

# 처음부터 시작하는 안드로이드 스튜디오



Android Studio  
버전 4.1 기준

선린인터넷고등학교  
IRIS

# 목 차

1

Android Studio  
설치

1-1	Android Studio 설치	7
1-2	Android Studio 시작	12

2

Activity

2-1	activity_main.xml	17
2-2	strings.xml	20
2-3	MainActivity.java	22

3

Intent

3-1	인텐트란?	26
-----	-------	----

4

View Components

4-1	View 요소 이해하기	30
4-2	Layout	34
4-3	View	38

<b>5</b>	<b>Material Components</b>	
	5-1 Material Design	46
	5-2 Material Components	47
<b>6</b>	<b>SharedPreference</b>	
	6-1 SharedPreference란?	53
	6-2 데이터 저장/불러오기	54
	6-3 회원가입 / 로그인 만들기	56
<b>7</b>	<b>RecyclerView</b>	
	7-1 RecyclerView란?	58
	7-2 RecyclerView 사용	59
<b>8</b>	<b>Fragment</b>	
	8-1 프래그먼트 파헤치기	64
	8-2 프래그먼트 생성하기	65
	8-3 프래그먼트 만들기	67
<b>9</b>	<b>View Components</b>	
	9-1 Permission이란	71
	9-2 Permission 부여하기	73
	9-3 Permission 예제	74

# 10

## Firebase

10-1	Firebase 개요 및 설치	80
10-2	Firebase 데이터 수신	89
10-3	Firebase 데이터 송신	92

# 11

## Github

11-1	Git 설치하기	99
11-2	Github 저장소 생성하기	101
11-3	Github 세팅하기	102

# X

## 기타

X-1	manifest	109
X-2	build.gradle	111
X-3	Values	113
X-4	다크모드	114
X-5	스마트폰 연결	117

# 부록

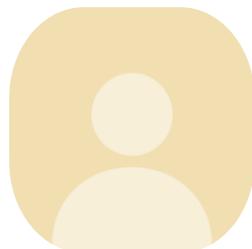
## 출처 및 참고자료

122

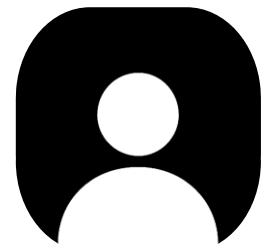
# 저자



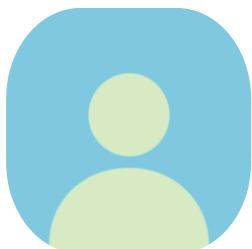
20' 부부장  
김우진



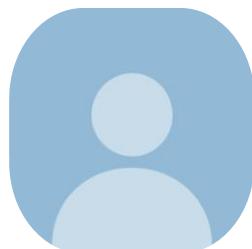
20' 부장  
이종우



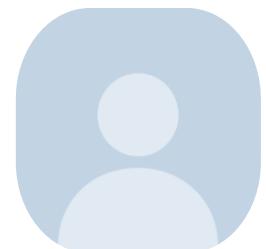
개발자  
정해성



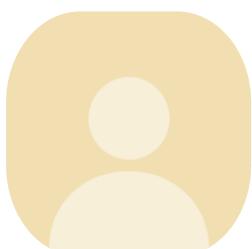
21' 부부장  
이예희



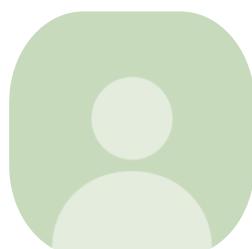
21' 부장  
박선균



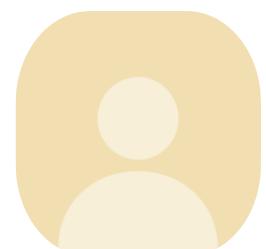
개발자  
심희윤



개발자  
박찬웅



개발자  
백영대



개발자  
이현주



개발자  
채예린



개발자  
최경진



# 1

## Android studio 설치

안드로이드 앱 제작을 위한 Google에서 만든 안드로이드 스튜디오를 사용하기 위해 개발환경 구축을 해보자.

by. 심희윤

**1-1**

# Android studio 설치

**안드로이드 스튜디오를 설치해 보자.**

## 1. 안드로이드 스튜디오 다운로드

<https://developer.android.com/studio> 사이트에 접속한다.



Android Studio provides the fastest tools for building apps on every type of Android device.

[DOWNLOAD ANDROID STUDIO](#)

4.1.1 for Windows 64-bit (896 MB)

[DOWNLOAD OPTIONS](#)

[RELEASE NOTES](#)

안드로이드 스튜디오를 다운로드할 컴퓨터가 64bit라면 [DOWNLOAD ANDROID STUDIO]를 클릭하고, 32bit라면 [DOWNLOAD OPTIONS]을 클릭한다.

[DOWNLOAD OPTIONS]를 클릭하면 운영체제에 맞춰 안드로이드 스튜디오를 선택해 설치할 수 있다.

## Android Studio downloads

Platform	Android Studio package	Size	SHA-256 checksum
Windows (64-bit)	<a href="#">android-studio-ide-201.6953283-windows.exe</a> Recommended	896 MB	88d1576b5f672e67035bc
	<a href="#">android-studio-ide-201.6953283-windows.zip</a> No .exe installer	900 MB	68717853fc1a99b51a009
Mac (64-bit)	<a href="#">android-studio-ide-201.6953283-mac.dmg</a>	877 MB	e9ca26f3a3da9610dcaa2
Linux (64-bit)	<a href="#">android-studio-ide-201.6953283-linux.tar.gz</a>	883 MB	68032184959c54576f119
Chrome OS	<a href="#">android-studio-ide-201.6953283-cros.deb</a>	743 MB	7621210773ef2ddaecbb

See the [Android Studio release notes](#). More downloads are available in the [download archives](#).

### 1-1. exe 패키지 선택한 경우

본인은 상기 사용 약관을 읽었으며 이에 동의합니다.

**다운로드: ANDROID STUDIO (WINDOWS 용)**

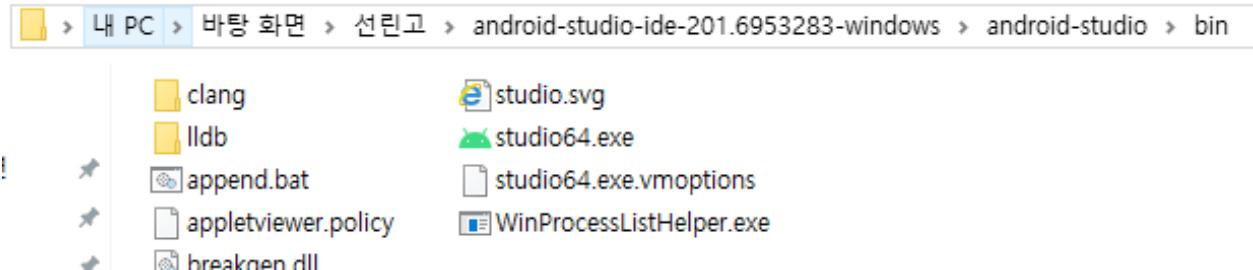
약관의 동의에 체크를 하고 다운로드를 시작한다.

 [android-studio-ide-201.6953283-windows.exe](#)

다운로드가 완료된 후 메시지를 클릭하면 안드로이드 스튜디오를 설치한다.

## 1-2. zip 패키지 선택한 경우

다운로드한 압축 파일을 푼다.



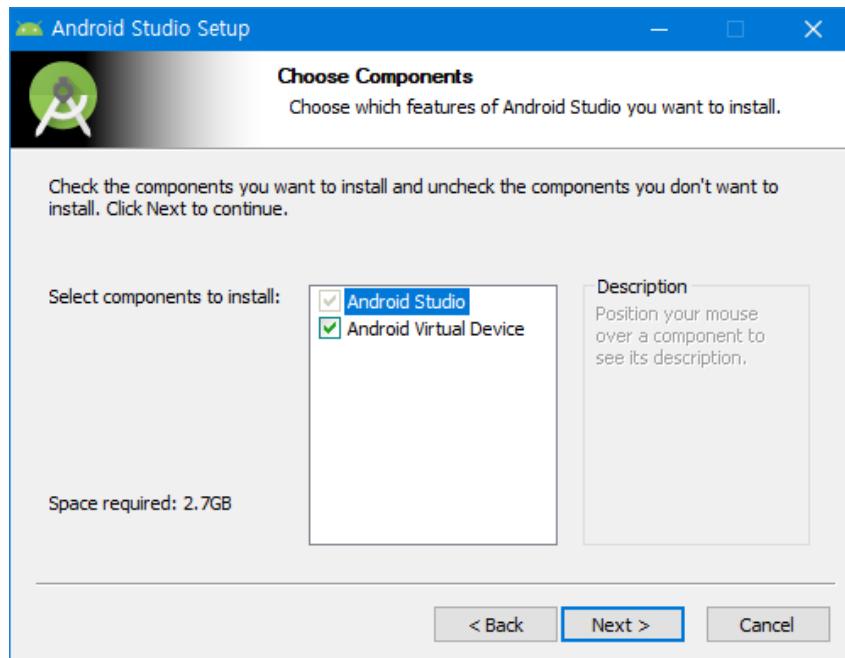
Bin 폴더에 들어가면 실행 파일이 존재한다. 컴퓨터가 64bit일 경우 studio64를 실행하면 안드로이드 스튜디오 설치가 실행된다.

## 설치 과정

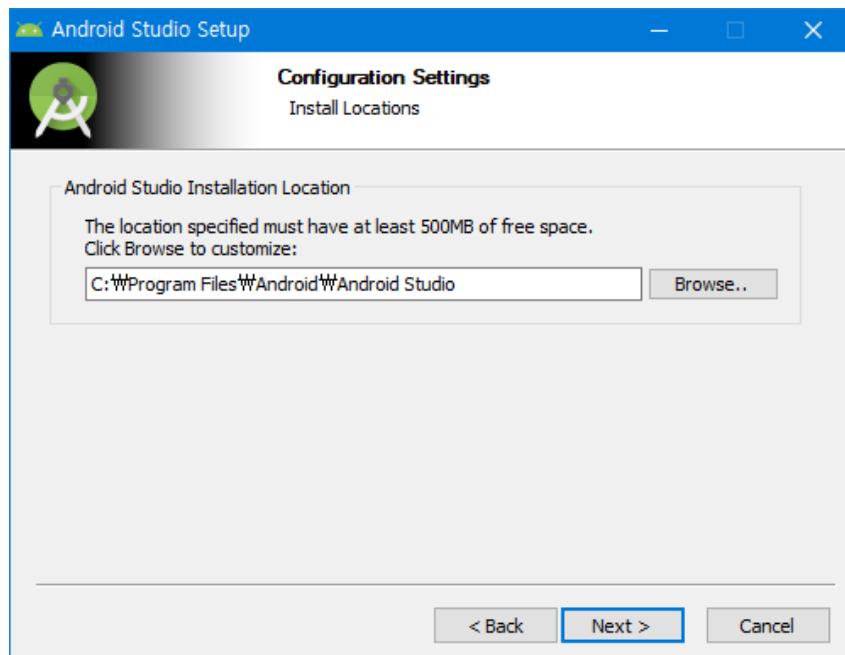
### 2. Android Studio Setup



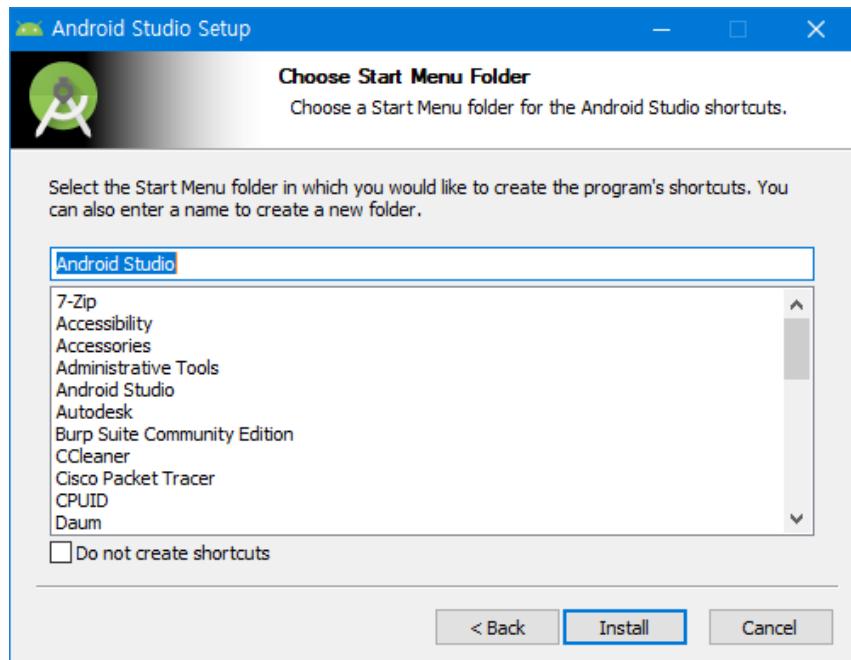
해당 창이 나오면 [Next] 버튼을 클릭한다.



