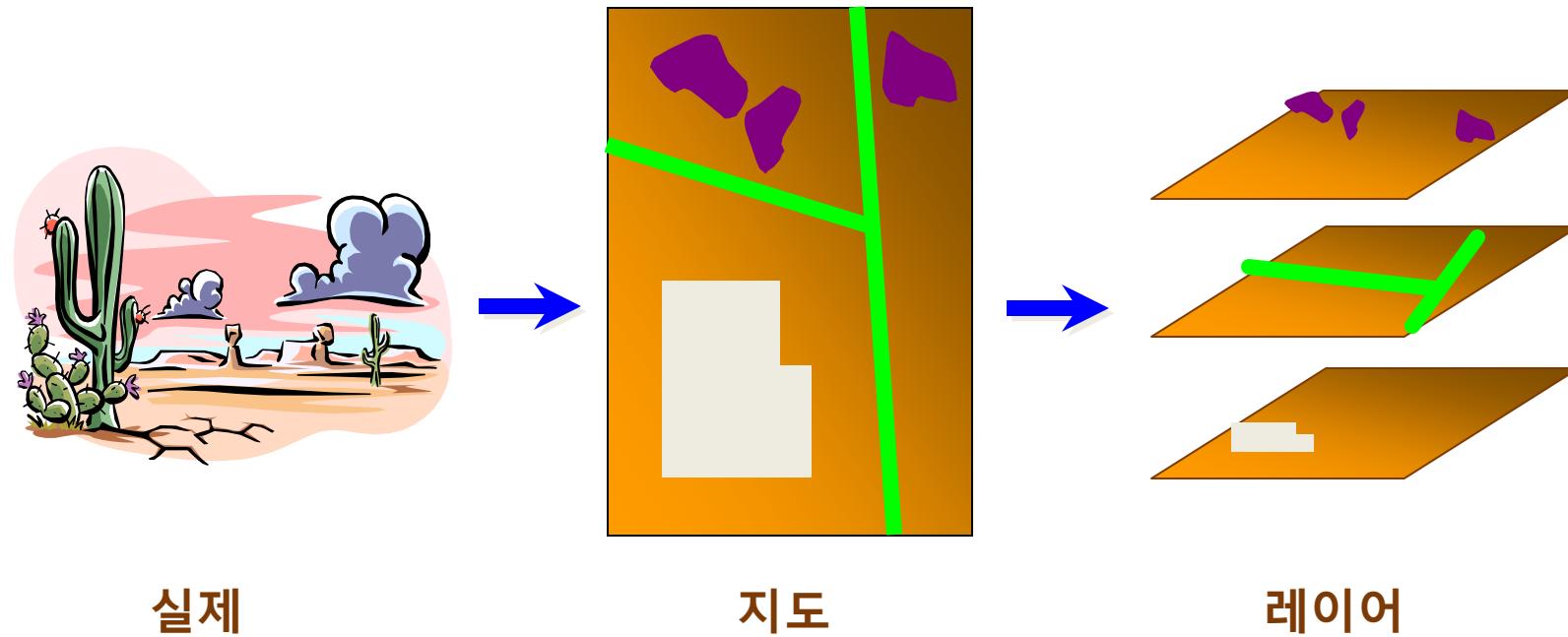


3.

지도에 아이콘 추가하기



오버레이란?



구글맵을 구성하는 레이어(오버레이)의 개념



마커를 이용해 위치를 이미지로 표시

참조파일 SampleLocationMapOverlay>/app/java/org.techtown.location/MainActivity.java

중략...

```
MarkerOptions myLocationMarker;
```

중략...

```
private void showCurrentLocation(Location location) {
```

중략...

```
    showMyLocationMarker(curPoint);
```

```
}
```

```
private void showMyLocationMarker(LatLng curPoint) {
```

```
    if (myLocationMarker == null) {
```

```
        myLocationMarker = new MarkerOptions(); —→ ① 마커 객체 생성하기
```

```
        myLocationMarker.position(curPoint);
```

```
        myLocationMarker.title("● 내 위치");
```

```
        myLocationMarker.snippet("● GPS로 확인한 위치");
```

```
        myLocationMarker.icon(BitmapDescriptorFactory.fromResource(R.drawable.mylocation));
```

```
        map.addMarker(myLocationMarker); —→ ② 지도에 마커 추가하기
```

```
    } else {
```

```
        myLocationMarker.position(curPoint);
```

```
}
```

```
}
```



실행 화면



현재 위치를 내 위치 표시 방식을 이용해 포인트 표시

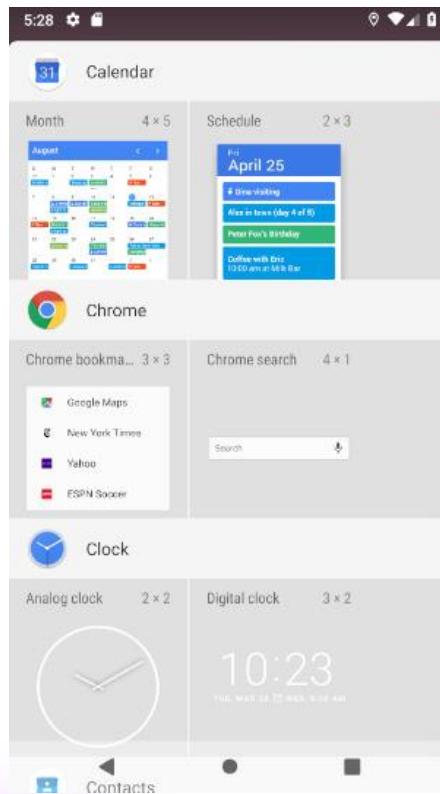


4.

앱위젯 만들기



앱위젯 만들기



앱 위젯 호스트(App Widget Host)	위젯을 담고 있는 그릇
앱 위젯 제공자(App Widget Provider)	위젯을 보여주는 제공자
위젯의 초기 뷰 레이아웃	앱 위젯이 처음에 화면에 나타날 때 필요한 레이아웃을 정의합니다. XML로 정의합니다.
앱 위젯 제공자 정보 (App Widget Provider Info) 객체	앱 위젯을 위한 메타데이터(레이아웃, 업데이트 주기 등)를 가지고 있습니다. 앱 위젯 제공자 클래스에 대한 정보를 가지고 있습니다. XML로 정의합니다.
앱 위젯 제공자 (App Widget Provider)	앱 위젯과 정보를 주고받기 위한 기본 클래스입니다. 브로드캐스트 수신자로 만들며 앱 위젯의 상태변화에 따른 기능을 구현합니다.



■ 앱위젯 호스트(App Widget Host)

- 위젯을 담고 있는 그릇

■ 앱위젯 제공자(App Widget Provider)

- 위젯을 보여주는 제공자



■ 앱위젯의 초기 뷰 레이아웃

- 앱위젯이 처음에 화면에 나타날 때 필요한 레이아웃을 XML로 정의

■ 앱위젯 제공자 정보(App Widget Provider Info) 객체

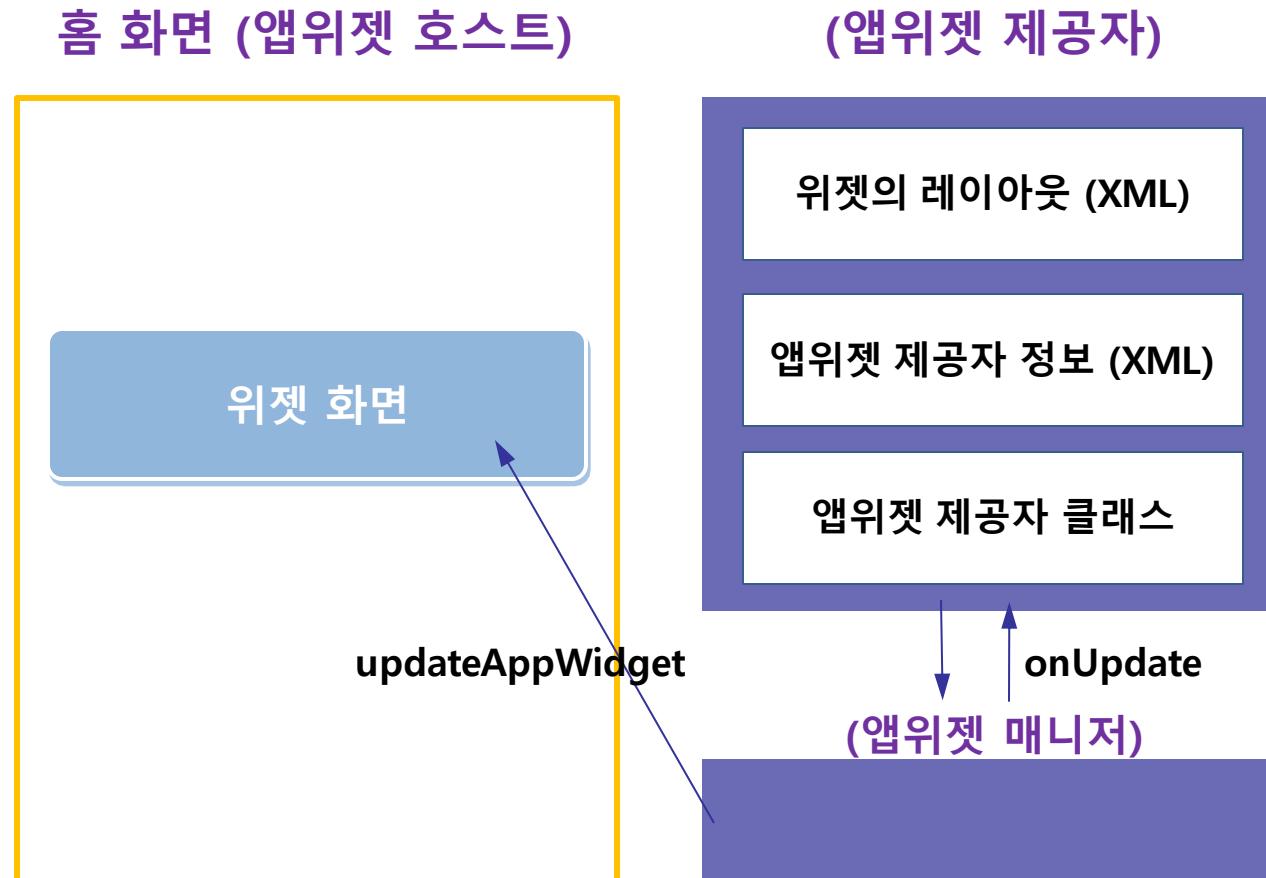
- 앱위젯을 위한 메타데이터(레이아웃, 업데이트 주기 등)를 XML로 정의

■ 앱위젯 제공자(App Widget Provider)

- 앱위젯과 정보를 주고받기 위한 기본 클래스



앱위젯의 구성





앱위젯 만들기 – 레이아웃 구성

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
    <LinearLayout  
        android:layout_width="match_parent"  
        android:layout_height="match_parent"  
        android:background="@drawable/background"  
        android:padding="10dp">  
        <TextView  
            android:layout_width="match_parent"  
            android:layout_height="match_parent"  
            android:id="@+id/txtInfo"  
            android:text="내 위치 정보 수신중 ..."  
            android:textColor="#FFFFFF"  
            android:gravity="center_horizontal|center_vertical"  
            android:lineSpacingExtra="4dp"/>  
    </LinearLayout>  
</LinearLayout>
```



앱위젯으로 보여줄 수 있는 뷰

■ 앱 위젯으로 보여줄 수 있는 뷰

유형	뷰 이름
뷰그룹	FrameLayout, LinearLayout, RelativeLayout
뷰	AnalogClock, Button, Chronometer ImageButton, ImageView, ProgressBar, TextView



앱위젯 제공자 정보 설정

```
<appwidget-provider xmlns:android= "http://schemas.android.com/apk/res/android"  
    android:minWidth= "294dp"  
  
    android:minHeight= "72dp"  
  
    android:updatePeriodMillis= "1800000"  
    android:initialLayout= "@layout/mylocation">  
</appwidget-provider>
```

1

앱위젯으로 표현될 뷰의 최소 폭

2

앱위젯으로 표현될 뷰의 최소 높이

3

위젯을 업데이트할 시간 간격

4

앱위젯으로 표현될 뷰의 레이아웃 리소스 지정



앱위젯 제공자 클래스 만들기