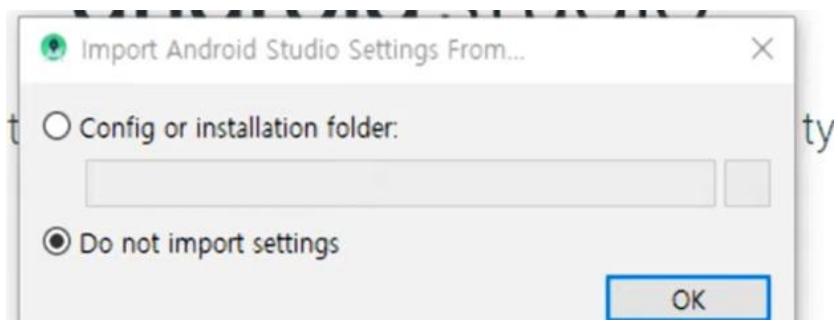
Android Virtual Device를 체크하고 [Next] 버튼을 클릭한다.



Android Studio를 설치할 경로를 지정한 후 [Next] 버튼을 클릭한다.  
+ 디폴트 경로는 C:\Program Files\Android\Android Studio



[Install] 버튼을 클릭한다. 이후 설치가 완료되면 [Next] 버튼을 클릭한다. [Finish] 버튼을 누른 후 안드로이드 스튜디오가 실행된다.



[Do not import settings]를 선택하고 [OK] 버튼을 클릭한다. [Next] 버튼을 클릭 후 설치 유형을 Standard로 선택하고 [Next], 원하는 UI 테마를 선택한 후 [Next] 버튼을 누른다.

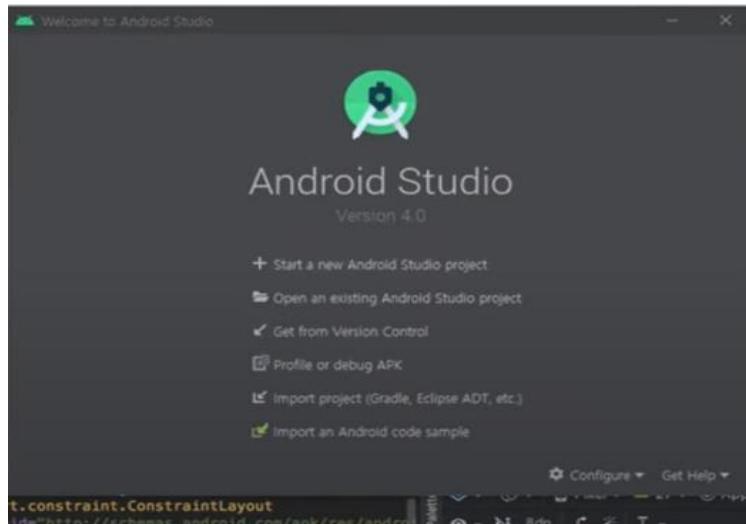
Verify Settings 창에서 [finish] 버튼을 누르고 설치가 완료되면 [Finish] 버튼을 누른다.

1-2

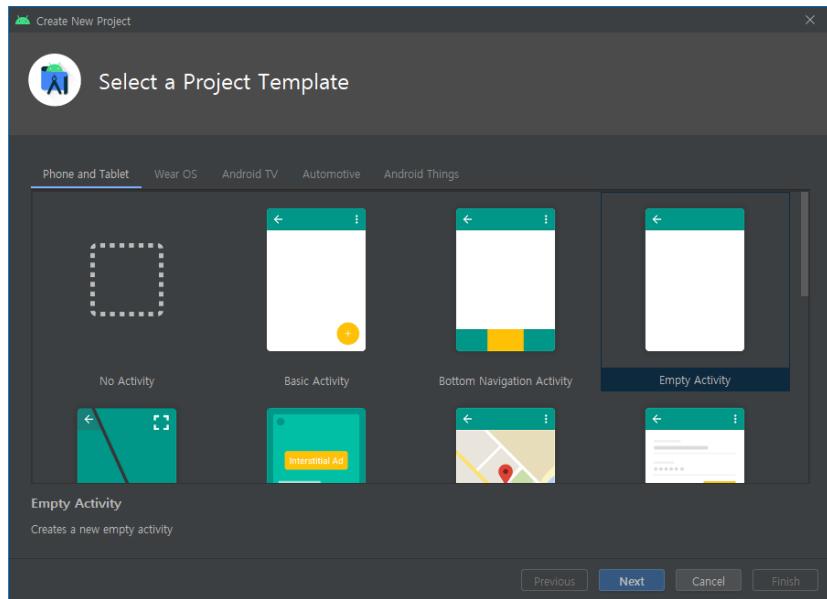
# Android studio 시작

## 안드로이드 스튜디오 시작하기

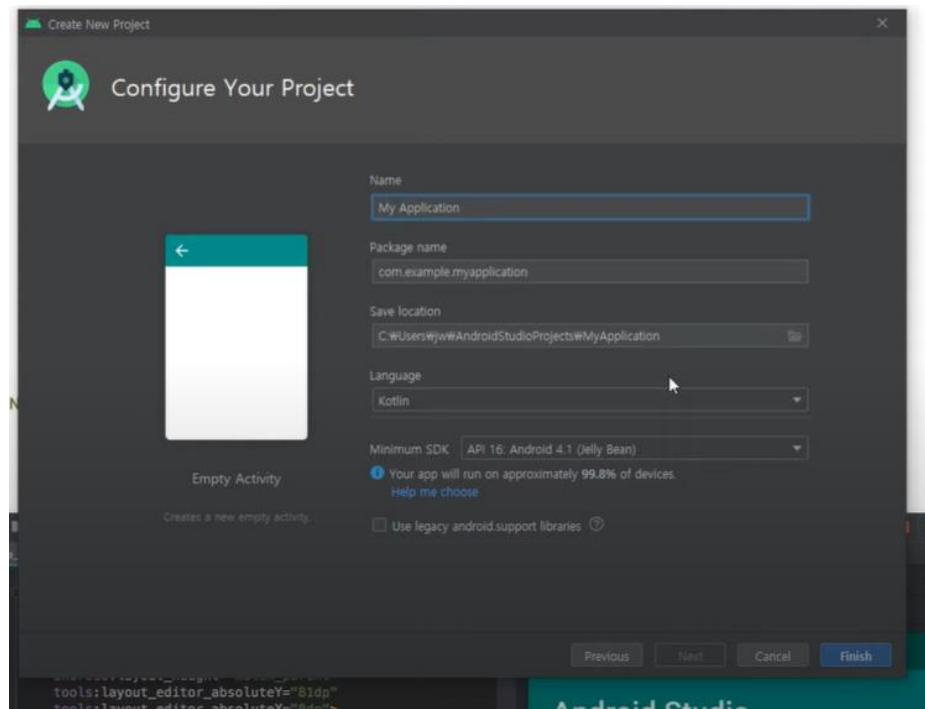
### 1. 새 프로젝트 만들기



안드로이드 스튜디오가 실행되면 [Start a new Android Studio project]를 클릭하여 새 프로젝트를 만든다.

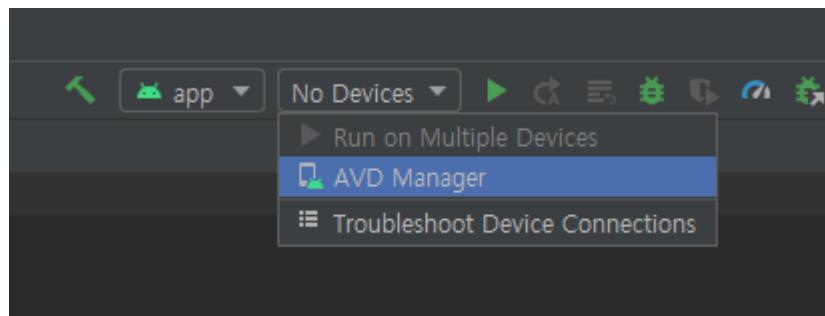


Empty Activity를 선택한다.

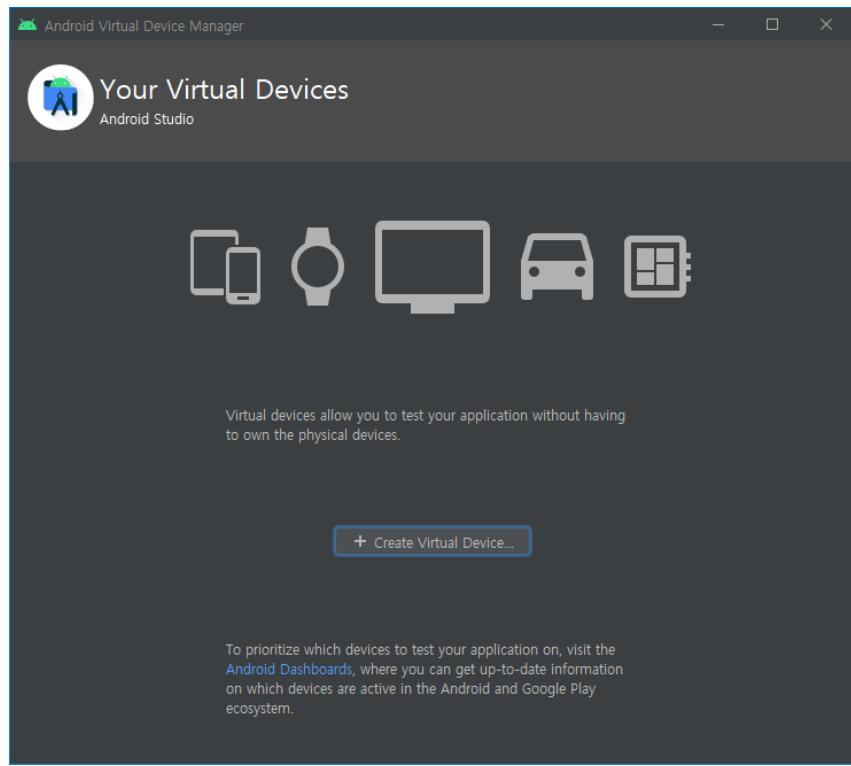


해당 창에서 Language를 Java로 선택한 뒤 [Next] 버튼을 클릭한다.

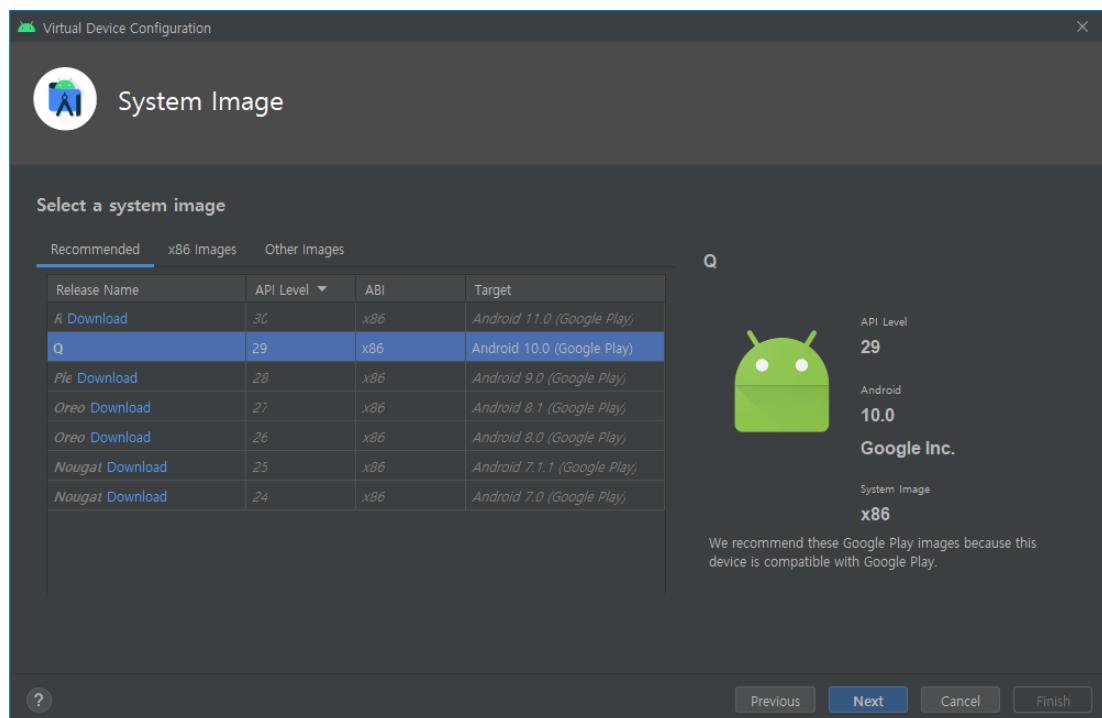
## 2. 에뮬레이터 설정



로딩이 다 끝난 후, 우측 상단의 No Devices를 클릭하고 AVD Manager를 클릭한다.



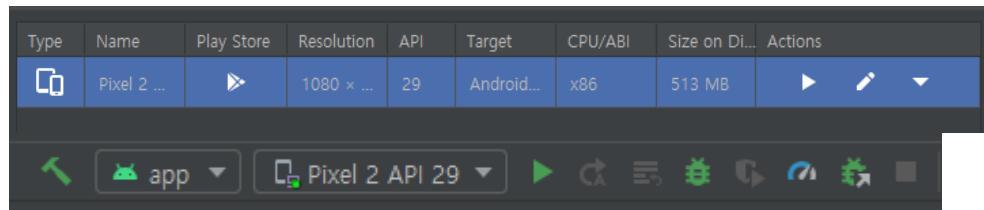
[Create Virtual Device..]를 클릭한다.  
Select Hardware창에서 [Next] 버튼을 클릭하고,



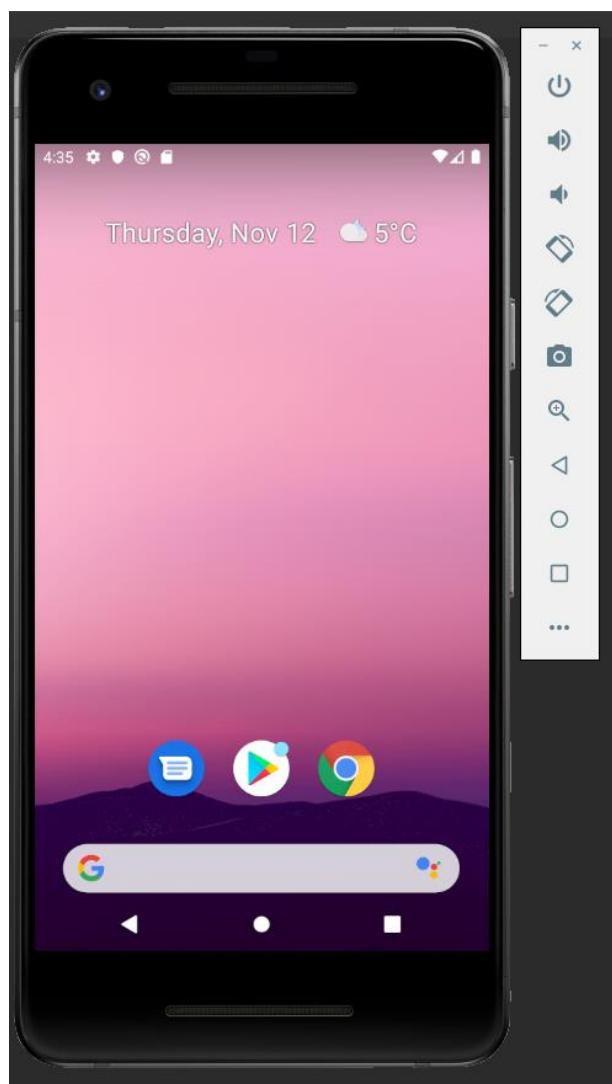
해당 창에서 자신이 사용하고 있는(앱을 구동할) 스마트폰의 API 버전보다 높은 버전을 찾아 Download를 누른다. 다음 약관에 동의하고 [Next]버튼을 클릭한다. 이후 [Finish]를 두 번 눌러 설정을 마무리 한다.

# 확인하기

설정 마무리 후 에뮬레이터를 > 버튼을 클릭하거나 우측 상단의 > 버튼을 클릭하여 에뮬레이터가 제대로 작동하는지 확인한다.



# 결과



+ 처음 실행시 시간이 걸릴 수 있다.

# 2

## Activity

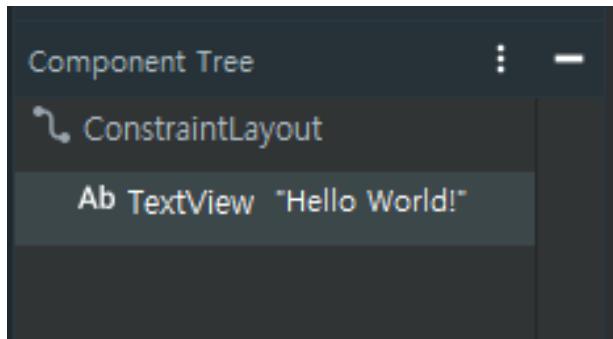
Activity에 대해 알아봅시다.

by. 정해성

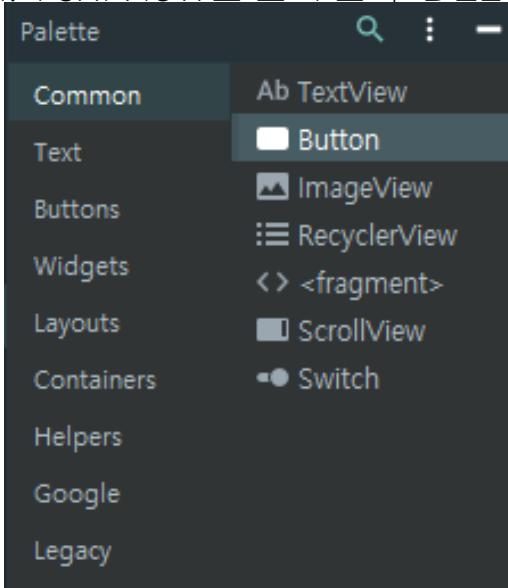
# 2-1

# activity\_main.xml

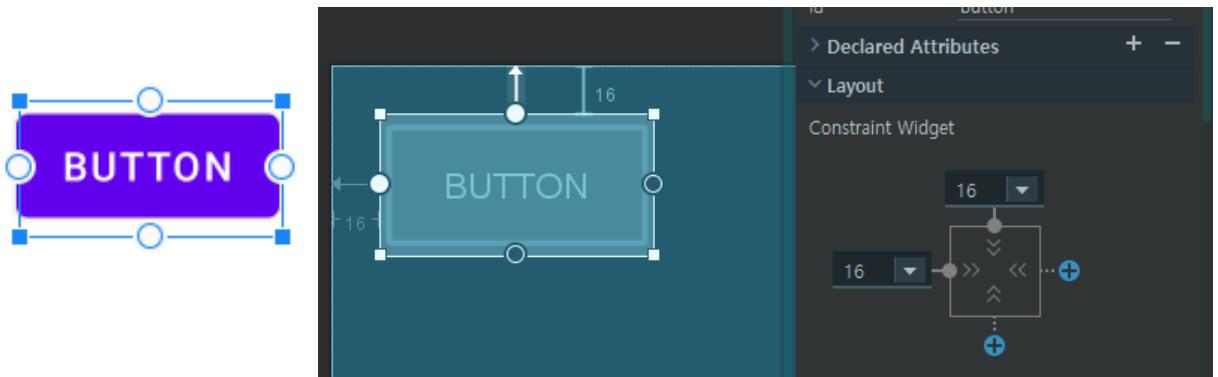
Android Studio Layout Editor를 이용  
해서 activity\_main.xml 수정



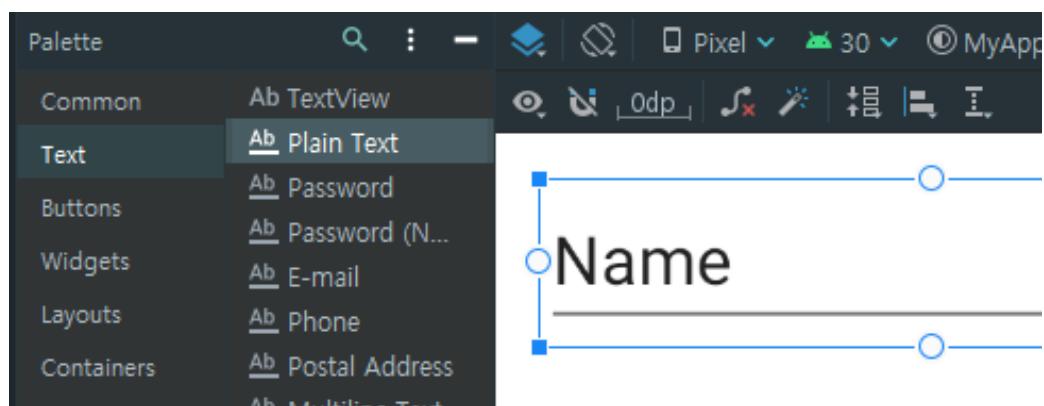
1. TextView를 클릭 한 후 DEL 키를 눌러 삭제하기



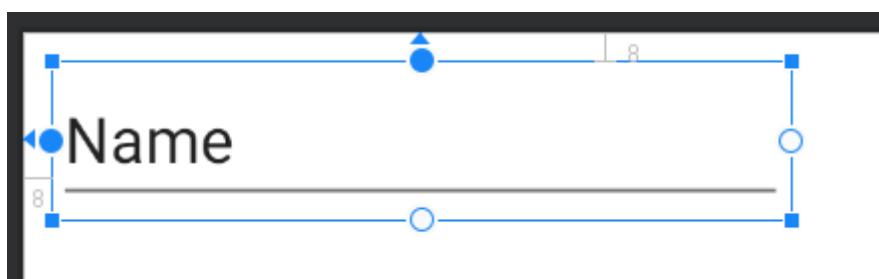
2. Palette 메뉴에서 Common > Button 찾아서 빈 화면에 드래그 해서 넣기



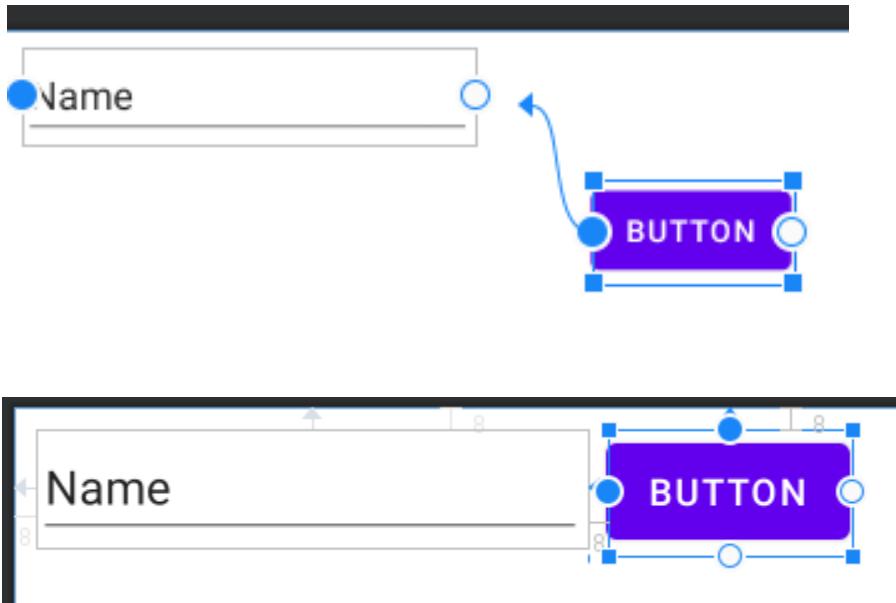
3. 버튼의 모서리의 앵커(동그란 점)를 가장자리로 끌어서 관계를 설정할 수 있습니다.



4. Text > Plain Text 드래그해서 화면에 가져다 놓은 후



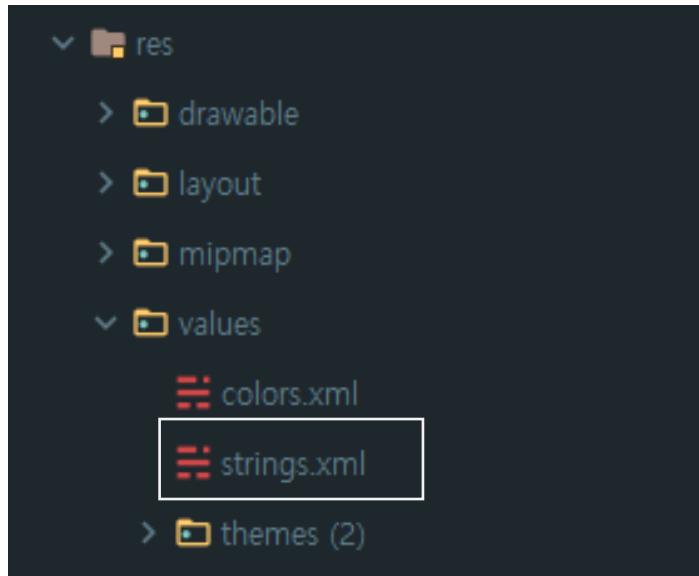
5. 위와 왼쪽에 앵커 지정하기



6. 버튼의 앵커를 EditText 오른쪽에 끌고 와서 사진처럼 구성하면 xml 완성

➔ 자세한 xml 사용 방법은 다음 챕터(View Components)에서 살펴 볼 예정입니다.

# 2-2 strings.xml



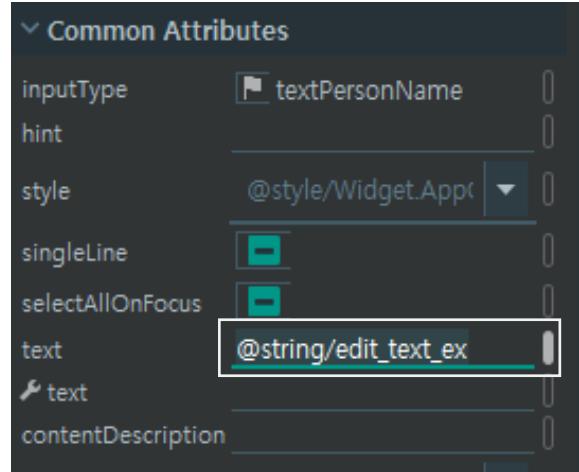
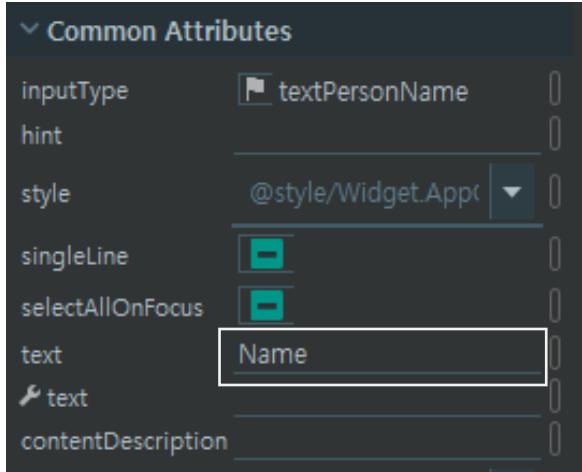
1. strings.xml로 이동

The screenshot shows the Android Studio editor with three tabs: 'activity\_main.xml', 'strings.xml', and 'MainActivity.java'. The 'strings.xml' tab is selected and highlighted with a red rectangle. Below the tabs, it says 'Edit translations for all locales in the translations editor.' The code editor displays the following XML:

```
1 <resources>
2     <string name="app_name">My Application</string>
3     <string name="edit_text_ex">Enter anything</string>
4 </resources>
```

A red rectangle highlights the string element at line 3.