```
...  
public class MyLocationProvider extends AppWidgetProvider {  
...  
    public void onUpdate(Context context, AppWidgetManager appWidgetManager,  
        int[] appWidgetIds) {  
        super.onUpdate(context, appWidgetManager, appWidgetIds);  
        Log.d("MyLocationProvider", "onUpdate() called : " + ycoord + ", " + xcoord);  
        final int size = appWidgetIds.length;  
        for (int i = 0; i < size; i++) {  
            int appWidgetId = appWidgetIds[i];  
            String uri = "geo:" + ycoord + "," + xcoord + "?z=10";  
            Intent intent = new Intent(android.content.Intent.ACTION_VIEW, Uri.parse(uri));  
            PendingIntent pendingIntent = PendingIntent.getActivity(context, 0, intent, 0);  
            RemoteViews views = new RemoteViews(context.getPackageName(),  
                R.layout.mylocation);  
            views.setOnClickPendingIntent(R.id.txtInfo, pendingIntent);  
            appWidgetManager.updateAppWidget(appWidgetId, views);  
        }  
        context.startService(new Intent(context, GPSLocationService.class));  
    }  
...  
}
```

1 앱위젯으로 표현될 뷰의 최소 폭

2 onUpdate() 메소드 다시 정의

3 지도를 띄우기 위한 URI 문자열 생성

4 지도를 띄우기 위한 인텐트 객체 생성

5 지도를 띄우기 위한 펜딩인텐트 객체 생성

6 뷰를 눌렀을 때 실행할 펜딩인텐트 객체 지정

7 앱위젯 매니저 객체의 updateAppWidget() 메소드 호출

8 GPS 위치 확인을 위한 서비스 시작



앱위젯 제공자 클래스 만들기 (계속)

```
public class MyLocationProvider extends AppWidgetProvider {  
    ...  
    public static class GPSLocationService extends Service {  
        public static final String TAG = "GPSLocationService";  
        private LocationManager manager = null;  
        private LocationListener listener = new LocationListener() {  
            ...  
            public void onLocationChanged(Location location) {  
                Log.d(TAG, "onLocationChanged() called.");  
                updateCoordinates(location.getLatitude(), location.getLongitude());  
                stopSelf();  
            }  
        };  
  
        public void onCreate() {  
            super.onCreate();  
            Log.d(TAG, "onCreate() called.");  
            manager = (LocationManager) getSystemService(LOCATION_SERVICE);  
        }  
    }  
}
```

1

위치 정보가 확인되면
updateCoordinates() 메소드 호출

2

서비스가 생성될 때 위치 관리자 객체 참조

Continued..



앱위젯 제공자 클래스 만들기 (계속)

```
startListening();  
}  
...
```

```
private void startListening() {  
    final Criteria criteria = new Criteria();  
    criteria.setAccuracy(Criteria.ACCURACY_COARSE);  
    criteria.setAltitudeRequired(false);  
    criteria.setBearingRequired(false);  
    criteria.setCostAllowed(true);  
    criteria.setPowerRequirement(Criteria.POWER_LOW);  
    final String bestProvider = manager.getBestProvider(criteria, true);  
    if (bestProvider != null && bestProvider.length() > 0) {  
        manager.requestLocationUpdates(bestProvider, 500, 10, listener);  
    } else {  
        final List<String> providers = manager.getProviders(true);  
        for (final String provider : providers) {  
            manager.requestLocationUpdates(provider, 500, 10, listener);  
        }  
    }  
}
```

3

서비스가 시작할 때 startListening() 메소드 호출

4

위치 관리자에 위치 정보 요청



앱위젯 제공자 클래스 만들기 (계속)

```
...
public class MyLocationProvider extends AppWidgetProvider {
    ...
    public static class GPSLocationService extends Service {
        ...
        private void updateCoordinates(double latitude, double longitude) {
            ...
            if (info.length() <= 0) {
                info = "[내 위치] " + latitude + ", " + longitude + "\n터치하면 지도로 볼 수 있습니다.";
            } else {
                info += ("\n" + "[내 위치] " + latitude + ", " + longitude + ")");
                info += "\n터치하면 지도로 볼 수 있습니다.";
            }
            RemoteViews views = new RemoteViews(getApplicationContext(), R.layout.mylocation);
            views.setTextViewText(R.id.txtInfo, info);
            ComponentName thisWidget = new ComponentName(this,
                MyLocationProvider.class);
            AppWidgetManager manager = AppWidgetManager.getInstance(this);
            manager.updateAppWidget(thisWidget, views);
            xcoord = longitude;
            ycoord = latitude;
            Log.d(TAG, "coordinates : " + latitude + ", " + longitude);
        ...
    }
}
```

1

위치좌표와 주소 정보를 포함하는 문자열 생성

2

앱위젯 업데이트



매니페스트에 태그 추가

```
...  
<application android:icon="@drawable/icon" android:label="@string/app_name">  
    ...  
    <receiver android:name=".MyLocationProvider">  
        <intent-filter>  
            <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />  
        </intent-filter>  
        <meta-data android:name="android.appwidget.provider"  
                  android:resource="@xml/mylocationinfo" />  
    </receiver>  
    <service android:name=".MyLocationProvider$GPSLocationService" />  
</application>  
...
```

1 <receiver> 태그에 새로 정의한 앱위젯 제공자 클래스 지정

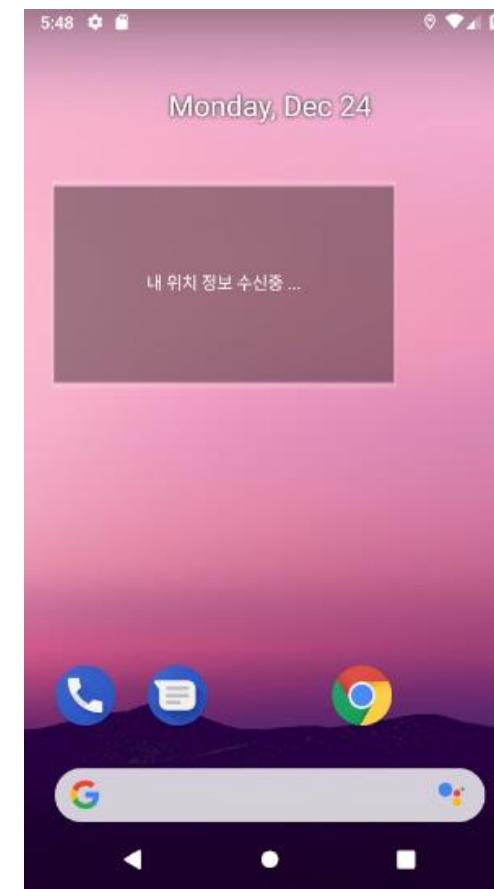
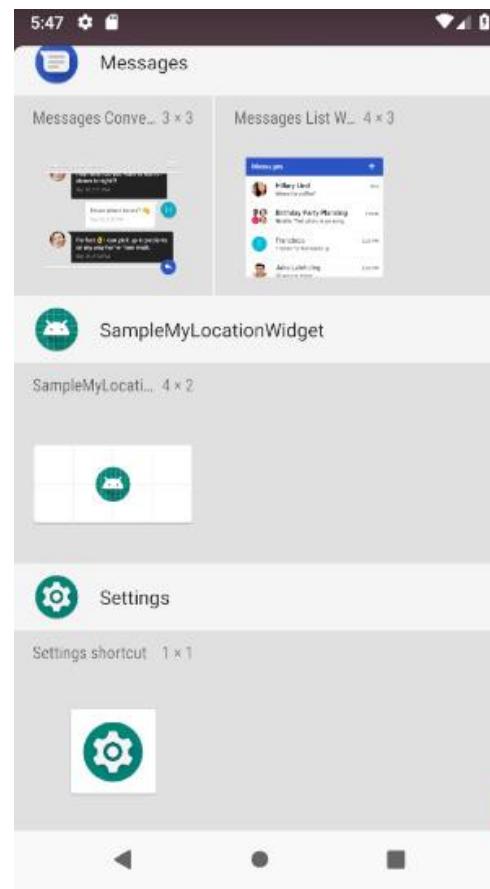
2 인텐트 필터의 액션 정보로 APPWIDGET_UPDATE 지정

3 <meta-data> 태그로 앱위젯 제공자 정보를 담고 있는 XML 리소스 지정

4 내 위치 확인을 위해 정의한 서비스 태그 정의



실행 화면



앱위젯을 홈 화면에 추가하고 위치 확인하기



푸시 서비스와 센서 그리고 단말 기능에 대한 이해



- 1 진동과 소리로 알려주기
- 2 상단 알림으로 알려주기
- 3 푸시 서비스 사용하기
- 4 센서 이해하기
- 5 시스템 서비스 활용하기
- 6 네트워크 기능 활용하기
- 7 다중 창 지원하기



어떤 정보를 알려주는 방법에는 어떤 것이 있나요?

- 진동과 소리로 알려주기
- 상단 알림으로 알려주기



푸시로 메시지를 보내주고 싶어요

- 푸시 서비스 사용하기



센서를 사용하고 싶어요

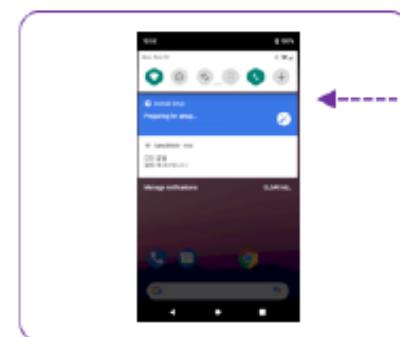
- 센서 사용하기



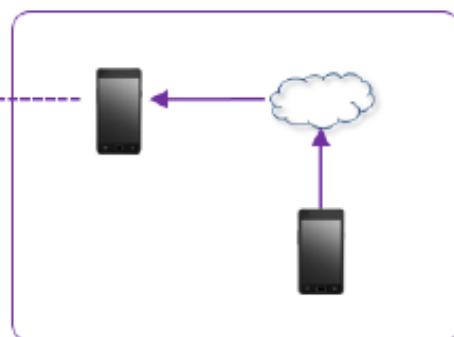
유용한 단말 기능을 알려주세요

- 네트워크 기능 활용하기
- 다중 창 지원하기

상단 알림으로 알려주기



푸시 서비스 사용하기





1.

진동과 소리로 알려주기



진동과 소리

- 사용자에게 무언가를 알려주는 가장 간단한 방법
- **vibrate**와 **play**

[API]

```
public void vibrate (long millisconds)  
public void vibrate (VibrationEffect vibe)
```

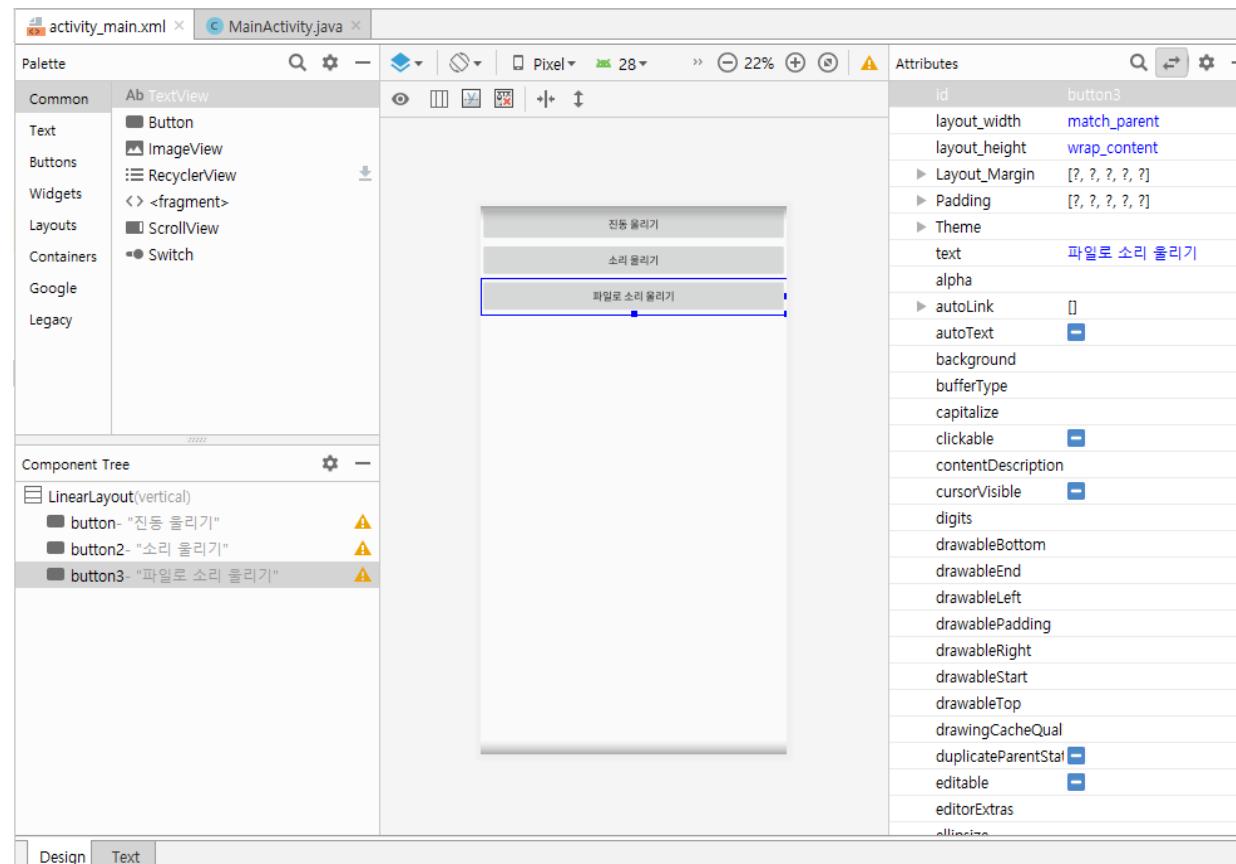
[API]

```
public void play ()
```



화면 레이아웃 만들기

- 상단에 버튼 추가





메인 액티비티 기능 추가

- 버튼 클릭 시 **Vibrator** 사용 코드 추가

참조파일 SampleVibrate>/app/java/org.techtown.vibrate/MainActivity.java

```
public class MainActivity extends AppCompatActivity {

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

        Button button = findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Vibrator vibrator = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE); →
                if (Build.VERSION.SDK_INT >= 26) {           ① 진동을 위해 Vibrator 객체 참조하기
                    vibrator.vibrate(VibrationEffect.createOneShot(1000,10));
                } else {
                    vibrator.vibrate(1000);
                }
            }
        });
    }
}
```

① 진동을 위해 Vibrator 객체 참조하기



메인 액티비티 기능 추가

- 버튼 클릭 시 소리 울리는 코드 추가

```
Button button2 = findViewById(R.id.button2);
button2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Uri uri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
        Ringtone ringtone = RingtoneManager.getRingtone(getApplicationContext(), uri);
        ringtone.play();                                ② 소리를 울리기 위해 Ringtone 객체 참조하기 ↗
    }
});
```



```
Button button3 = findViewById(R.id.button3);
button3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        MediaPlayer player = MediaPlayer.create(getApplicationContext(), R.raw.beep);
        player.start();
    }
});
```



매니페스트에 권한 추가 후 실행

- VIBRATE 권한 추가

참조파일 SampleVibrate>/app/manifests/AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="org.techtown.vibrate">

    <uses-permission android:name="android.permission.VIBRATE"/>
```

중략...