2. 흰 박스 친 부분 타이핑하기

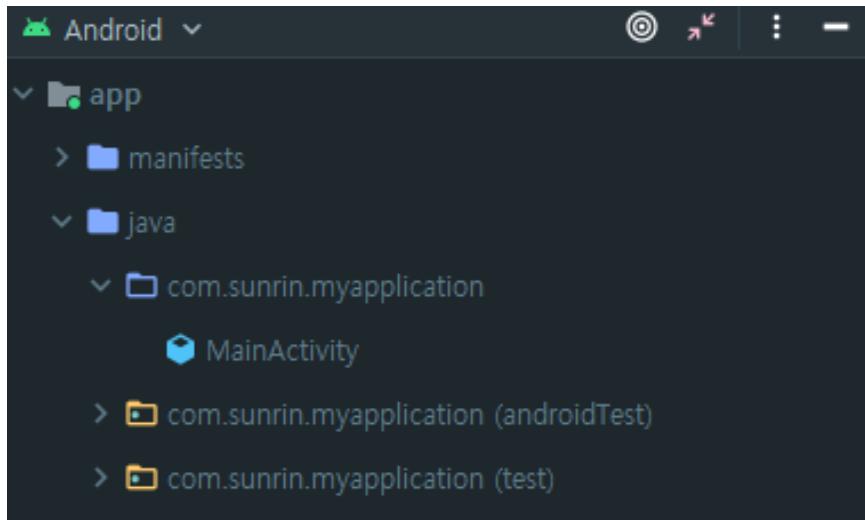


3. activity\_main으로 돌아와서 EditText의 Text값을 위처럼 변경하기



4. 결과

# 2-3 MainActivity.java



## 1. MainActivity.java로 이동

A screenshot of the Android Studio code editor. The tab bar shows 'activity\_main.xml', 'strings.xml', and 'MainActivity.java', with 'MainActivity.java' currently selected. The code editor displays the following Java code for the MainActivity class:1 package com.sunrin.myapplication;  
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}  
13The code is color-coded, with keywords in blue and identifiers in purple. A red error icon is visible near the line 'super.onCreate(savedInstanceState);'.

## 2. 기본 화면

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    Button button = findViewById(R.id.button);  
    // 버튼이 눌렸을 때 동작  
    button.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View view) {  
            Toast.makeText(MainActivity.this, "Hello World!",  
Toast.LENGTH_SHORT).show();  
        }  
    });  
}
```

### 3. 위와 같이 코드 입력하고 테스트하기



### 4. 결과 (화면 아래에 'Hello World!'라고 토스트 메시지가 뜸)

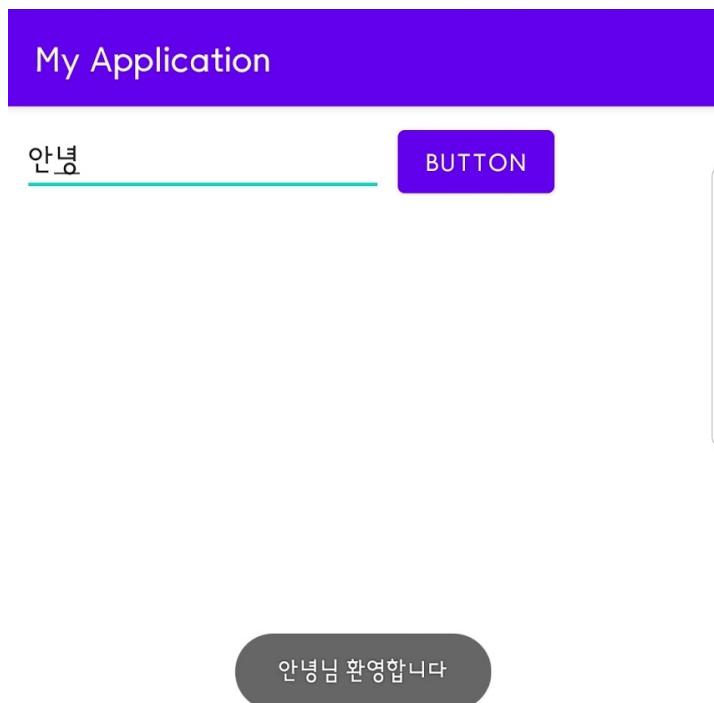
```

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

    Button button = findViewById(R.id.button);
    EditText editText = findViewById(R.id.editTextTextPersonName);
    // 버튼이 눌렸을 때 동작
    button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Toast.makeText(MainActivity.this,
                editText.getText().toString() + "님 환영합니다",
                Toast.LENGTH_SHORT).show();
        }
    });
}

```

5. 버튼을 누르면 입력한 텍스트를 불러와 토스트로 불러오는 코드로 수정하기



6. 완성

# 3

## Intent

안드로이드 화면전환에 대해 알아보자.

by. 박선균

# 3-1

# 인텐트란?

## 인텐트란?

안드로이드 스튜디오에서 화면과 화면사이를 넘어갈 때 쓰는 방법

### 사용방법

인텐트를 사용하려면 최소 2개의 xml 파일과 java 파일이 있어야 한다.

## 예제 어플리케이션 만들기

### activity\_main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".MainActivity">
8
9     <TextView
10         android:layout_width="wrap_content"
11         android:layout_height="wrap_content"
12         android:text="1번째 창입니다"
13         android:id="@+id/text1"
14         app:layout_constraintBottom_toBottomOf="parent"
15         app:layout_constraintLeft_toLeftOf="parent"
16         app:layout_constraintRight_toRightOf="parent"
17         app:layout_constraintTop_toTopOf="parent" />
18
19     <Button
20         android:id="@+id/button"
21         android:layout_width="102dp"
22         android:layout_height="50dp"
23         android:layout_marginTop="80dp"
24         android:text="Button"
25         app:layout_constraintBottom_toBottomOf="parent"
26         app:layout_constraintEnd_toEndOf="parent"
27         app:layout_constraintStart_toStartOf="parent"
28         app:layout_constraintTop_toBottomOf="@+id/text1" />
29
30 </androidx.constraintlayout.widget.ConstraintLayout>
```

## activity\_sub.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent">
8
9
10    <TextView
11        android:id="@+id/text2"
12        android:layout_width="272dp"
13        android:layout_height="141dp"
14        android:text="2번째 액티비티 친입니다."
15        app:layout_constraintBottom_toBottomOf="parent"
16        app:layout_constraintEnd_toEndOf="parent"
17        app:layout_constraintStart_toStartOf="parent"
18        app:layout_constraintTop_toTopOf="parent" />
19 </androidx.constraintlayout.widget.ConstraintLayout>
```

## MainActivity.java

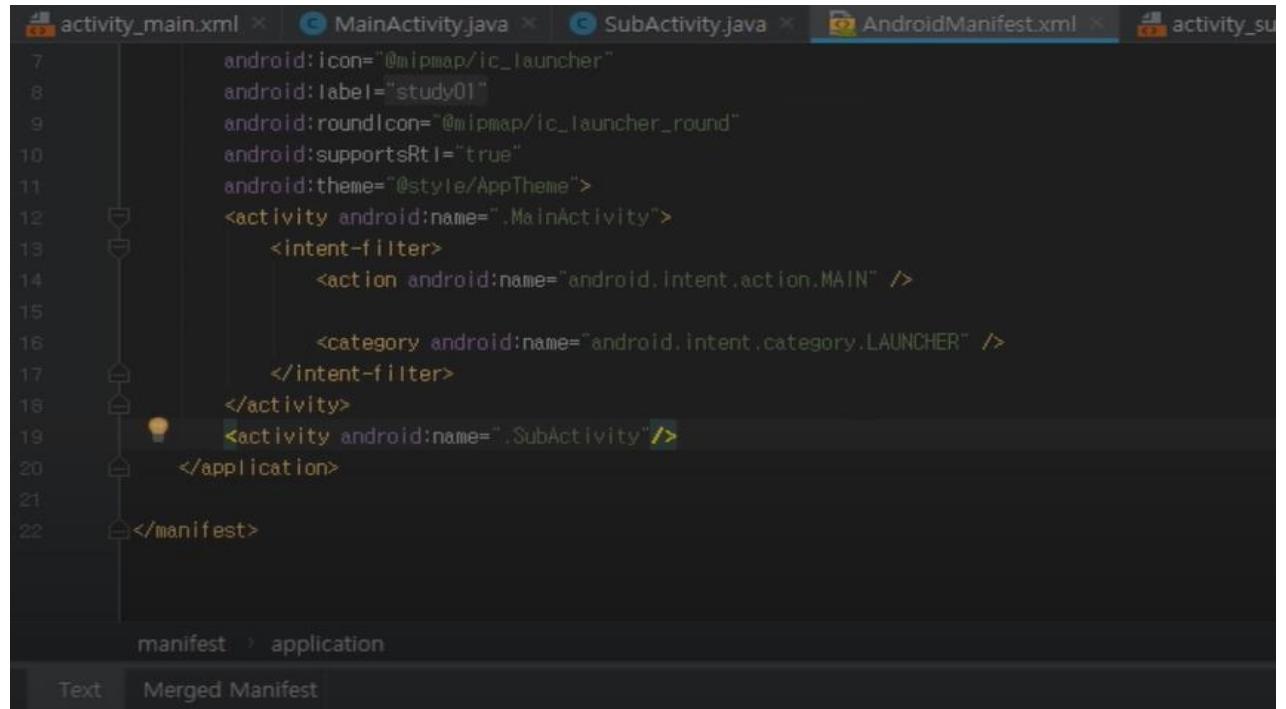
```
public class MainActivity extends AppCompatActivity {
    TextView text1;
    Button button;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        button = findViewById(R.id.button);
        text1 = findViewById(R.id.text1);

        button.setOnClickListener(v->{
            Intent intent = new Intent( mContext: MainActivity.this, SubActivity.class);
            startActivityForResult(intent);
        });
    }
}
```

13,14번에서 각각 선언을 해준다  
21번과 23번에서 저 함수를 불러온다  
24번 onclicklistener를 선언해 버튼을 클릭했을 시에 함수를 만들어준다  
27번에서 intent를 선언해 어디로 이동할지를 설정한다  
28번에서 intent를 시작한다 즉 화면을 이동한다

## AndroidManifest.xml



```
7     android:icon="@mipmap/ic_launcher"
8     android:label="study01"
9     android:roundIcon="@mipmap/ic_launcher_round"
10    android:supportsRtl="true"
11    android:theme="@style/AppTheme">
12    <activity android:name=".MainActivity">
13        <intent-filter>
14            <action android:name="android.intent.action.MAIN" />
15
16            <category android:name="android.intent.category.LAUNCHER" />
17        </intent-filter>
18    </activity>
19    <activity android:name=".SubActivity"/>
20
21 </application>
22
23 </manifest>
```

manifest > application

Text Merged Manifest

19번

새로운 액티비티를 만들어 줬으니 Manifest에서 activity를 추가해 줘야한다.

## 요약 : Intent를 사용하는 방법

1. Xml파일을 만든다.
2. Java파일을 만든다
3. Manifest에 가서 그 xml을 추가해준다

# 4

## View Components

Android Studio의 View와 Layout에 대해 알아봅시다.

by : 이종우

## View란?

어플리케이션에서 버튼, 이미지, 테이블, 레이블 등 눈에 보이는 모든 것을 말합니다. 이러한 뷰들 중에 최고 상위 클래스가 바로 View 클래스입니다. 이 View 클래스를 베이스로 두고 각각의 기능을 가지는 뷰들이 만들어집니다.

→ 클래스의 개념을 잘 모르는 경우 그냥 ‘View라는 것이 있고, 사용자에게 보여지는 것’ 정도로 이해하고 넘어가면 됩니다.

## Widget

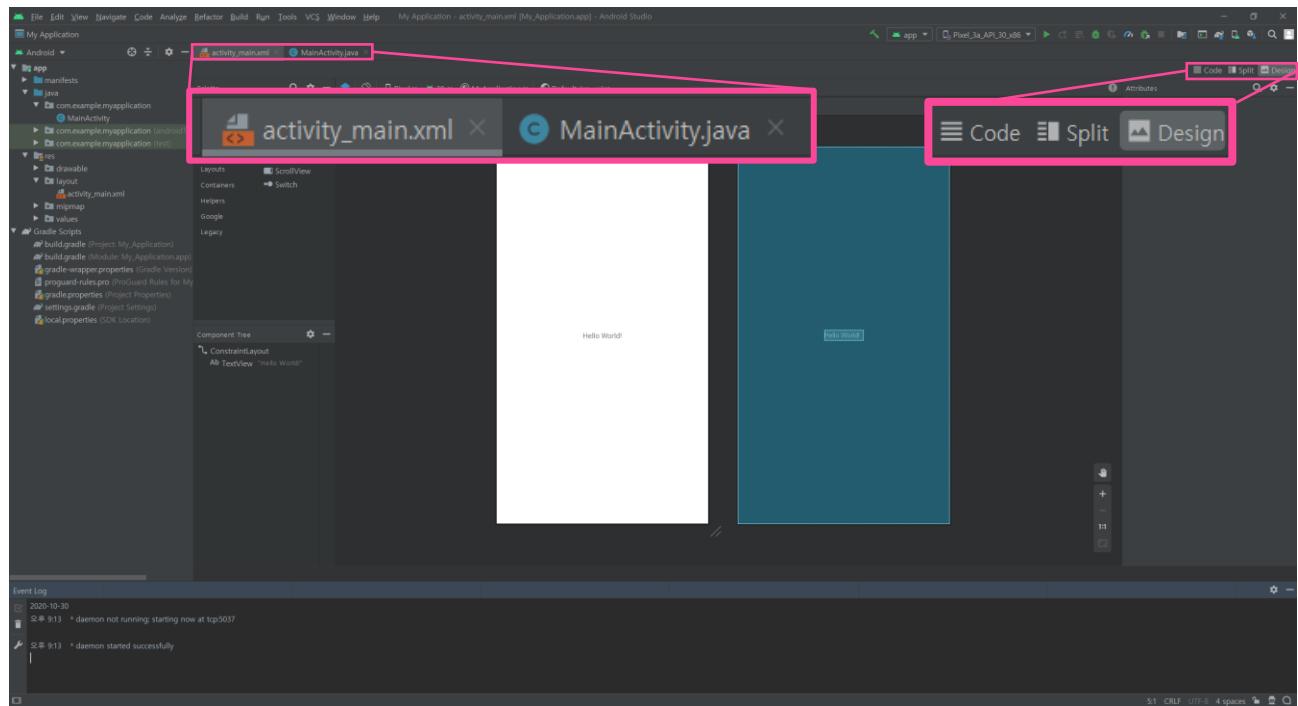
2단원(Activity)에서 다른 TextView, EditText, Button 등이 Widget입니다. View 또는 ViewGroup을 상속받아 사용자와 상호작용하는 요소입니다. ViewGroup은 View들을 담을 수 있는 요소들입니다.

→ ViewGroup도 다른 View와 마찬가지로 View를 상속받으므로, ViewGroup 안에 또 다른 ViewGroup을 넣을 수 있습니다.

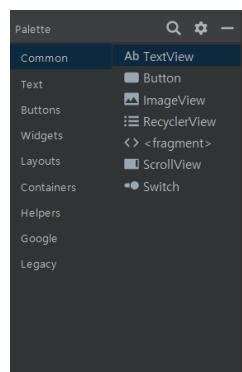
## Designer View

그렇다면, 이제 2단원(Activity)에서 간단히 다뤄 보았던 메인 화면에 대해 더 자세히 알아봅시다.

일반적으로, 안드로이드 스튜디오를 실행하고 새 프로젝트를 만들면 아래와 같은 화면이 나올 것입니다.



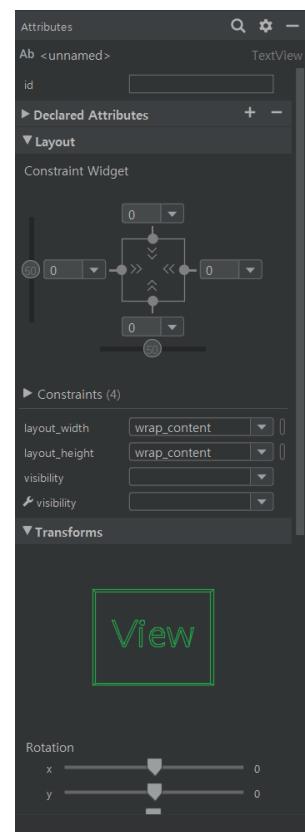
여기에서 왼쪽 상단을 보면, 위 사진처럼 activity\_main.xml과 MainActivity.java 파일이 열려 있는 것을 확인할 수 있습니다. 또한, 오른쪽 상단을 보면, Code / Split / Design 탭 중 Design 탭이 열려 있는 것을 확인할 수 있습니다.



팔레트(Palette)



컴포넌트 트리  
(Components Tree)



속성 탭  
(Attributes)

Design 탭을 네 부분으로 나누면 위 사진과 같습니다.

먼저, 팔레트(Palette)는 Widget의 목록을 띄워주는 역할을 합니다. 이곳에서 원하는 Widget을 찾아 실제 화면 부분으로 드래그하면 해당 Widget이 실제 화면에 들어가게 됩니다.

다음으로, 컴포넌트 트리(Component Tree)는 배치된 Widget들을 나타냅니다. 이 화면에는 어떤 Widget이 있는지, 또한 그 Widget들이 어떠한 포함 관계에 있는지 한 눈에 알아볼 수 있습니다.

그 다음, 속성 탭에서는 선택한 Widget의 속성을 표시해 줍니다. 이곳에서 Widget의 id, 글자, 가로/세로 길이 등의 속성을 변경할 수 있으며, Widget을 선택하지 않았을 때는 아무것도 표시되지 않습니다.

- 속성을 변경하는 방법은 두 가지로, Designer의 속성 탭에서 수정하거나, Code나 Split 탭에서 xml로 직접 타이핑하는 방법이 있습니다.
- 아직 Android Studio가 익숙하지 않다면 앞에서 설명한 부분을 본인의 컴퓨터에서 직접 다뤄 보시는 것을 추천합니다.

## View의 속성

앞에서 말했듯, View는 속성을 지정할 수 있습니다. 하지만 수많은 속성을 항상 정의(설정)해줄 수는 없겠죠.

### 가로, 세로

다만, 모든 View가 필수로 가지는 속성이 있습니다. 바로 가로, 세로 속성입니다.

가로 속성은 layout\_width, 세로 속성은 layout\_height로 나타냅니다.

그리고, 이 속성에 적용될 수 있는 값으로는 4가지가 있습니다.

1. wrap\_content: 뷰를 표시하는데 반드시 필요한 공간만 차지
2. match\_parent: 최상위 View 그룹의 가로/세로 크기만큼 차지
3. 예 직접 지정: 예 단위로 직접 지정
4. 0에: 레이아웃별로 작동 방식이 다름(레이아웃 부분에서 설명)

### Background

또한, 자주 사용하는 속성으로 background 속성이 있습니다. 이름에서도 알 수 있듯, 배경을 지정하는 속성으로, @drawable을 통해 icon과 image 등을 배경으로 지정하거나, #ARGB 값을 이용하여 16진수 값으로 배경색을 지정할 수 있습니다.

→ ARGB는 Alpha - Red - Green - Blue라는 뜻으로, 투명도 - 빨강 - 초록 - 파랑 순으로 색상을 지정하는 것입니다.

## **Id**

id 속성은 id(이름)를 지정하는 속성으로, @+id/{id}: 해당 View의 id를 설정하는 것입니다. 다른 곳(코드)에서 해당 View를 참조 시 필요합니다.

## **Visibility**

visibility 속성은 View가 보여질지를 결정하는 속성입니다. 3가지 속성이 있습니다.

먼저, visible은 일반적인 상태로, View가 보여집니다.

다음으로, invisible은 View가 보이지는 않지만, 빈 공간으로 자리는 차지하는 상태입니다.

마지막으로, gone은 View가 공간 자체를 차지하지 않고 사라지게 됩니다.

## 4-2 Layout

### Layout이란?

Layout의 정의는 무엇을 어디에 배치할지를 결정하거나 배치하는 것으로, Android Studio에서는 앞에서 살펴 봤던 View Components를 배치하는 역할을 하는 View Components 가 바로 Layout입니다.

우리가 뒤에서 살펴볼 LinearLayout, ConstraintLayout 이외에도 FrameLayout, RelativeLayout 등 다양한 종류의 Layout이 있지만, 가장 많이 사용하는 이 두 가지만 알고 있어도 충분히 어플리케이션을 제작할 수 있습니다.

- 본인이 모르는 Layout이 자동 생성되었을 경우, 본인이 알고 있는(사용하기 편한) Layout으로 변경해서 사용하면 됩니다.

### LinearLayout

LinearLayout은 다른 View를 가로 또는 세로로, 한 줄로 배열하는 Layout입니다. 아래 예시처럼 orientation 속성을 horizontal로 하거나 생략하면 가로로, vertical로 하면 가로로 배열되게 됩니다.

```
<LinearLayout  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content" />
```



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



```
<LinearLayout  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:orientation="vertical" />
```



- 이 코드에서는 Button에 대한 부분을 생략해 작성했지만, 실제로 코딩을 할 때에는 Button에 대한 코드를 <LinearLayout> … </LinearLayout>에 넣으면 해당 예시처럼 구현이 될 것입니다.

## layout\_gravity

layout\_gravity 속성은 안에 담긴 View를 정렬합니다.

left, right, center, top, bottom 중 선택 가능하며, 이 때 width& height가 match\_parent이면 화면을 꽉 채우게 되어 적용되지 않습니다.

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="vertical">  
  
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="left"  
    android:text="btn1" />  
  
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:text="btn2" />  
  
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="right"  
    android:text="btn3" />  
  
</LinearLayout>
```



- ➔ btn1, btn2, btn3를 소문자로 썼는데도 오른쪽 사진처럼 대문자로 나오는 것을 확인할 수가 있습니다. 대문자로 나오는 것을 원치 않는다면 android:textAllCaps = "false" 로 설정하면 됩니다.
- ➔ LinearLayout의 layout\_width가 wrap\_content로 설정되어 있으면 제대로 표시되지 않을 것입니다.

## layout\_weight

layout\_weight 속성은 안에 담긴 View들에게 남은 공간을 분할해 줍니다.

예를 들어, 세 개의 Button을 1:2:1로 분할해 주고 싶다면 아래와 같이 코드를 작성하면 됩니다.

이 경우, height를 0dp로 설정해 주는 것이 좋습니다. height를 0dp로 설정하더라도 실제 길이가 0dp가 되어 표시되지 않는 것이 아니라, weight 속성을 우선적으로 적용해 아래와 동일한 결과가 나오게 됩니다.

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="vertical">  
  
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_weight="1"  
    android:text="btn1" />  
  
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_weight="2"  
    android:text="btn2" />  
  
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_weight="1"  
    android:text="btn3" />  
  
</LinearLayout>
```

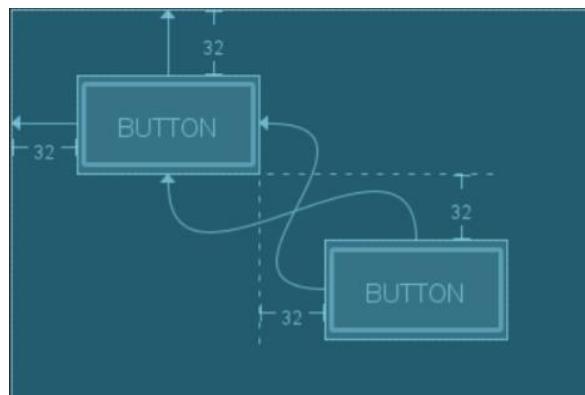
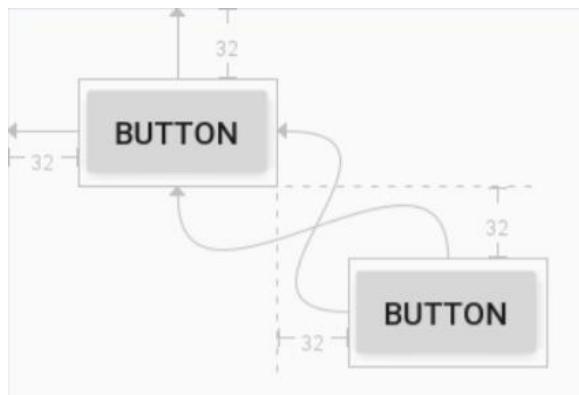


# ConstraintLayout

ConstraintLayout은 가장 어렵습니다. 그러나, LinearLayout을 포함해 기존의 모든 Layout을 대체할 수 있을 만큼 사용도가 높아 많은 개발자들이 사용하는 Layout입니다.

특징으로는 View와 View 사이의 상대적 관계를 바탕으로 배치하며, 따라서 다른 Layout보다 좀 더 유연하게 사용이 가능합니다.

백문이 불어일견. 아래 사진을 통해 확인해 볼시다.



이 책을 통해 더 설명하고 싶지만, 영상을 통해 설명을 듣는 것이 더 빠르기도 하고, 책을 통해 설명할 수 있는 부분이 한계가 있어 아래 영상 링크와 QR코드를 첨부합니다.

한 번에 이해가 되지 않을 수도 있는데, 나중에 프로그램을 만들다 보면 언젠가는 의숙해지게 될 것입니다.

<https://www.youtube.com/watch?v=tC-lHeOC7so&t=1377s>  
<https://www.youtube.com/watch?v=cOnF14szjZU>



# 4-3 View

이제, 어떤 View 요소가 있는지 알아봅시다.

## TextView

TextView는 텍스트를 출력할 때 사용하는 View입니다.

### TextView의 사용 방법

#### XML

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Hello World!"  
    android:textSize="20dp"  
    android:textColor="@color/colorPrimaryDark"  
/>
```



#### Java

```
TextView textView = findViewById(R.id.textView);  
textView.setText("Text is changed.");
```

### TextView의 속성

- 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

# EditText

EditText는 사용자에게 값을 입력받을 때 사용하는 View로, TextView를 상속해 TextView의 속성을 같이 사용 가능합니다.