2.

상단 알림으로 알려주기

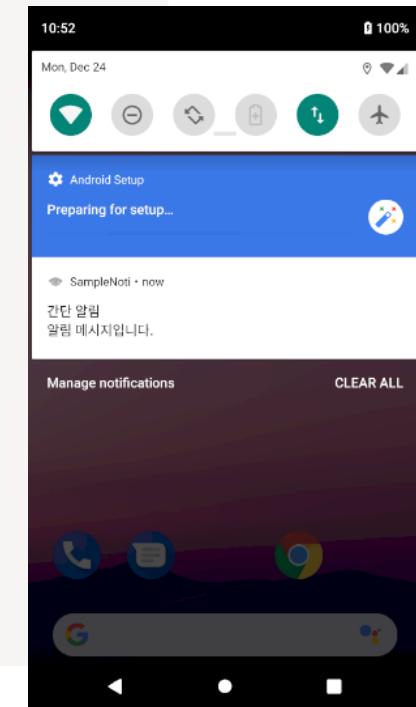


NotificationManager 사용

- 메인 액티비티에서 NotificationManager를 사용하는 코드 추가

```
public void showNoti() {  
    manager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);  
    NotificationCompat.Builder builder = null;  
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {  
        if (manager.getNotificationChannel(CHANNEL_ID) != null) {  
            manager.createNotificationChannel(new NotificationChannel(  
                CHANNEL_ID, CHANNEL_NAME, NotificationManager.IMPORTANCE_DEFAULT  
            ));  
  
            builder = new NotificationCompat.Builder(this, CHANNEL_ID);  
        }  
    } else {  
        builder = new NotificationCompat.Builder(this);  
    }  
    builder.setContentTitle("간단 알림");  
    builder.setContentText("알림 메시지입니다.");  
    builder.setSmallIcon(android.R.drawable.ic_menu_view);  
    Notification noti = builder.build();  
  
    manager.notify(1, noti); → ❷ 상단 알림 띄우기  
}
```

❶ NotificationManager
객체 참조하기





NotificationManager 사용

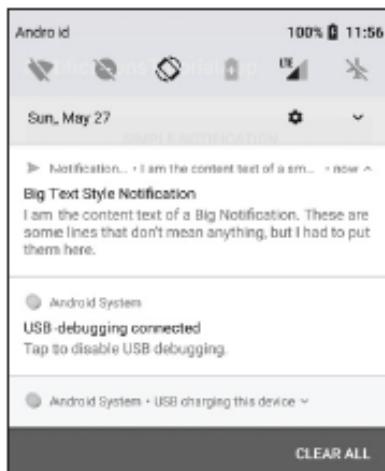
- 두 번째 버튼을 클릭했을 때 코드 추가

```
public void showNoti2() {  
    manager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);  
  
    NotificationCompat.Builder builder = null;  
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {  
        if (manager.getNotificationChannel(CHANNEL_ID2) == null) {  
            manager.createNotificationChannel(new NotificationChannel(  
                CHANNEL_ID2, CHANNEL_NAME2, NotificationManager.IMPORTANCE_DEFAULT  
            ));  
  
            builder = new NotificationCompat.Builder(this, CHANNEL_ID2);  
        }  
    } else {  
        builder = new NotificationCompat.Builder(this);  
    }  
  
    Intent intent = new Intent(this, MainActivity.class);  
    PendingIntent pendingIntent = PendingIntent.getActivity(this, 101, intent,  
        PendingIntent.FLAG_UPDATE_CURRENT);  
    builder.setContentTitle("간단 알림");  
    builder.setContentText("알림 메시지입니다.");  
    builder.setSmallIcon(android.R.drawable.ic_menu_view);  
    builder.setAutoCancel(true);  
    builder.setContentIntent(pendingIntent); —→ ❷ 빌더에 PendingIntent 객체 설정하기  
  
    Notification noti = builder.build();  
  
    manager.notify(2, noti);
```

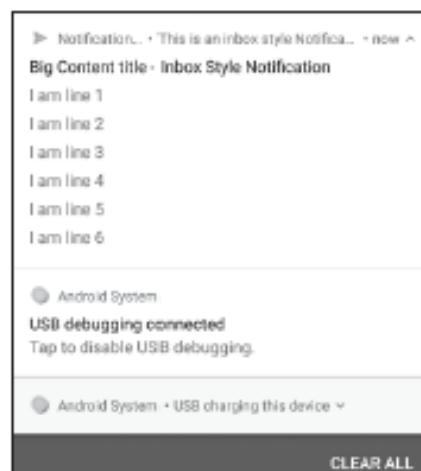
❶ PendingIntent
객체 만들기



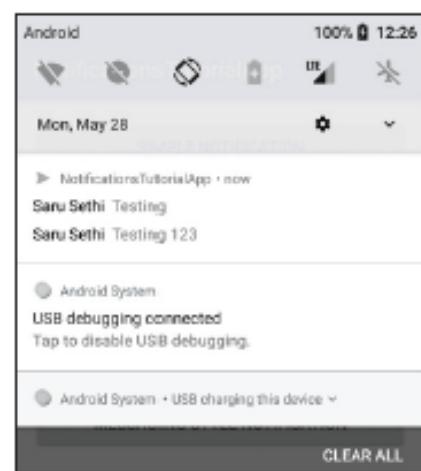
스타일 별로 다른 알림을 표시



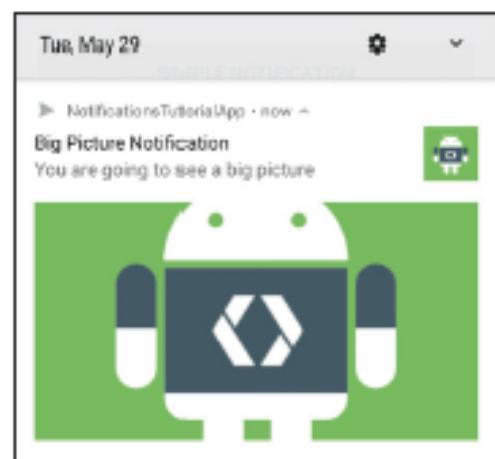
▲ BigTextStyle로 만든 알림



▲ InboxStyle로 만든 알림



▲ MessagingStyle로 만든 알림



▲ BigPictureStyle로 만든 알림



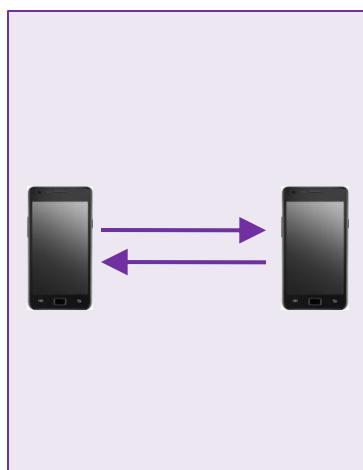
3.

푸시 서비스 사용하기



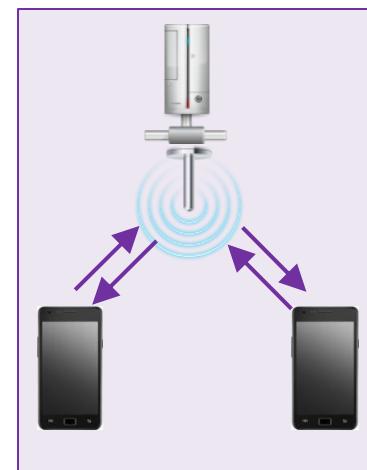
메시징 서비스의 구분

1. 단말 간 메시지 송수신



SMS / MMS

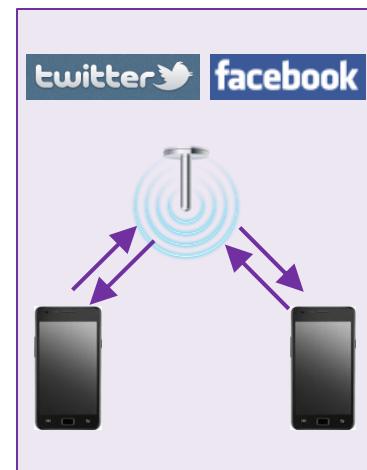
2. 서버를 통한 단말 간 메시지 송수신



XMPP

SIP

3. SNS를 통한 단말 간 메시지 공유



Twitter

Facebook

4. 푸시 서비스를 통한 메시지 전송



FCM



푸시 서비스 사용하기

● 단순 SMS를 이용한 알림

- 간단하지만 비용 발생

● 연결을 만들어 놓은 상태에서 알림

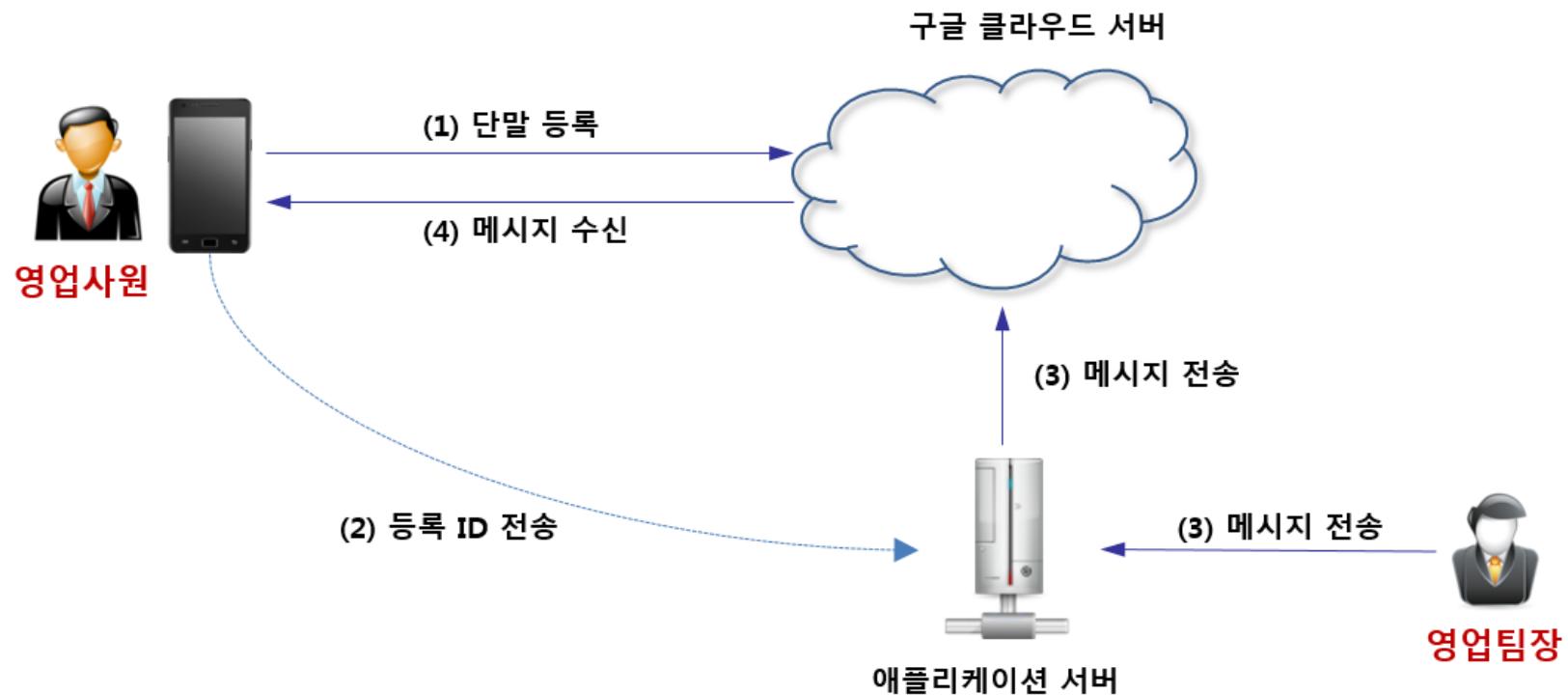
- 연결을 만들어 놓고 폴링하는 과정이 필요하며, 백그라운드 서비스를 이용해야 하므로 간단하지만은 않고 단말 리소스를 많이 사용함

● 구글의 푸시 서비스(FCM)를 사용하여 알림

- 연결을 만들어 놓고 폴링을 일부 사용하지만 구글의 클라우드 서버를 이용해 최적화한 서비스로 단말의 Play스토어에서 제공하는 기능이므로 효율적인 푸시 메시지 전송 가능



푸시 서비스 처리 과정



FCM 푸시 메시지 처리 과정



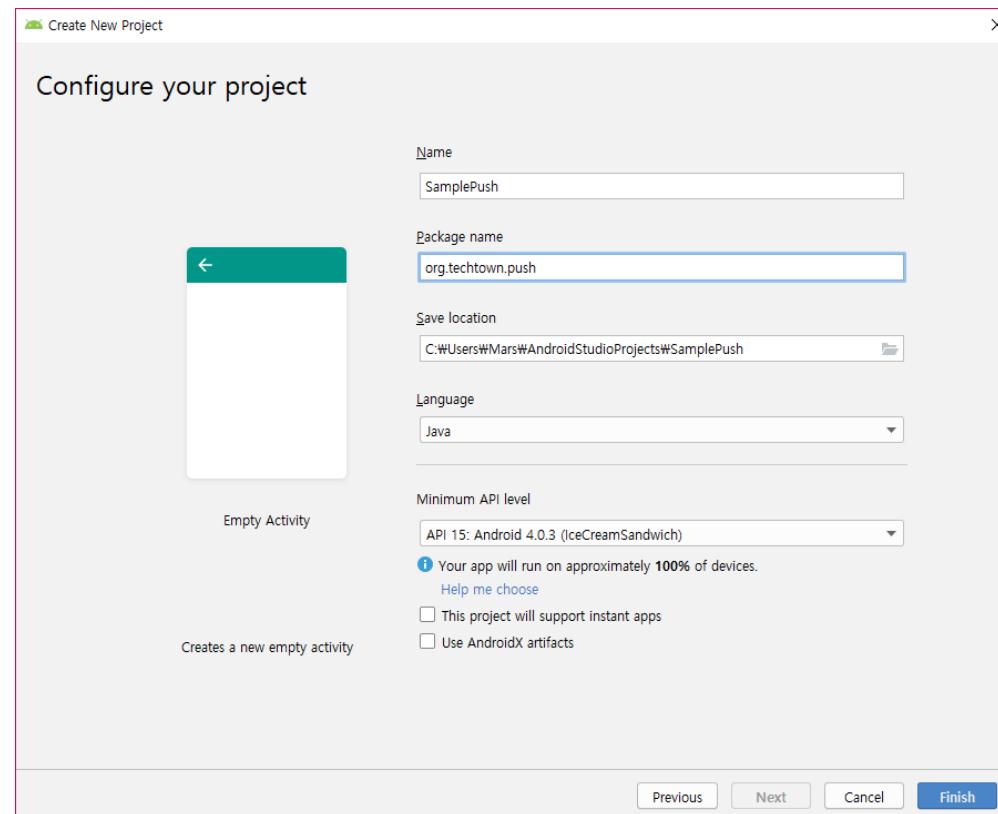
푸시 서비스 처리에 필요한 주요 정보

- (1) 애플리케이션 서버에 저장된 단말의 등록 ID
- (2) 애플리케이션 서버에서 클라우드 서버로 접속하기 위한 인증 정보



새로운 프로젝트 생성

- SamplePush라는 이름의 새로운 프로젝트 생성

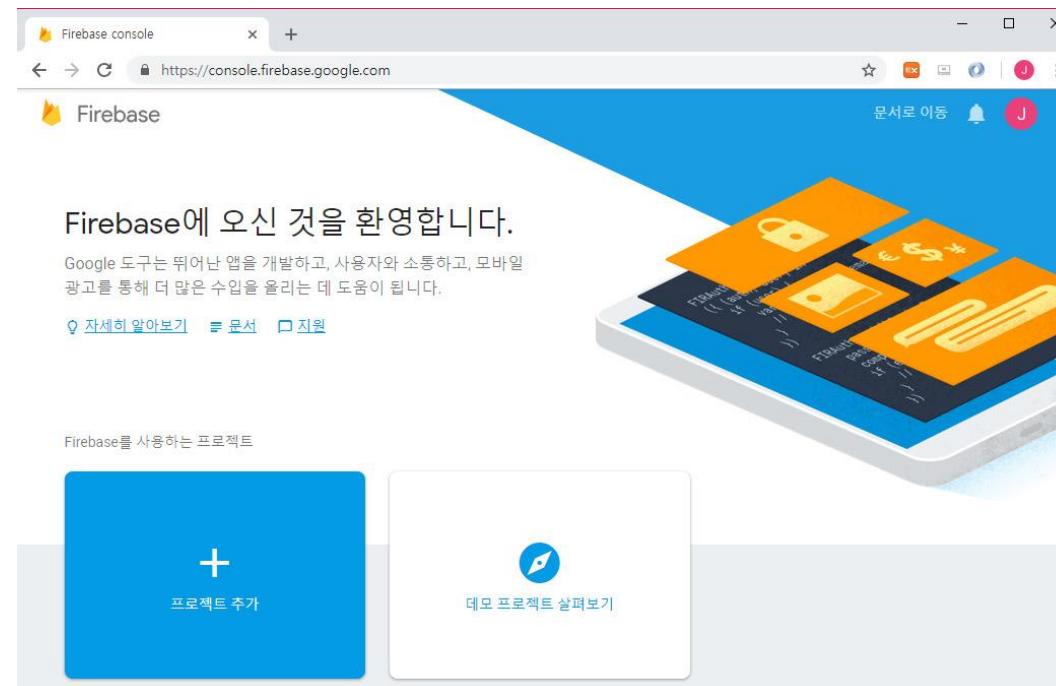




FCM 사이트에서 프로젝트 등록

- FCM 개발자 콘솔 페이지 접속하여 프로젝트 등록

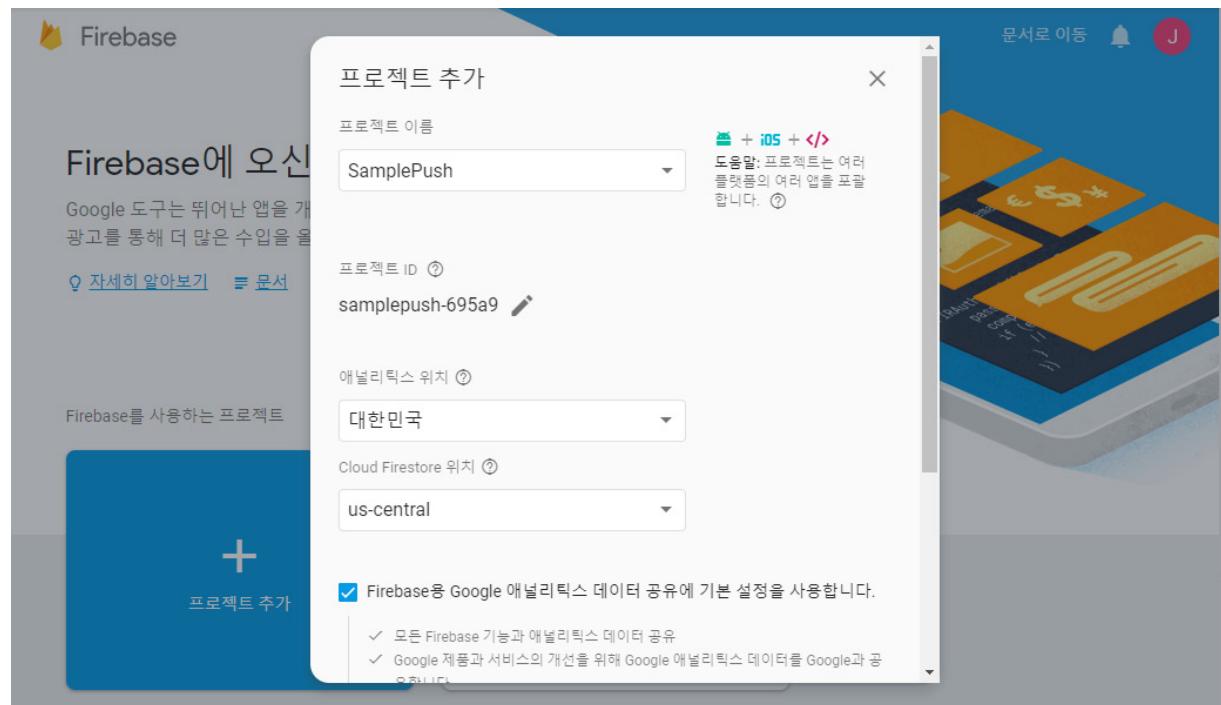
Firebase 개발자 콘솔(Console) 페이지 ► <https://console.firebaseio.google.com>





FCM 사이트에서 프로젝트 등록

- 프로젝트를 만든 후 [프로젝트 추가] 아이콘 클릭





FCM 사이트에서 프로젝트 등록

- 첫번째 단계에서 앱의 패키지 이름 입력
- 두번째 단계에서 google-services.json 파일 자동 다운로드 : 프로젝트의 [app] 폴더 안에 저장

SamplePush > app 폴더

SamplePush - 개요 – Firebase

https://console.firebaseio.google.com/project/samplepush-695a9/overview

Android 앱에 Firebase 추가

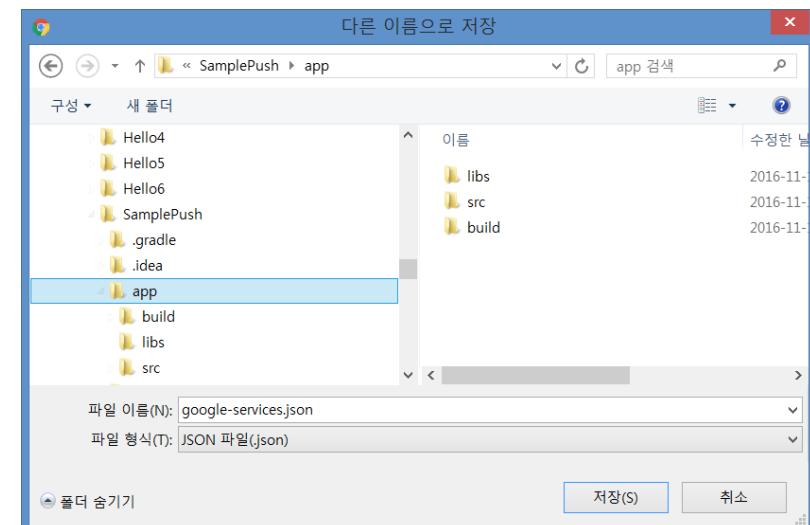
1 앱 등록

Android 패키지 이름 (②)
org.techtown.push

앱 네임(선택사항) (②)
Freemium Android App

디버그 서명 인증서 SHA-1(선택사항) (②)
00:00
인증에서 동적 링크, 조대, Google 로그인, 전화번호를 지원하는 데 필요합니다. 설정에서 SHA-1을 수정하세요.

App 등록





앱 설정 수정

- build.gradle(Project:SamplePush) 파일 수정

classpath 'com.google.gms:google-services:3.3.0'

The screenshot shows the Android Studio interface with the project 'SamplePush' open. The left sidebar displays the project structure with 'app' as the selected module. The main editor window shows the top-level build.gradle file. A yellow bar at the top of the code editor says 'Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.' with a 'Sync Now' button. The code in the editor is:

```
// Top-level build file where you can add configuration options common to all sub-projects under this build.  
//  
buildscript {  
    repositories {  
        google()  
        jcenter()  
    }  
    dependencies {  
        classpath 'com.android.tools.build:gradle:3.3.0'  
        // NOTE: Do not place your application dependencies here; they belong  
        // in the individual module build.gradle files  
        classpath 'com.google.gms:google-services:4.2.0'  
    }  
}  
  
allprojects {  
    repositories {  
        google()  
        jcenter()  
    }  
}
```

The line 'classpath 'com.google.gms:google-services:4.2.0'' is highlighted with a yellow background. The bottom status bar shows 'Gradle build finished in 6 s 809 ms (10 minutes ago)'.