## EditText의 사용 방법

### XML

```
<EditText  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:inputType="textPersonName"  
    android:text="Name"  
    />
```



Name

### Java

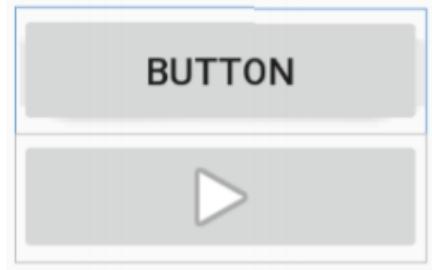
```
EditText editText1 = (EditText)  
        findViewById(R.id.editText1);  
editText1.setText("EditText 내용이 바뀌었습니다. 이걸로");
```

## EditText의 속성

- inputType : EditText에 입력되는 데이터 종류를 결정(여러 줄 입력, 숫자만 입력 등)
- lines : 처음부터 지정된 줄만큼 공간을 차지하도록 함
- maxLines : 최대 사용할 수 있는 줄 수

# Button & ImageButton

사용자가 누르는 것입니다.



## Button & ImageButton의 사용 방법 XML

```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Button"  
    android:onclick="newName"  
    android:drawableLeft="@drawable/ic_launcher_background"/>  
<ImageButton  
    android:id="@+id/imageButton"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:onclick="setName"  
    android:src="@drawable/ic_launcher_background"/>
```

onClick을 통해 이 버튼을 클릭했을 때 어떤 메소드가 호출될지 지정할 수 있습니다.

## onClick

Java(≤ JDK 1.7)

```
Button button1 = findViewById(R.id.button1);  
button1.setOnClickListener(new Button.OnClickListener()  
{  
    @Override  
    public void onClick(View view) {  
        // 클릭했을 때 이벤트  
    }  
});
```

Java(≥ JDK 1.8)

```
button1.setOnClickListener(v -> {  
    // 클릭했을 때 이벤트  
});
```

# ImageView

화면에 Image를 표시해줍니다.



## ImageView의 사용 방법

XML

```
<ImageView android:layout_width="wrap_content"  
          android:layout_height="wrap_content"  
          android:src="@drawable/android_icon"  
          android:scaleType="center"  
          android:tint="#ffffffff"/>
```

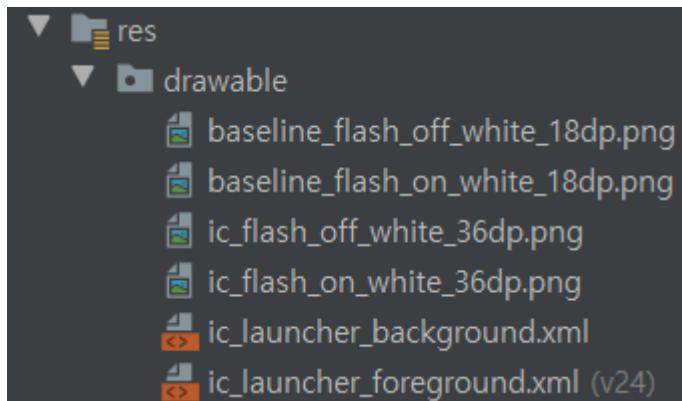
Java

```
ImageView imageView1 = findViewById(R.id.imageView1) ;  
imageView1.setImageResource(R.drawable.myimage) ;
```

## ImageView의 속성

- src : 원본 이미지 설정
- tint : 이미지 위에 색상 필터 적용
- scaleType : 이미지가 원본 이미지와 다르게 보이는 경우 확대 / 축소를 어떤 방식으로 적용할 것인지 설정

# Drawable



jpg/png 등의 이미지 파일을 넣는 폴더입니다.

폴더에 이미지 파일을 복사해 넣으면 리소스 생성 확장자를 제외한 파일명이 R.drawable.파일명 id로 지정됩니다.

또한, Vector 그래픽을 생성할 수도 있습니다.

# CompoundButton

CompoundButton을 상속받는 View Components입니다.

## CheckBox

사용자가 특정 항목을 체크하거나, 체크 해제할 수 있게 합니다.

```
<CheckBox  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="checkbox"/>
```



## Switch

사용자가 특정 옵션을 켜거나 끌 수 있게 합니다.

```
<Switch  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:checked="true"  
    android:text="switch"/>
```



## RadioButton & RadioGroup

RadioGroup과 RadioButton은 사용자가 여러 아이템 중 하나를 고를 수 있습니다.

RadioButton은 같은 RadioGroup 안에 있어야 하나만 선택할 수 있습니다.

```
<RadioGroup  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginStart="32dp"  
    android:layout_marginTop="32dp">  
  
<RadioButton  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="select" />  
  
<RadioButton  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="select" />  
</RadioGroup>
```



## RadioGroup의 사용 방법

```
RadioGroup radioGroup = findViewById(R.id.radioGroup);
radioGroup.setOnCheckedChangeListener(new
RadioGroup.OnCheckedChangeListener() {
@Override
public void onCheckedChanged(RadioGroup radioGroup, int checkedItemId)
{}});
```

RadioGroup.OnCheckedChangeListener는 체크 상태가 변경되면 체크 된 RadioButton의 id를 반환합니다. 이를 통해 어떤 아이템이 체크되었는지 확인할 수 있습니다.

동일하게, CheckBox와 Switch 또한 OnCheckedChangeListener를 갖고 있습니다. 다만, 아이템 id 대신 체크 여부, 즉 체크가 되었는지 되지 않았는지를 Boolean으로 전달합니다.

각 요소별로 isChecked()와 setChecked() 메서드를 통해 체크 상태를 가져오거나 설정할 수 있습니다.

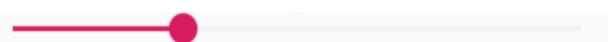
## SeekBar

SeekBar는 슬라이더 형태의 게이지 바를 뜻합니다.

### SeekBar의 사용 방법

#### XML

```
<SeekBar
    android:id="@+id/seekBar"
    android:max="100"
    android:progress="30"/>
```



### SeekBar의 속성

- style : SeekBar의 스타일을 지정할 수 있습니다. 연속적인 값을 위해서는, style 지정 없이 사용하면 되고, 이산적인 값을 위해서는 "@style/Widget.AppCompat.SeekBar.Discrete" 스타일을 지정해야 합니다.
- android:max : "android:progress"의 최대값을 지정 가능합니다.
- android:progress : SeekBar의 thumb 위치를 지정합니다.

## onSeekBarChangeListener

onSeekBarChangeListener는 SeekBar의 위치가 변경될 때 호출되는 Listener입니다.

onProgressChanged(), onStart/StopTrackingTouch()을 implement 해야 합니다.

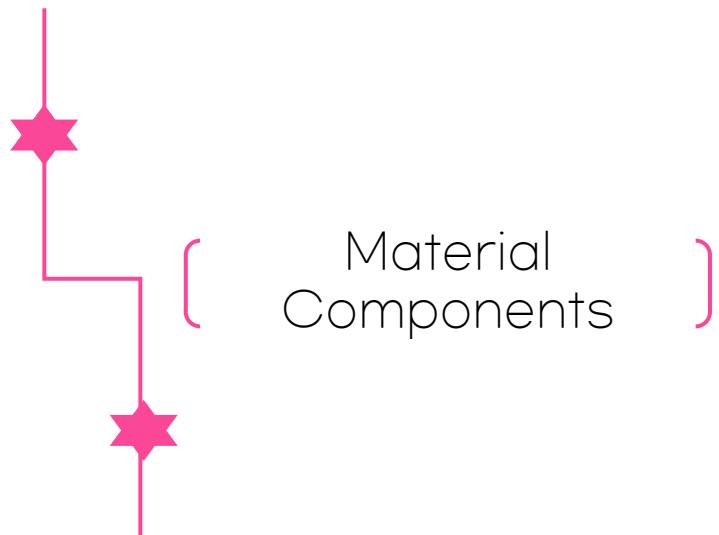
### Java

```
SeekBar seekBar = findViewById(R.id.seekBar);
seekBar.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {
@Override
public void onProgressChanged(SeekBar seekBar, int i, boolean
b) {
}
@Override
public void onStartTrackingTouch(SeekBar seekBar) {
}
@Override
public void onStopTrackingTouch(SeekBar seekBar) {
}
});
```

# 5

## Material Components

[ Material Design ]



Material  
Components

[ Android Completion ]

by : 채예린

# 5-1

# Material Design

## Material Design 이란?

안드로이드 개발을 위해 구글이 만든 앱디자인 가이드라인 입니다.

## Google Material Design 요소

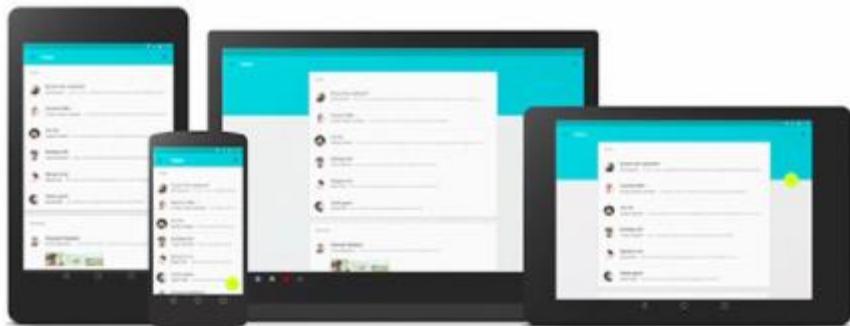
**애니메이션**: 애니메이션의 속도와 방향

**색상**: 어떠한 테마색상으로 구현해야 하는지

**아이콘/이미지**: 아이콘을 어떻게 만드는지

**폰트**: 언어 타입별로 굵기와 크기를 지정

**문구작성**: 입력폼을 작성할때 문구를 어떻게 해야하는지



Material Design 주소 : [material.io](http://material.io)

# 5-2 Material Components

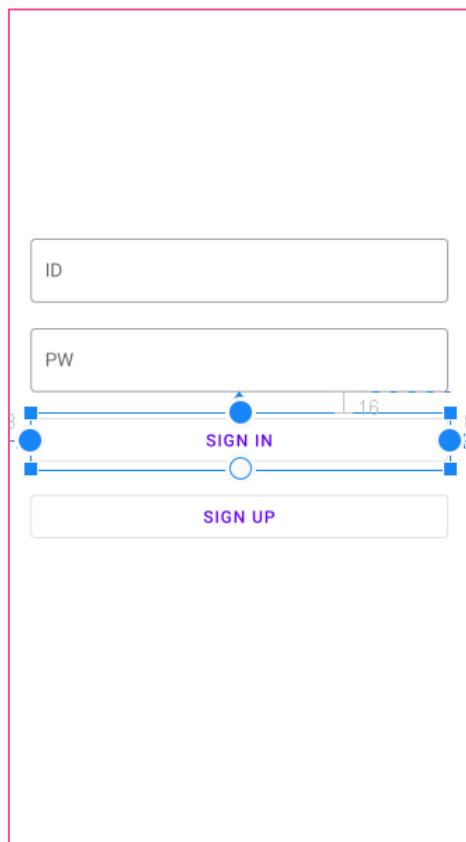
## Material Components 란?

사용자 인터페이스를 생성하기 위한 대화형 구성 요소다.

예) 상단 바, 하단 바, 배경 등

## Material Components 구현하기 !

Material Components 중에 가장 둘보이면서 구현하기 쉬운 Text fileds를 만들어 보자.



### Text fileds 란 ?

텍스트를 입력하고 편집할수 있는 공간이다.

Material Components를 사용하여 간단히 회원가입 창 구현하였다.

### 코드보러가기 :

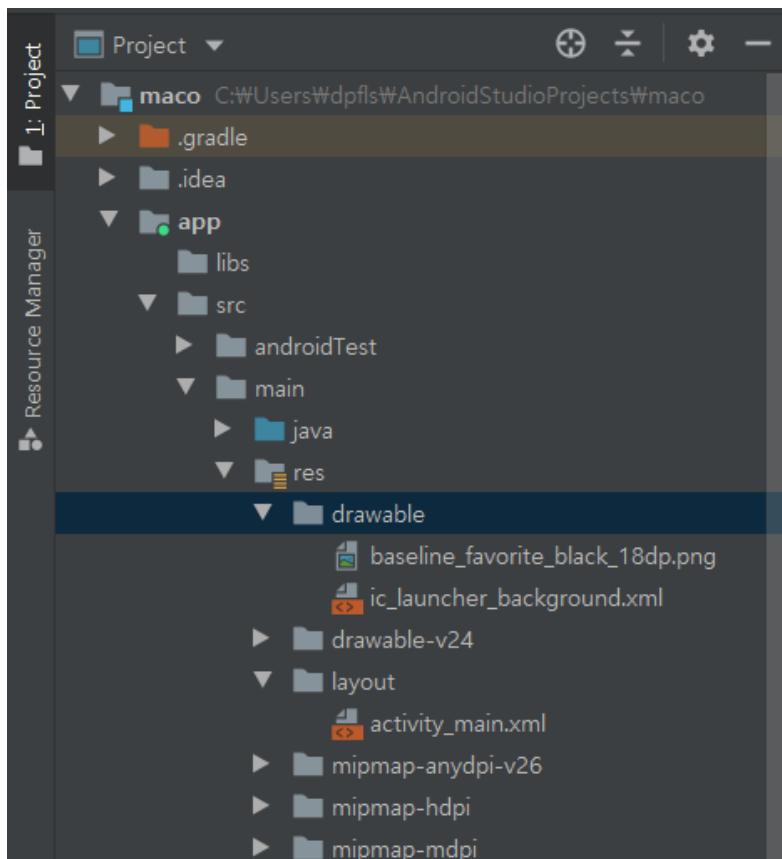
<https://dpflspowe040223.tistory.com/3>

비밀번호 : 0223

# 아이콘을 넣어보자 !

여러가지 아이콘 다운받기 주소 :