앱 설정 수정

• build.gradle(Module:app) 파일 수정

The screenshot shows the Android Studio interface with the project 'SamplePush' open. The code editor displays the `build.gradle(Module:app)` file. The file contains Gradle configuration code, including dependencies for various Android Support libraries and Google Firebase messaging. The code is color-coded, with green for Java code and blue for Gradle keywords.

```
16     proguardFiles getDefaultProguardFile('proguard-android-optimize.txt')
17 }
18 }
19 }
20
21 dependencies {
22     implementation fileTree(dir: 'libs', include: ['*.jar'])
23     implementation 'com.android.support:appcompat-v7:28.0.0'
24     implementation 'com.android.support:animated-vector-drawable:28.0.0'
25     implementation 'com.android.support:support-media-compat:28.0.0'
26     implementation 'com.android.support:support-v4:28.0.0'
27     implementation 'com.android.support.constraint:constraint-layout:1.1.3'
28     testImplementation 'junit:junit:4.12'
29     androidTestImplementation 'com.android.support.test:runner:1.0.2'
30     androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
31
32     implementation 'com.google.firebase:firebase-messaging:17.3.4'
33 }
34
35 apply plugin: 'com.google.gms.google-services'
36 }
```



에러 발생 시

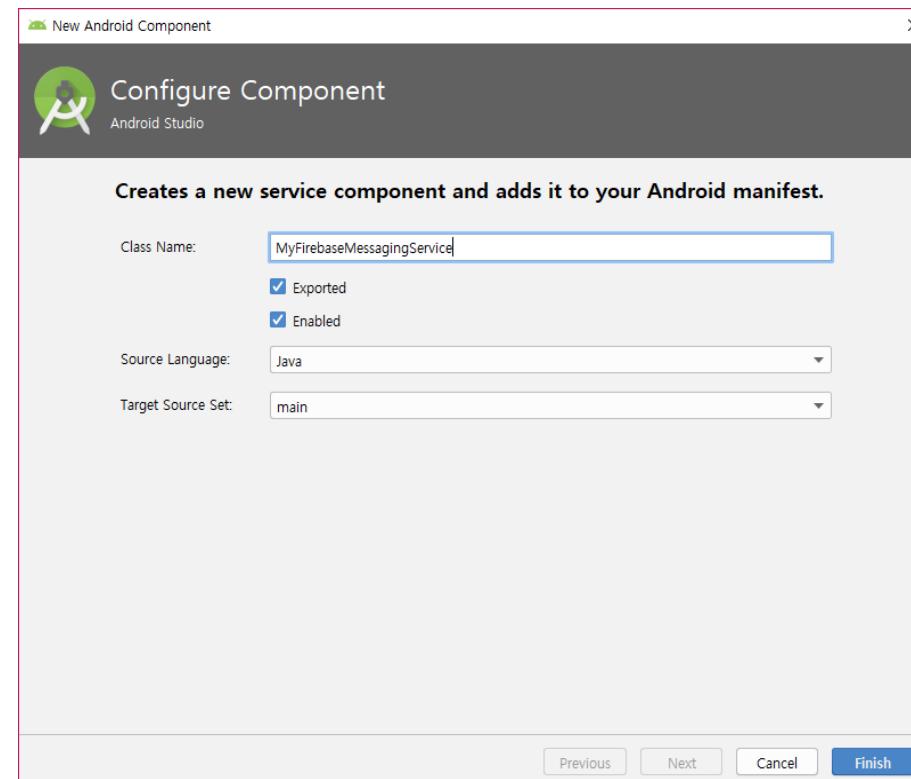
- google-services.json 파일 안에 Key 정보가 누락된 경우 다시 다운로드해야 함

The screenshot shows two windows side-by-side. On the left is the 'Messages Gradle Build' window from Android Studio, which displays an error message: 'Execution failed for task ':app:processDebugGoogleServices'. > Missing api_key/current_key object'. Below this window is a status bar with tabs for 'Messages', 'Terminal', 'Android Monitor', and 'TODO'. A message at the bottom says 'Gradle build finished with 1 error(s) in 1s 773ms (3 minutes ago)'. On the right is a browser window showing the Firebase console at <https://console.firebaseio.google.com/project/samplepush-695a9/settings/general/android:org.techtown.push>. The Firebase dashboard is visible on the left, and the 'SamplePush - 설정 - Firebase' tab is selected in the browser. In the main content area, it shows the 'Android 앱' section for the app 'org.techtown.push'. It displays the '최신 구성 파일 다운로드' button, which has a download icon and the file name 'google-services.json'. Below this, it shows the '앱 ID' (1:554003969263:android:10c691ce32fc7e03), '앱 닉네임' (닉네임 추가), '패키지 이름' (org.techtown.push), 'SHA 인증서 지문' (유형), and a '디지털 지문 추가' button. At the bottom right of the Firebase window, there is a '선택한 앱 삭제' button.



앱에서 서비스 추가하기

- 두 개의 서비스를 추가해야 함
- 등록ID를 위한 MyFirebaseInstanceIdService 서비스 추가





앱에서 서비스 추가하기

- **onNewToken** 메소드 재정의

참조파일 SamplePush>/app/java/org.techtown.push/MyFirebaseMessagingService.java

```
public class MyFirebaseMessagingService extends FirebaseMessagingService {
    private static final String TAG = "FMS";

    public MyFirebaseMessagingService() {

    }

    @Override
    public void onNewToken(String token) { → ① 새로운 토큰을 확인했을 때 호출되는 메서드
        super.onNewToken(token);
        Log.e(TAG, "onNewToken 호출됨: " + token);
    }

    @Override
    public void onMessageReceived(RemoteMessage remoteMessage) { → ② 새로운 메시지를 받았을 때
        Log.d(TAG, "onMessageReceived() 호출됨.");
    }
}
```



매니페스트에 서비스 등록

- 새로 생성한 서비스 등록

참조파일 SamplePush>/app/manifests/AndroidManifest.xml

중략...

```
<service  
    android:name=".MyFirebaseMessagingService"  
    android:enabled="true"    android:exported="true"  
    android:stopWithTask="false" >  
    <intent-filter>  
        <action android:name="com.google.firebase.MESSAGING_EVENT" />  
    </intent-filter>  
</service>
```

중략...

참조파일 SamplePush>/app/manifests/AndroidManifest.xml

중략...

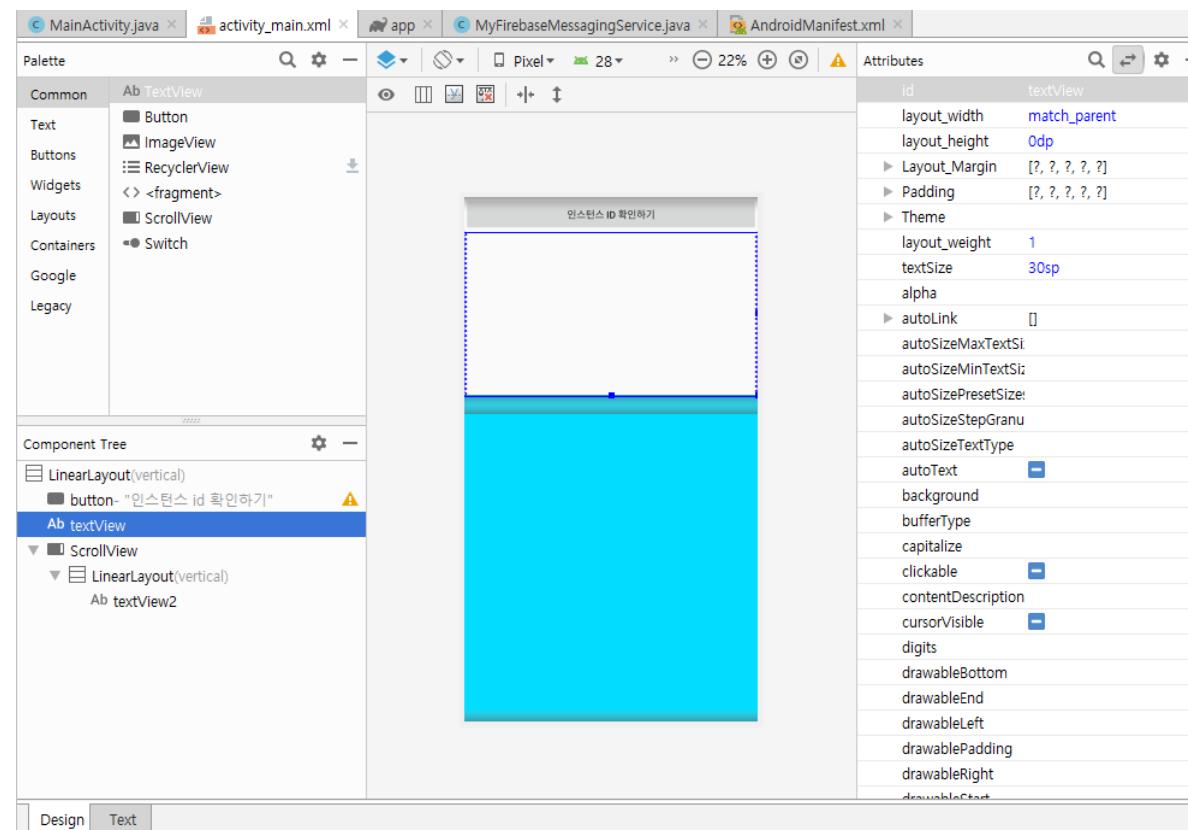
```
<uses-permission android:name="android.permission.INTERNET" />
```

중략...



화면 레이아웃 구성

- 받는 사람의 화면 레이아웃 구성





단말 등록 기능 추가

- 메인 액티비티에 단말 등록 및 등록ID 확인 기능 추가

```
FirebaseInstanceId.getInstance().getInstanceId().addOnSuccessListener(this,  
    new OnSuccessListener<InstanceIdResult>() {  
        @Override  
        public void onSuccess(InstanceIdResult result) {  
            String newToken = result.getToken();  
  
            println("등록id : " + newToken);  
        }  
    });  
  
Button button = findViewById(R.id.button);  
button.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        String instanceId = FirebaseInstanceId.getInstance().getId(); → ② 등록 id 값 확인을 위한  
        println("확인된 인스턴스 id : " + instanceId);  
    }  
});  
}
```

① 등록 id 확인을 위한
리스너 설정하기

② 등록 id 값 확인을 위한
메서드 호출하기



메시지 수신 기능 추가

• 메시지 수신하여 표시하는 기능 추가

```
@Override —→ ① 푸시 메시지를 받았을 때 그 내용 확인한 후 액티비티 쪽으로 보내는 메서드 호출  
public void onMessageReceived(RemoteMessage remoteMessage) {  
    Log.d(TAG, "onMessageReceived 호출됨.");  
  
    String from = remoteMessage.getFrom();  
    Map<String, String> data = remoteMessage.getData();  
    String contents = data.get("contents");  
  
    Log.d(TAG, "from : " + from + ", contents : " + contents);  
    sendToActivity(getApplicationContext(), from, contents);  
}  
  
private void sendToActivity(Context context, String from, String contents) {  
    Intent intent = new Intent(context, MainActivity.class);  
    intent.putExtra("from", from);  
    intent.putExtra("contents", contents);  
  
    intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK |  
        Intent.FLAG_ACTIVITY_SINGLE_TOP |  
        Intent.FLAG_ACTIVITY_CLEAR_TOP);  
  
    context.startActivity(intent);  
}
```

② 액티비티 쪽으로 데이터를 보내기 위해 인텐트 객체를 만들고 startActivity() 메서드 호출



메시지 수신 기능 추가

- 메인 액티비티에서 인텐트 받는 코드 추가

```
@Override —> ❶ 서비스로부터 인텐트를 받았을 때의 처리
protected void onNewIntent(Intent intent) {
    println("onNewIntent 호출됨");

    if (intent != null) {
        processIntent(intent);
    }

    super.onNewIntent(intent);
}

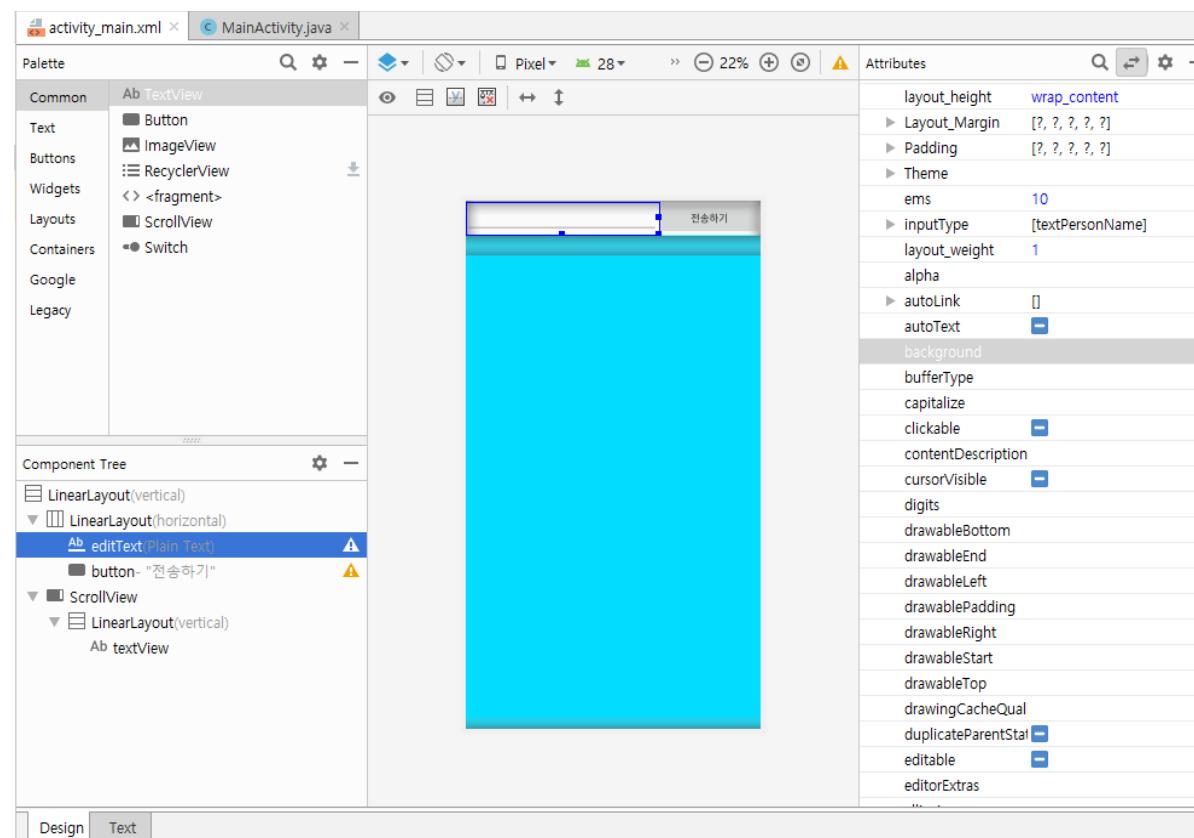
private void processIntent(Intent intent) {
    String from = intent.getStringExtra("from");
    if (from == null) {
        println("from is null.");
        return;
    }

    String contents = intent.getStringExtra("contents"); —> ❷ 보낸 데이터는 contents 키(Key)를
                                                                사용해 확인
    println("DATA : " + from + ", " + contents);
    textView.setText("[ " + from + " ]로부터 수신한 데이터 : " + contents);
}
```



메시지 송신을 위한 화면 레이아웃 구성

- 메시지 송신 앱의 화면 레이아웃 구성





Volley 라이브러리를 이용해 송신

- 외부 라이브러리 추가 및 Volley를 위한 기본 코드 추가

참조파일 SamplePushSend>/Gradle Scripts

```
implementation 'com.android.volley:volley:1.1.0'
```

참조파일 SamplePushSend>/app/java/org.techtown.push.send/MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
    EditText editText;  
    TextView textView;  
  
    static RequestQueue requestQueue;  
    static String regId;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        editText = findViewById(R.id.editText);  
        textView = findViewById(R.id.textView);  
  
        Button button = findViewById(R.id.button);  
        button.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                String input = editText.getText().toString();  
                send(input);  
            }  
        });  
  
        if (requestQueue == null) {  
            requestQueue = Volley.newRequestQueue(getApplicationContext());  
        }  
    }  
}
```



메시지 전송 기능 추가

- Volley를 이용한 메시지 전송

```
public void send(String input) {  
    JSONObject requestData = new JSONObject(); → ❶ 전송 정보를 담아둘 JSONObject 객체 생성  
  
    try {  
        requestData.put("priority", "high"); → ❷ 옵션 추가  
  
        JSONObject dataObj = new JSONObject();  
        dataObj.put("contents", input); ] ❸ 전송할 데이터 추가  
        requestData.put("data", dataObj);  
  
        JSONArray idArray = new JSONArray();  
        idArray.put(0, regId); ] ❹ 푸시 메시지를 수신할 단말의 등록 ID를  
        requestData.put("registration_ids", idArray); JSONArray에 추가한 후 requestData  
    } catch(Exception e) {  
        e.printStackTrace(); } 객체에 추가  
  
    sendData(requestData, new SendResponseListener() { → ❺ 푸시 전송을 위해 정의한 메서드 호출  
        @Override  
        public void onRequestCompleted() {  
            println("onRequestCompleted() 호출됨.");  
        }  
    }  
}
```



메시지 전송 기능 추가

• Volley를 이용한 메시지 전송

참조파일 SamplePushSend>/app/java/org.techtown.push.send/MainActivity.java

중략...

```
public interface SendResponseListener {
    public void onRequestStarted();
    public void onRequestCompleted();
    public void onRequestWithError(VolleyError error);
}

public void sendData(JSONObject requestData, final SendResponseListener listener) {
    JsonObjectRequest request = new JsonObjectRequest(→ ❶ Volley 요청 객체를 만들고 요청을 위한
        Request.Method.POST,                               데이터 설정
        "https://fcm.googleapis.com/fcm/send",
        requestData,
        new Response.Listener<JSONObject>() {
            @Override
            public void onResponse(JSONObject response) {
                listener.onRequestCompleted();
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                listener.onRequestWithError(error);
            }
        }
}
```



전송 키 값 입력

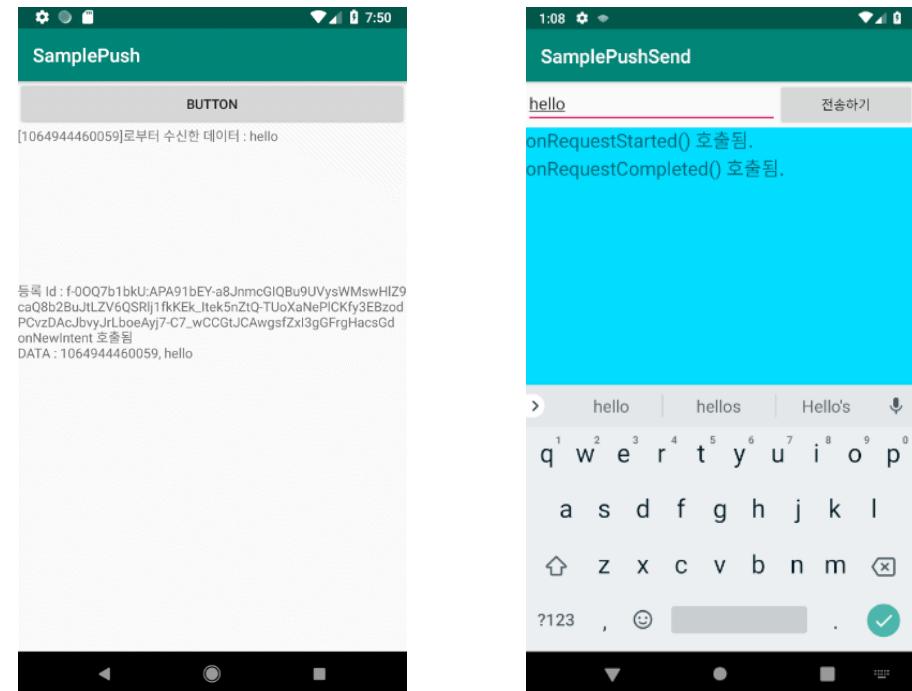
- Firebase 콘솔 페이지의 설정 메뉴에서 전송 키 값 확인하여 입력

The screenshot shows the Firebase Cloud Messaging settings page for a project named 'SamplePush'. The 'Cloud messaging' tab is selected. In the 'Server key' section, there is a table with two rows:

키	토큰
서버 키	AAAAAgP01tO8:APA91bG-LZuYlidaB8cpCpog0y47idO8NvbNha1rGrjHPDcoLiIK_hmI1xBAcicDTlrLZ6-qHR9pzDjejXBfZCSwCpDZVo2t6b2J0h0Sxp9AUaTrsgsC6nf690R-kTDJFQSTum4bCn5
이전 서버 키	AlzaSyCmYZekLoQY13Gg2EG08yIRU3VWKFn7TM
발신자 ID	554003969263



실행 화면





4.

센서 이해하기



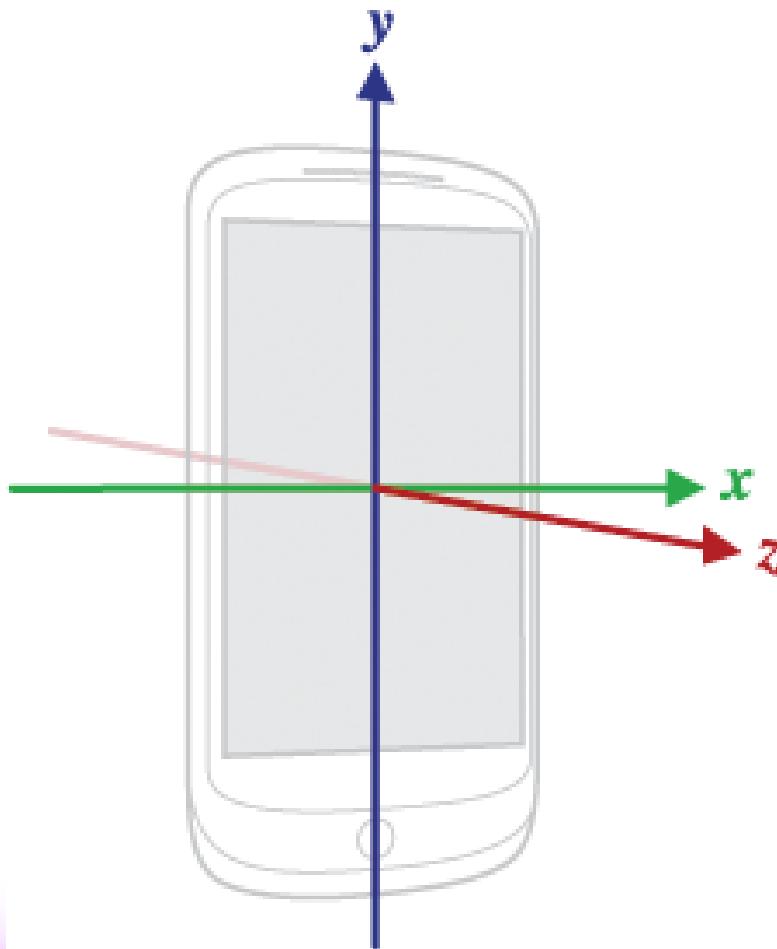
센서 이해하기

■ 안드로이드에서 지원하는 대표적인 센서들

센서 이름	상수	설명
가속 센서 (Accelerometer)	TYPE_ACCELEROMETER	가속도 감지, 외부의 충격량과 방향 감지(단위: m/s ²) (중력 + 선형가속)
자이로스코프 센서 (Gyroscope)	TYPE_GYROSCOPE	회전 정보 감지, 다양한 축을 따른 회전각 감지
중력 센서 (Gravity)	TYPE_GRAVITY	축의 방향과 중력 감지(단위: m/s ²)
조도 센서 (Light)	TYPE_LIGHT	빛의 세기 감지(단위: lux)
선형 가속 센서 (Linear Acceleration)	TYPE_LINEAR_ACCELERATION	각 축에 따른 가속도를 감지하는 3차원 벡터(단위: m/s ²)
근접 센서 (Proximity)	TYPE_PROXIMITY	근접한 물체와의 거리 감지(단위: cm) 단말에 따라 가깝거나 먼 정도만 표시할 수도 있음
온도 센서 (Temperature)	TYPE_TEMPERATURE	온도 감지
방향 센서 (Orientation)	TYPE_ORIENTATION	방향각 감지(단위: degree)



가속도 센서의 좌표계

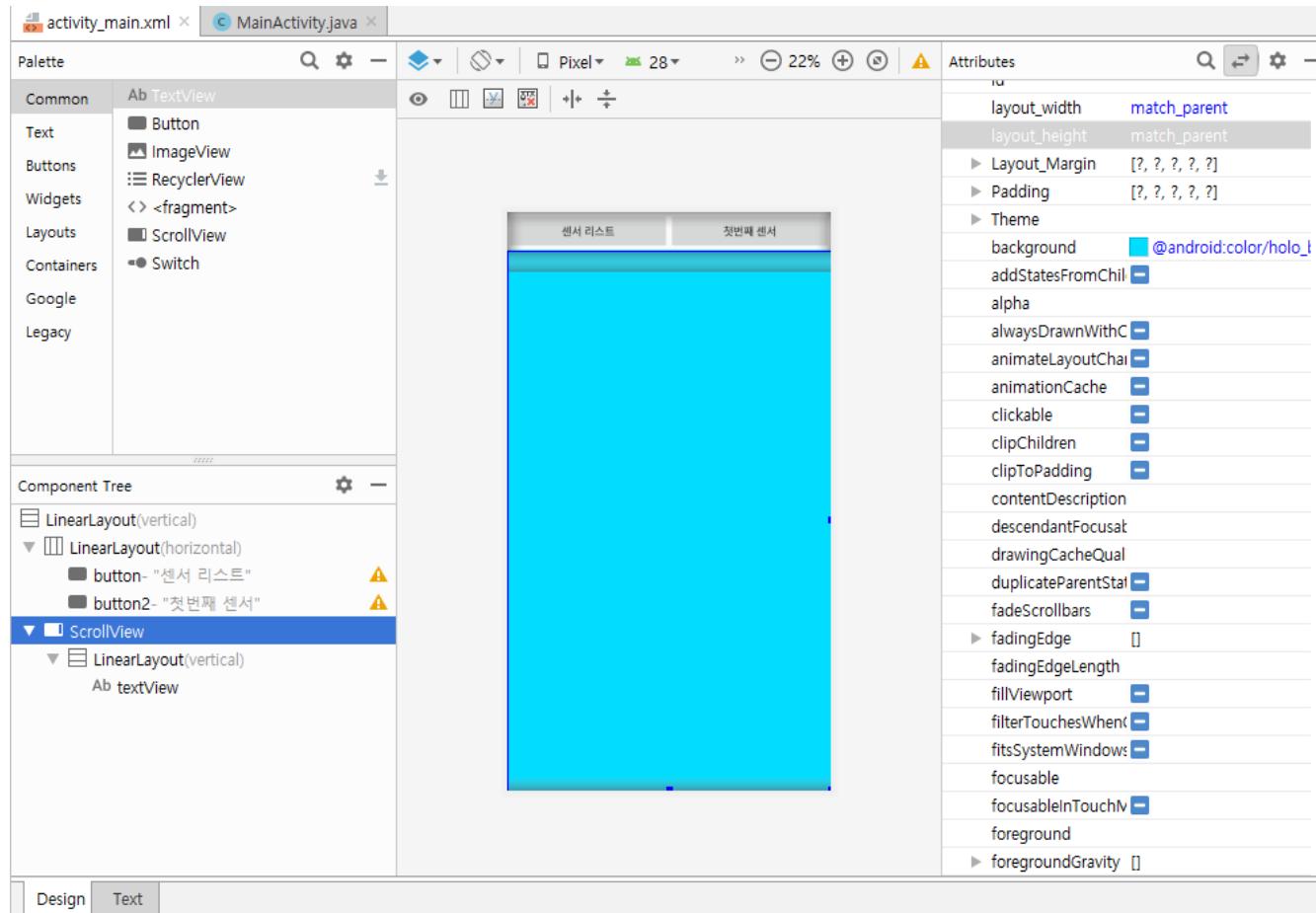


가속도 센서에서 사용하는 좌표계

<Reference>
SensorManager
Sensor
SensorEvent



화면 레이아웃 구성





센서 이해하기 예제 – 메인 액티비티 만들기