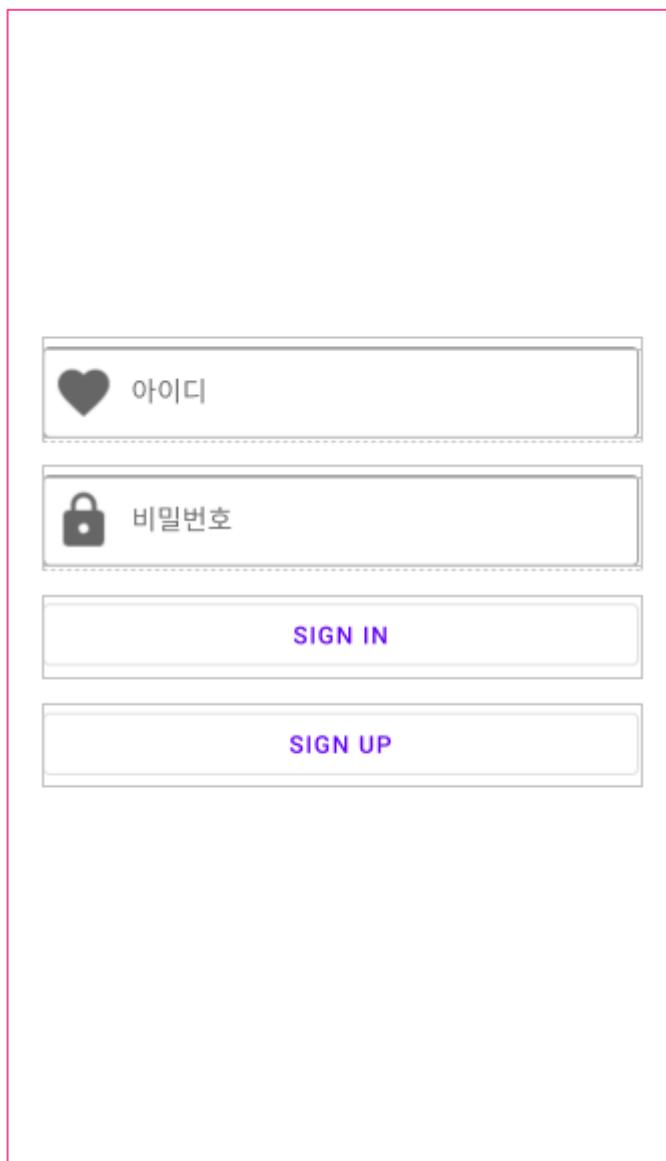
<https://material.io/resources/icons/?icon=favorite&style=baseline>



자신이 마음에 드는 아이콘을 다운로드후 drawable에 넣어준다.

Main.xml->TextInputLayout에  
app:startIconDrawable="@drawable/사진의 이름" 을 추가한다.

넣어주면 각 Text filed영역에 아이콘이 생긴다



## 비밀번호 가리기! 구현해보자



눈 모양으로 된 아이콘을 클릭하면  
비밀번호가 암호화 된다.

# 어떻게 해야할까?

```
<com.google.android.material.textfield.TextInputLayout  
    android:id="@+id/TextInput_PW"  
    style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"  
    android:layout_width="350dp"  
    android:layout_height="wrap_content"  
    android:layout_marginStart="8dp"  
    android:layout_marginTop="16dp"  
    android:layout_marginEnd="8dp"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/TextInput_ID"  
    app:startIconDrawable="@drawable/baseline_lock_black_18dp"  
    app:endIconMode="password_toggle">
```

간단하게 맨 밑줄에 있는 코드를 써서 넣어주면 된다.

startIcon : 실행  
endIcon : 후행

Toggle은 켜고 끌 수 있도록 하게 해주는 기능이다.

## 이번에는 삭제기능을 구현해보도록 하자!

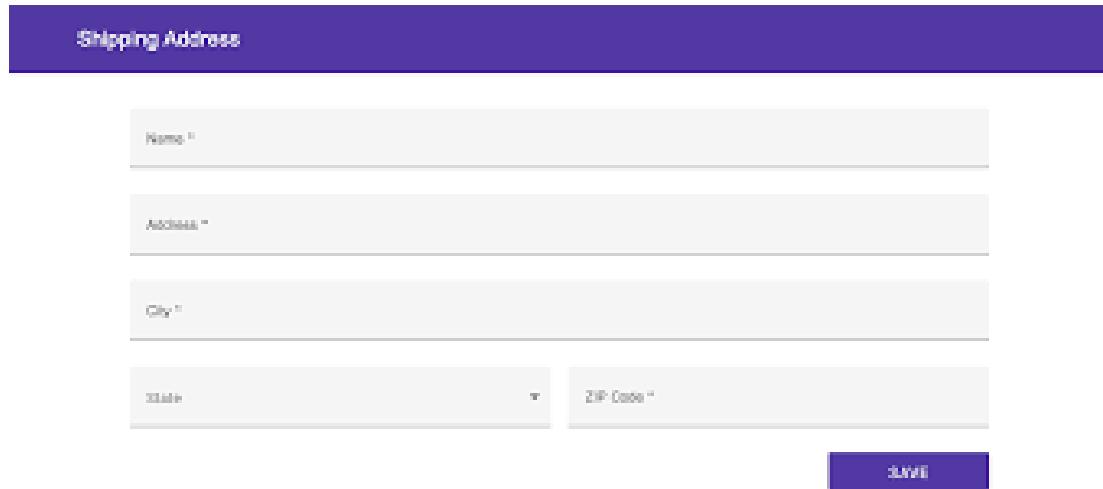


X버튼의 아이콘을 누르면 삭제가 된다.

# 어떻게 해야할까?

```
<com.google.android.material.textfield.TextInputLayout  
    android:id="@+id/TextInput_ID"  
    style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"  
    android:layout_width="350dp"  
    android:layout_height="wrap_content"  
    android:layout_marginStart="8dp"  
    android:layout_marginEnd="8dp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.31"  
    app:startIconDrawable="@drawable/baseline_favorite_black_18dp"  
    app:endIconMode="clear_text">
```

Toggle과 같이 맨 밑줄에 똑같이 코드를 써서 넣어주면 된다.



# 6

## SharedPreferences

간단한 데이터를 SharedPreferences를 통해  
전송해보자.

by. 이종우, 최경진

## 데이터 전달

일반적으로, Java에서 한 클래스에서 다른 클래스로 데이터를 전달하는 것이 가능한 하지만, 복잡하기도 하고, 여러 가지 이유로 안드로이드 어플리케이션을 개발할 때는 잘 사용하지 않습니다.

그렇지만, 우리는 어플리케이션을 사용하면서 한 화면에서 다른 화면으로 넘어갈 때, 값이 전달되는 것을 볼 수 있을 것입니다. 예를 들어, 통장 잔고를 확인하는 어플리케이션에서, 통장의 잔액이 메인 화면과 상세 조회 화면 모두에서 동일하게 표시됩니다.

그렇다면, 이를 어떻게 구현했을까요?

Android Studio에서 이를 구현하는 방법에는 크게 2가지가 있습니다. 하나는 인터넷 연결을 통해 Firebase(챕터 10 참고)와 같은 Database를 사용하는 것이고, 다른 하나는 SharedPreferences를 사용하는 것입니다.

## SharedPreferences

SharedPreferences는 인터넷 연결 없이 각종 정보(숫자, 문자 등)를 자바 파일 간, 혹은 화면 간 데이터 전송에 쉽게 사용할 수 있습니다.

이번 챕터에서는 SharedPreferences를 사용해 회원가입, 로그인 창을 구현해 보도록 하겠습니다.

그 전에, 4-2챕터에서 만들었던 xml 파일을 준비해 주세요.

# 6-2 데이터 저장 / 불러오기

## SharedPreference의 사용법

SharedPreference의 사용법에 대해 살펴봅시다.

간단해서 금방 살펴볼 수 있을 것입니다.

```
SharedPreferences pref = mContext.getSharedPreferences("파일 이름", Activity.MODE_PRIVATE);
SharedPreferences.Editor editor = pref.edit();
editor.putString("키 값", "저장할 String값");
editor.commit();
```

## 데이터 저장

SharedPreference에 데이터를 저장하기 위해, 기본적으로 입력해야 할 값은 4가지가 있습니다.

1) 1번째 줄의 "파일 이름" => SharedPreferences가 저장되는 파일의 이름이라고 생각하면 됩니다.

2) 1번째 줄의 Activity.MODE\_PRIVATE => 생성된 SharedPreferences 파일의 접근 권한 설정이라 생각하면 됩니다.  
다른 앱에서 접근 가능하게 설정 가능하지만, 보통은 해당 앱에서만 접근 가능한 MODE\_PRIVATE를 많이 씁니다.

3) 3번째 줄의 " 키 값 " => 저장하는 값을 나타내는 키 값을 설정합니다.

4) 3번째 줄의 "저장할 String값" => .putString() 메소드이므로 저장할 String 값을 입력합니다.

➔ 반드시 마지막 부분에 editor.apply() 혹은 editor.commit()을 작성해 주어야 합니다. apply와 commit은 동기 / 비동기의 차이로, 둘 중 어느 것을 사용해도 일반적인 경우 어플리케이션 실행 및 구동에 영향을 미치지 않습니다.

# 데이터 불러오기

```
SharedPreferences pref = mContext.getSharedPreferences("파일 이름",  
Activity.MODE_PRIVATE);  
String s = pref.getString("키 값", "기본값");
```

입력 값의 대부분은 전 페이지를 참고하면 되고, “기본 값”은 getString()에서 가져올 값이 없을 때 주어질 기본 값입니다.  
즉, 가져올 값이 없을 경우, s에는 “기본 값”이 들어가게 됩니다.

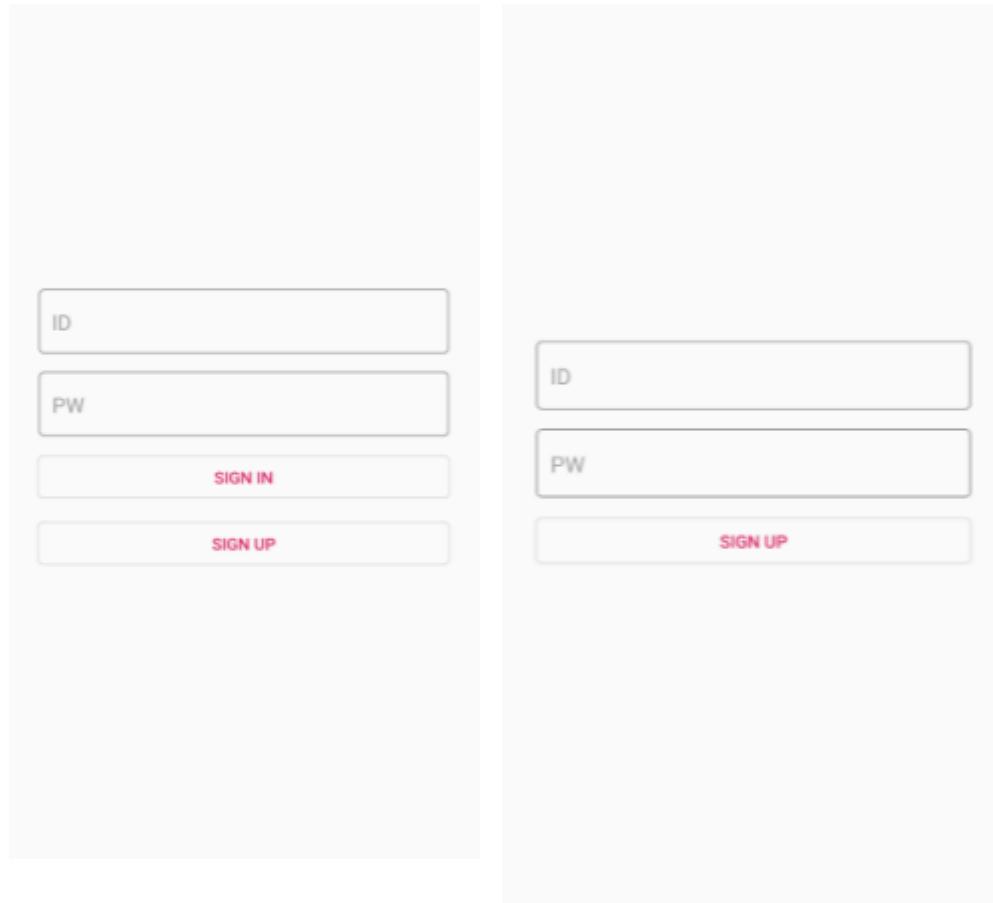
# 데이터 삭제하기

```
SharedPreferences pref = mContext.getSharedPreferences("파일 이름", Activity.MODE_PRIVATE);  
SharedPreferences.Editor editor = pref.edit();  
editor.remove("키 값");  
editor.commit();
```

→ 만약 기존 데이터의 형을 변경시키고 싶다면 반드시 기존 데이터를 삭제 후 다시 변경사항을 적용해야 합니다. 그렇지 않으면 오류가 발생할 수 있습니다.

## 6-3 회원가입, 로그인 만들기

### 회원가입, 로그인 어플리케이션 만들기



위와 같이 회원가입 창에서는 ID가 중복되지 않도록 회원가입을 하고, SharedPreference에 해당 데이터를 기록하도록 합니다.  
로그인 창에서는 SharedPreference에서 정보를 받아 와서 ID와 PW가 맞으면 로그인에 성공했다는 정보를 Toast 메시지로 띄워 주면 됩니다.