```
...  
public class MainActivity extends ListActivity {  
    public static final String TAG = "SampleSensorActivity";  
    SensorManager manager = null;  
    List<Sensor> sensors = null;  
    SensorListAdapter adapter = null;  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        manager = (SensorManager) getSystemService(SENSOR_SERVICE);  
        sensors = manager.getSensorList(Sensor.TYPE_ALL);  
        adapter = new SensorListAdapter(this, R.layout.listitem, sensors);  
        setListAdapter(adapter);  
    }  
}
```

1 시스템 서비스인 센서 매니저 객체 참조

2 단말에서 지원하는 모든 종류의 센서 정보 확인

3 알아낸 센서 객체를 이용해 리스트뷰에 표시하기 위한 어댑터 객체 생성



센서 이해하기 예제 – 메인 액티비티 만들기 (계속)

```
protected void onListItemClick(ListView l, View v, int position, long id) {  
    super.onListItemClick(l, v, position, id);
```

```
Sensor sensor = sensors.get(position);
```

4

선택한 아이템의 센서 객체 확인

```
String sensorName = sensor.getName();
```

```
Log.d(TAG, "selected sensor: " + sensorName);
```

```
Intent intent = new Intent(this, SensorDataActivity.class);
```

```
intent.putExtra(SensorDataActivity.SENSOR_INDEX, position);
```

```
startActivity(intent);
```

```
}
```

```
}
```

5

선택한 센서의 값을 확인하기 위한 액티비티 띄우기



실행 화면



센서 리스트 화면

<Reference>

Public List<Sensor> getSensorList (int type)



센서 이해하기 예제 – 데이터 확인

```
...  
public class SensorDataActivity extends Activity implements SensorEventListener {  
...  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.data);  
...  
    Intent passedIntent = getIntent();  
    if (passedIntent != null) {  
        sensorIndex = getIntent().getIntExtra(SENSOR_INDEX, 0);  
        sensorName = sensors.get(sensorIndex).getName();  
        txtSensorName.setText(sensorName);  
    }  
}
```

1

SensorEventListener 인터페이스를
구현하는 액티비티 클래스 정의

2

전달받은 센서 객체 확인 후
센서 이름 표시



센서 이해하기 예제 – 데이터 확인 (계속)

```
protected void onResume() {  
    super.onResume();
```

```
    manager.registerListener(this, sensors.get(sensorIndex),  
        SensorManager.SENSOR_DELAY_UI);  
}
```

```
protected void onStop() {
```

```
    manager.unregisterListener(this);  
    super.onStop();  
}
```

```
public void onAccuracyChanged(Sensor sensor, int accuracy) {  
    txtSensorAccuracy.setText("Sensor Accuracy: " +  
        getSensorAccuracyAsString(accuracy));  
}
```

3 화면에 보이기 전에 센서이벤트 리스너 설정

4 액티비티 종지 전에 센서이벤트 리스너 해제

5 센서 정확도가 변할 때 화면에 표시



센서 이해하기 예제 – 데이터 확인 (계속)

```
public void onSensorChanged(SensorEvent event) {    6 센서 값이 변할때 화면에 표시  
    String data = "Sensor Timestamp: " + event.timestamp + "\n\n";  
    for(int index=0; index<event.values.length; ++index) {  
        data += ("Sensor Value #" + (index + 1) + ": " + event.values[index] + "\n");  
    }  
    txtSensorValues.setText(data);  
}  
...  
}
```



주요 이벤트

<Reference>

```
abstract void onAccuracyChanged(Sensor sensor, int accuracy)
```

```
abstract void onSensorChanged(SensorEvent event)
```

<Reference>

```
SensorManager.SENSOR_STATUS_ACCURACY_HIGH
```

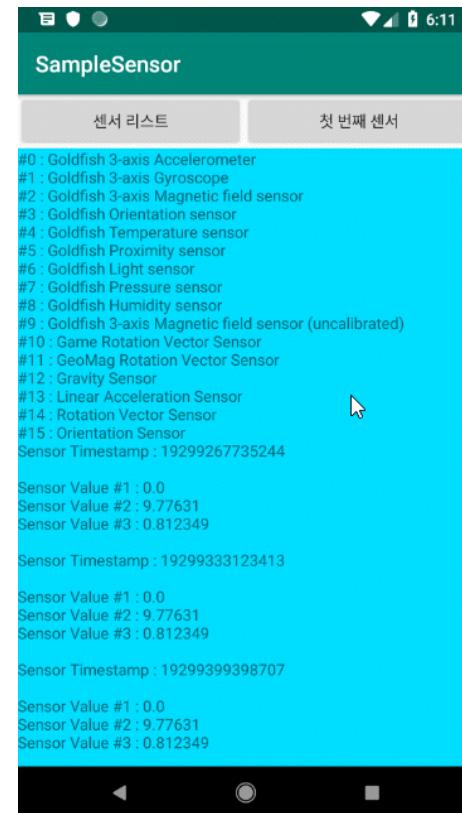
```
SensorManager.SENSOR_STATUS_ACCURACY_MEDIUM
```

```
SensorManager.SENSOR_STATUS_ACCURACY_LOW
```

```
SensorManager.SENSOR_STATUS_UNRELIABLE
```



실행 화면 – 가속도 센서





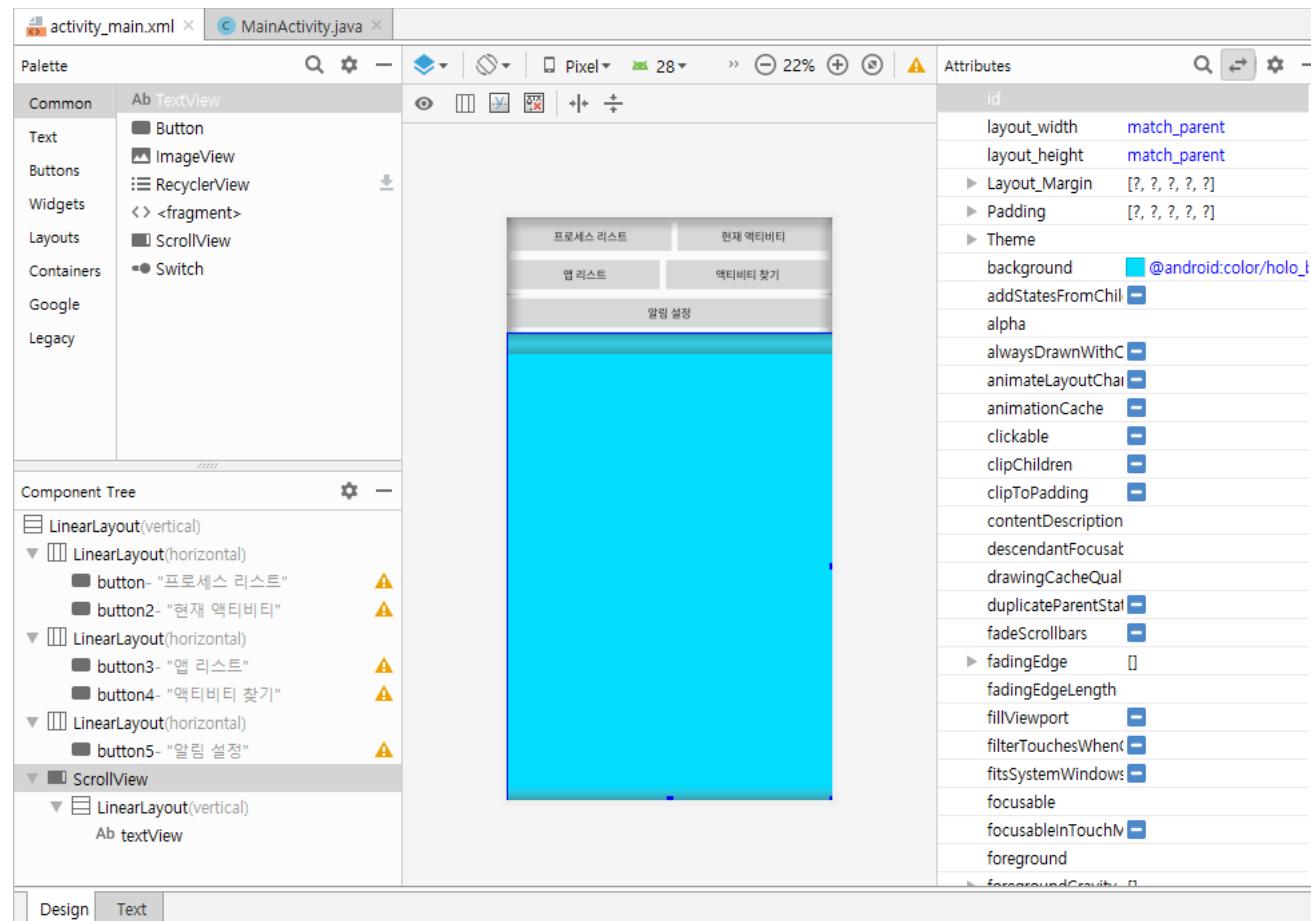
5.

시스템 서비스 활용하기



화면 레이아웃 구성

- 상단에 버튼들 배치





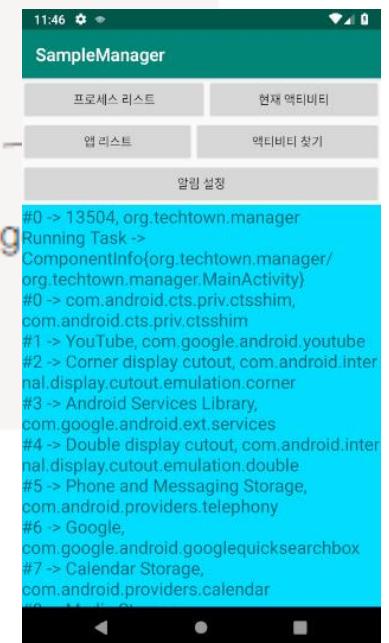
알람 등록

- AlarmManager를 이용해 알람 등록

```
public void setAlarm() {  
    AlarmManager manager = (AlarmManager) getSystemService(Context.ALARM_SERVICE);  
  
    Intent intent = new Intent(this, MainActivity.class);  
    PendingIntent pendingIntent = PendingIntent.getActivity(this, 101, intent,  
        PendingIntent.FLAG_UPDATE_CURRENT);  
  
    manager.set(AlarmManager.RTC, System.currentTimeMillis() + 60000, pendingIntent); -  
}  
  
public void println(String data) { textView.append(data + "\n"); }  
}
```

② AlarmManager

알람 등록하기





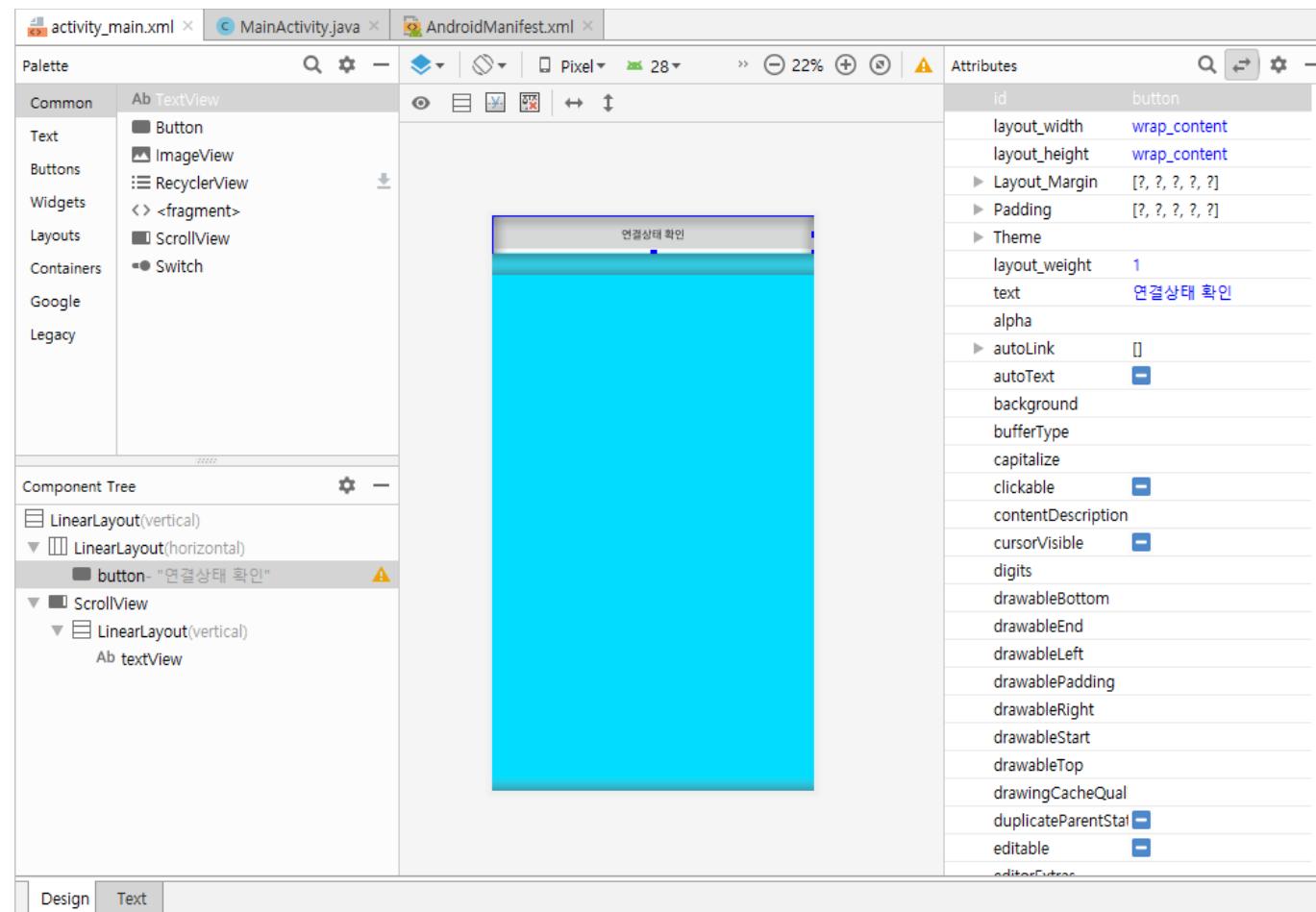
6.

네트워크 기능 활용하기



화면 레이아웃 구성

- 상단에 버튼들 배치





코드 추가

- 인터넷 연결상태 확인하는 코드 추가

```
public void checkConnectivity() {  
    ConnectivityManager manager = → ① ConnectivityManager 객체 확인하기  
        (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);  
    NetworkInfo info = manager.getActiveNetworkInfo();  
    if (info != null) {  
        if (info.getType() == ConnectivityManager.TYPE_WIFI) {  
            println("WiFi로 설정됨");  
        } else if (info.getType() == ConnectivityManager.TYPE_MOBILE) {  
            println("일반망으로 설정됨");  
        }  
  
        println("연결 여부: " + info.isConnected()); → ② 연결 여부 확인하기  
    } else {  
        println("데이터통신 불가");  
    }  
}
```



코드 추가

- WiFi 상태 확인 가능

```
IntentFilter filter = new IntentFilter();
filter.addAction(WifiManager.WIFI_STATE_CHANGED_ACTION);
filter.addAction(WifiManager.NETWORK_STATE_CHANGED_ACTION);
registerReceiver(wifiReceiver, filter); —→ ① 코드에서 수신자 등록하기
}

class WiFiReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();
        if (action.equals(WifiManager.WIFI_STATE_CHANGED_ACTION)) { —→ ② WiFi 상태 체크하기
            int state = intent.getIntExtra(WifiManager.EXTRA_WIFI_STATE, -1);
            if (state == WifiManager.WIFI_STATE_ENABLED) {
                println("WiFi enabled");
            } else if (state == WifiManager.WIFI_STATE_DISABLED) {
                println("WiFi disabled");
            }
        } else if (action.equals(WifiManager.NETWORK_STATE_CHANGED_ACTION)) {
            NetworkInfo info = intent.getParcelableExtra(WifiManager.EXTRA_NETWORK_INFO);
            WifiManager manager =
                (WifiManager) getApplicationContext().getSystemService(Context.WIFI_SERVICE);
            String ssid = manager.getConnectionInfo().getSSID();
```

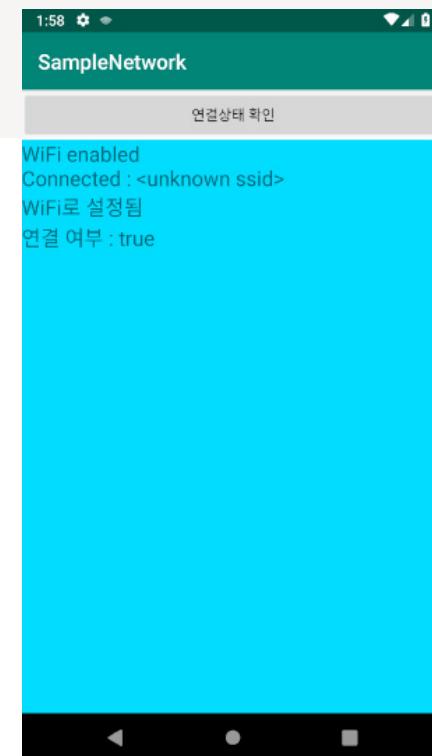


권한 추가 및 실행

- WiFi 상태 확인을 위한 권한 필요

참조파일 SampleNetwork>/app/manifests/AndroidManifest.xml

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
```





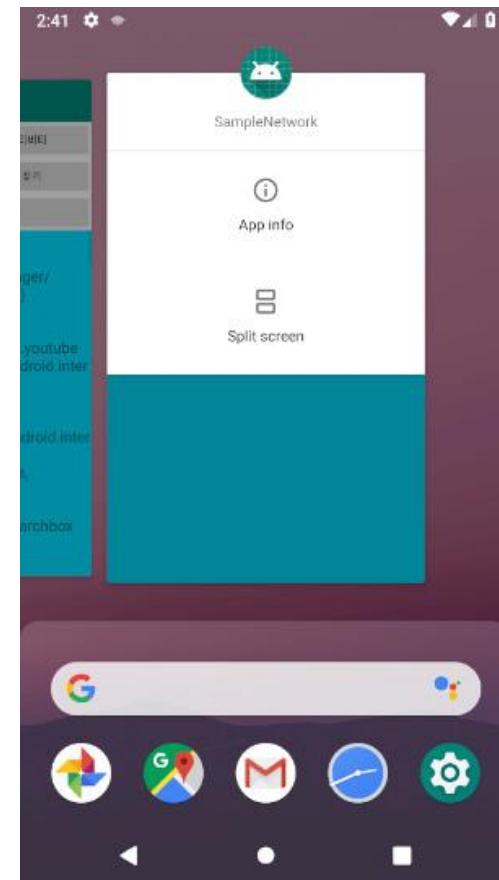
7.

다중 창 지원하기



다중 창 띄우는 기능

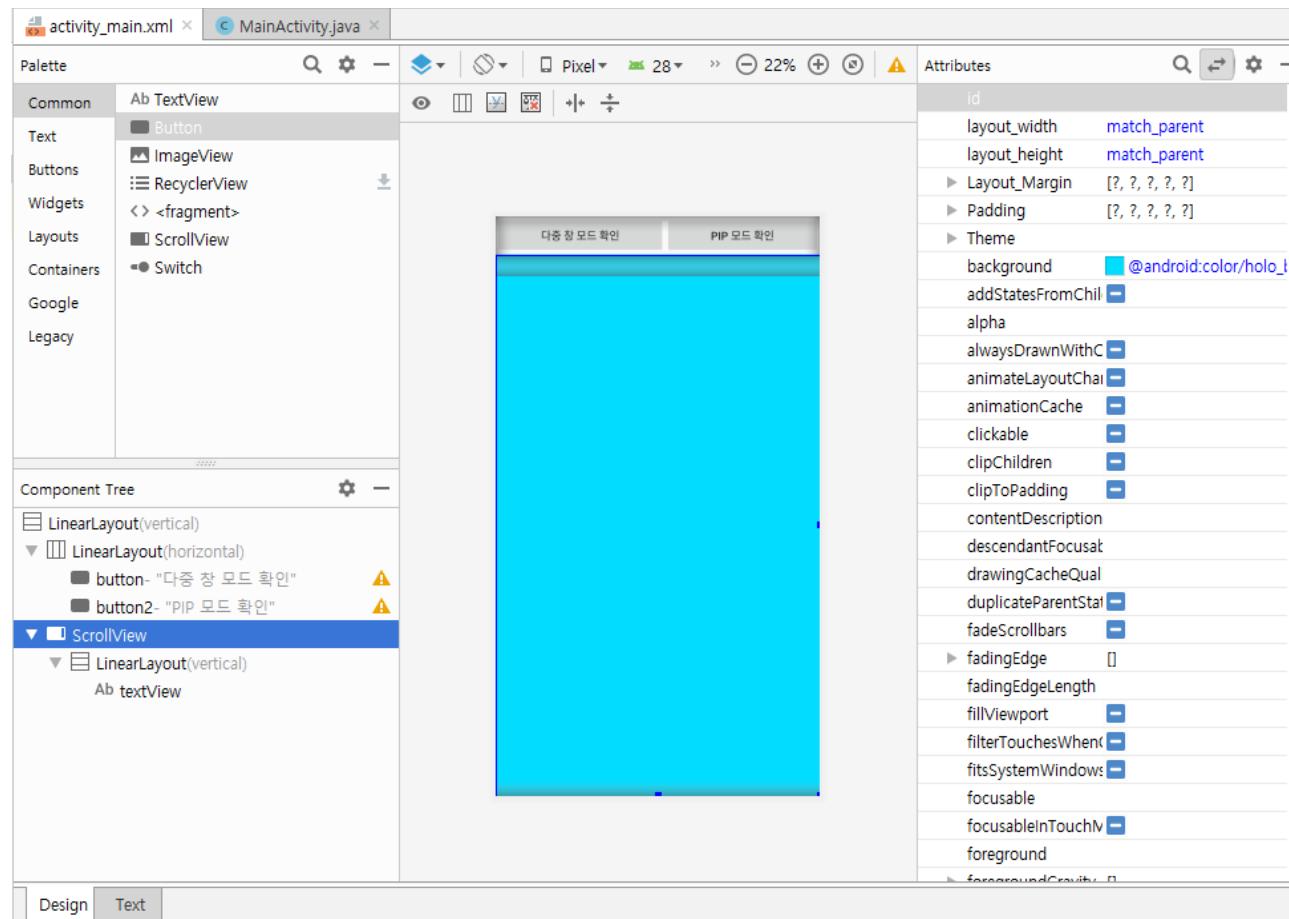
- 단말에서 다중 창을 띄우는 기능 사용





화면 레이아웃 만들기

• 화면에 버튼과 텍스트뷰 추가





코드에서 다중창 여부 확인

- 다중 창 모드 사용 여부 확인 가능

```
Button button = findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (Build.VERSION.SDK_INT >= 24) {
            boolean isIn = isInMultiWindowMode(); → ❶ 다중 창 모드 여부 확인하기
            println("다중 창 모드 여부: " + isIn);
        }
    }
});
```

```
@Override
public void onClick(View v) {
    if (Build.VERSION.SDK_INT >= 24) {
        boolean isIn = isInPictureInPictureMode();
        println("PIP 모드 여부: " + isIn);
    }
}
```