정답 소스코드는 ‘부록’에 첨부했으니, 본인이 할 수 있는 데까지 해 보고 정답 소스 코드를 보시는 것을 추천합니다.

# 7

## RecyclerView

Android Studio의 RecyclerView에 대해 알아봅시다.

by : 박찬웅

7-1

# RecyclerView란?

## RecyclerView란?

RecyclerView는 ListView의 향상되고 유연한 버전으로 ListView와 달리 Item을 표시하는 각각의 View를 효율적으로 재사용할 수 있도록 구성되어 있습니다.

→ ListView 호환 상위 위젯으로 받아들여도 무방할 것 같습니다.

## ViewHolder란?

RecyclerView는 ViewHolder를 통해 각 Item을 표시할 View를 관리하도록 하는 구조로 설계되어 있습니다.

ViewHolder는 RecyclerView.ViewHolder와 여분의 ViewHolder를 기본으로 생성하여 스크롤에 대비하며, 스크롤등의 동작으로 인해 사라지는 아이템을 표시하고 있는 ViewHolder를 새로 표시되는 아이템을 그리는데 재사용하여 대용량의 데이터라도 소수의 View를 이용하여 표현할 수 있습니다.

→ ViewHolder는 RecyclerView에서 Item 표시, View 관리를 맡고 있으며 스크롤을 내릴 때를 대비하여 여분의 ViewHolder를 생성하고 스크롤하면서 사라지는 아이템들을 새로 표시되는 아이템을 그리는데 재사용이 됩니다.

## Adapter란?

ViewHolder 객체는 RecyclerView Adapter에 의해 생성 및 관리 되며, Adapter는 RecyclerView.Adapter를 상속받아 작성하게 됩니다.

Adapter는 ViewHolder를 위치 기반으로 할당하고, Data를 ViewHolder에 반영할 수 있도록 onBindViewHolder() 콜백을 호출하게 됩니다.

→ onBindViewHolder 콜백 호출을 받았을 때, 위치 기반의 데이터를 할당된 ViewHolder에 표시할 수 있습니다.

## 7-2 RecyclerView 사용

### RecyclerView

#### RecyclerView의 사용 방법



Match\_parent  
id:recyclerView

### ArrayList의 사용방법

#### 기본형

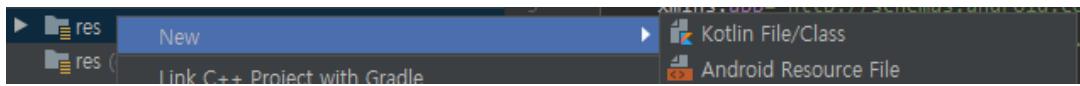
ArrayList<자료형> 변수명 = new ArrayList<>();

#### Java

```
ArrayList<String> data = new ArrayList<>();
data.add("정해성");
data.add("이종우");
data.add("김우진");
```

# item 만들기

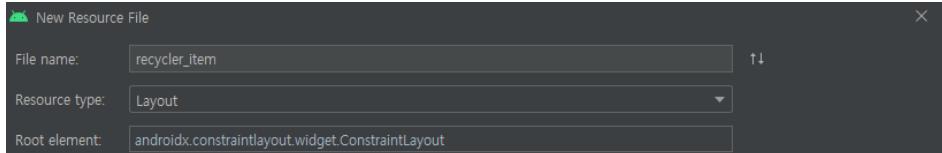
res > New > Android Resource File



File name: recycler\_item

Resource type: Layout

Root element: ConstraintLayout



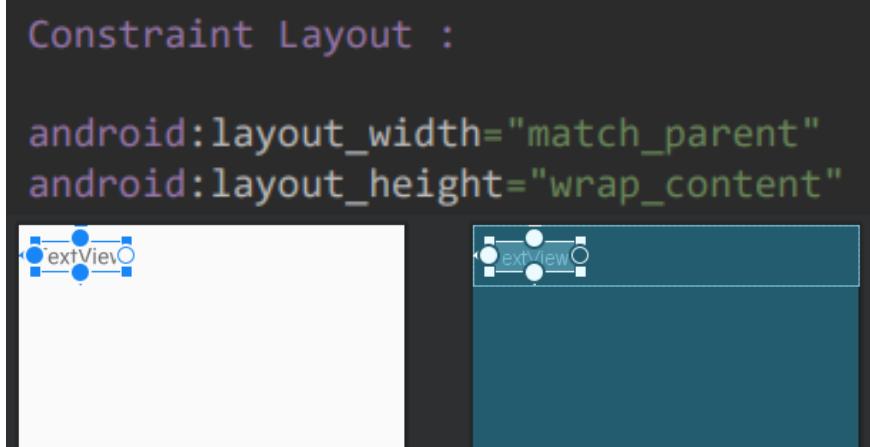
## XML

id: textView

textSize: 24sp

Anchor: top, left, bottom

Margin: 16dp

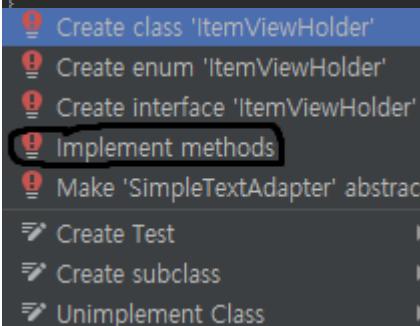


## SimpleTextAdapter.java

```
package com.java.myapplication;

import androidx.recyclerview.widget.RecyclerView;

public class SimpleTextAdapter extends RecyclerView.Adapter<SimpleTextAdapter.ItemViewHolder>{
```



alt+Enter+Implement methods

```
class ItemViewHolder extends RecyclerView.ViewHolder{  
}  
  
class ItemViewHolder extends RecyclerView.ViewHolder{  
  
    public ItemViewHolder(@NonNull View itemView) {  
        super(itemView);  
    }  
}
```

alt+Enter 이전

alt+Enter 이후

```
TextView textView;  
public ItemViewHolder(@NonNull View itemView) {  
    super(itemView);  
    textView = itemView.findViewById(R.id.textView);  
}  
  
public SimpleTextAdapter.ItemViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
    LayoutInflater inflater = LayoutInflater.from(parent.getContext());  
    View view = inflater.inflate(R.layout.recycler_item, parent, attachToRoot: false);  
    return new SimpleTextAdapter.ItemViewHolder(view);  
}  
  
private ArrayList<String> mData = null;  
SimpleTextAdapter(ArrayList<String> list){  
    mData = list;  
}
```

1. 데이터들을 저장할 ArrayList 생성
2. 생산자에서 데이터를 전달받음
3. RecyclerView에 추가할 아이템을 등록

## 나머지 두 함수도 채워주기

```
@Override  
public void onBindViewHolder(@NonNull SimpleTextAdapter.ItemViewHolder holder, int position) {  
    String text = mData.get(position);  
    holder.textView.setText(text);  
}  
  
@Override  
public int getItemCount() {  
    return mData.size();  
}
```

## MainActivity.java

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    ArrayList<String> data = new ArrayList<>();  
    data.add("정해성");  
    data.add("이종우");  
    data.add("김우진");  
  
    RecyclerView recyclerView = findViewById(R.id.recyclerView);  
    recyclerView.setLayoutManager(new LinearLayoutManager(this)); //  
    SimpleTextAdapter simpleTextAdapter = new SimpleTextAdapter(data);  
    recyclerView.setAdapter(simpleTextAdapter);  
}  
recyclerView.setLayoutManager(new LinearLayoutManager(this, LinearLayoutManager.HORIZONTAL, false));
```

## 실행화면



# 8

## fragment

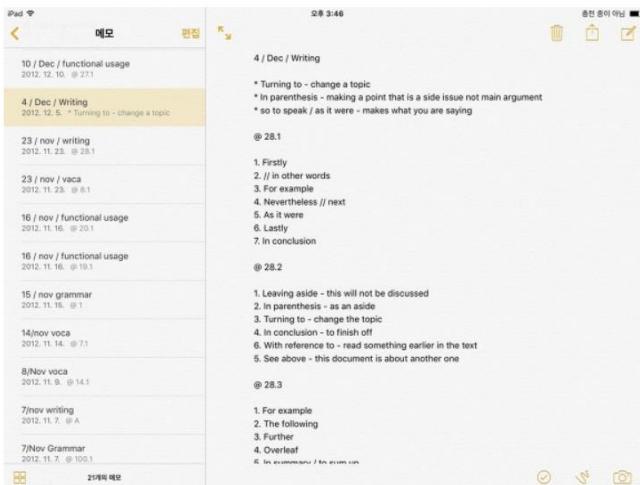
여러 개의 화면생성, fragment에 대해 배워봅시다.

by : 백영대, 채예린



## 프래그먼트 (fragment)란?

프래그먼트 (fragment)는 하나의 액티비티가 여러 개의 화면을 가지도록 만들기 위해 고안된 개념입니다. 다양한 크기의 화면을 가진 모바일 환경이 늘어남에 따라 하나의 디스플레이 안에서 다양한 화면 요소들을 보여주는 방법입니다. 대표적인 예시로 태블릿을 들 수 있습니다.



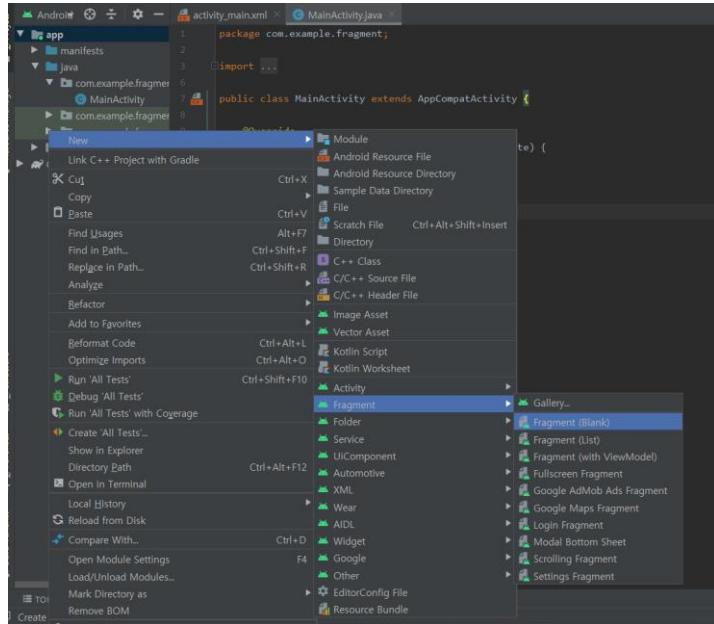
사진과 같이, 프래그먼트를 사용하면 화면을 분할하여 사용할 수 있는 장점이 있습니다.

## 프래그먼트 (fragment)의 특징

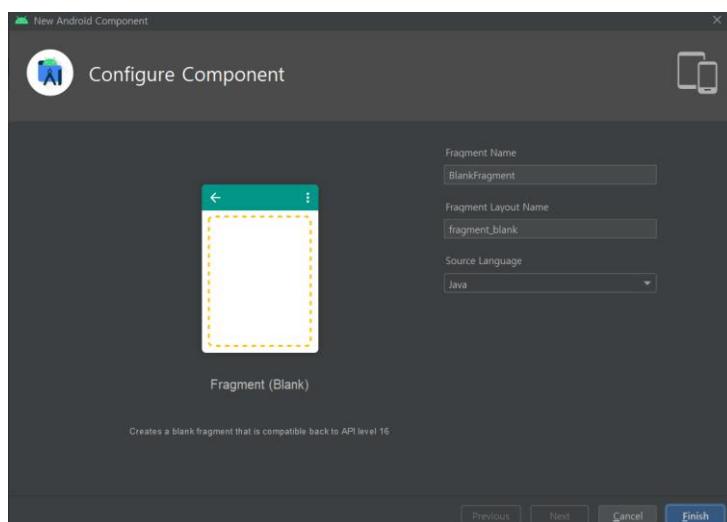
- 액티비티를 분할하여 화면의 한 부분을 정의한다.
- 액티비티와 같이 레이아웃, 동작 처리, 생명주기를 가지는 독립적인 모듈이다.
- 다른 액티비티에서도 사용 할 수 있어 재사용성이 뛰어나다.
- 액티비티 내에서 실행 중에 추가, 제거가 가능하다.

## 프래그먼트 생성

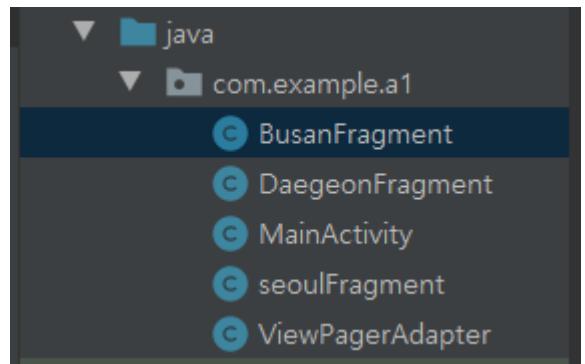
[프로젝트 패키지에서 우클릭] -> [New] -> [Fragment] 를 클릭하여 프래그먼트 마법사를 띠웁니다.



Finish 클릭 후 생성하기.(2개 생성!!)



각 main 과 viewPagerAdater.java를 제외한 것들은  
전에 장을 똑같이 따라 하면 코드는 자동생성이 됩니다  
건드릴 필요 없습니다.



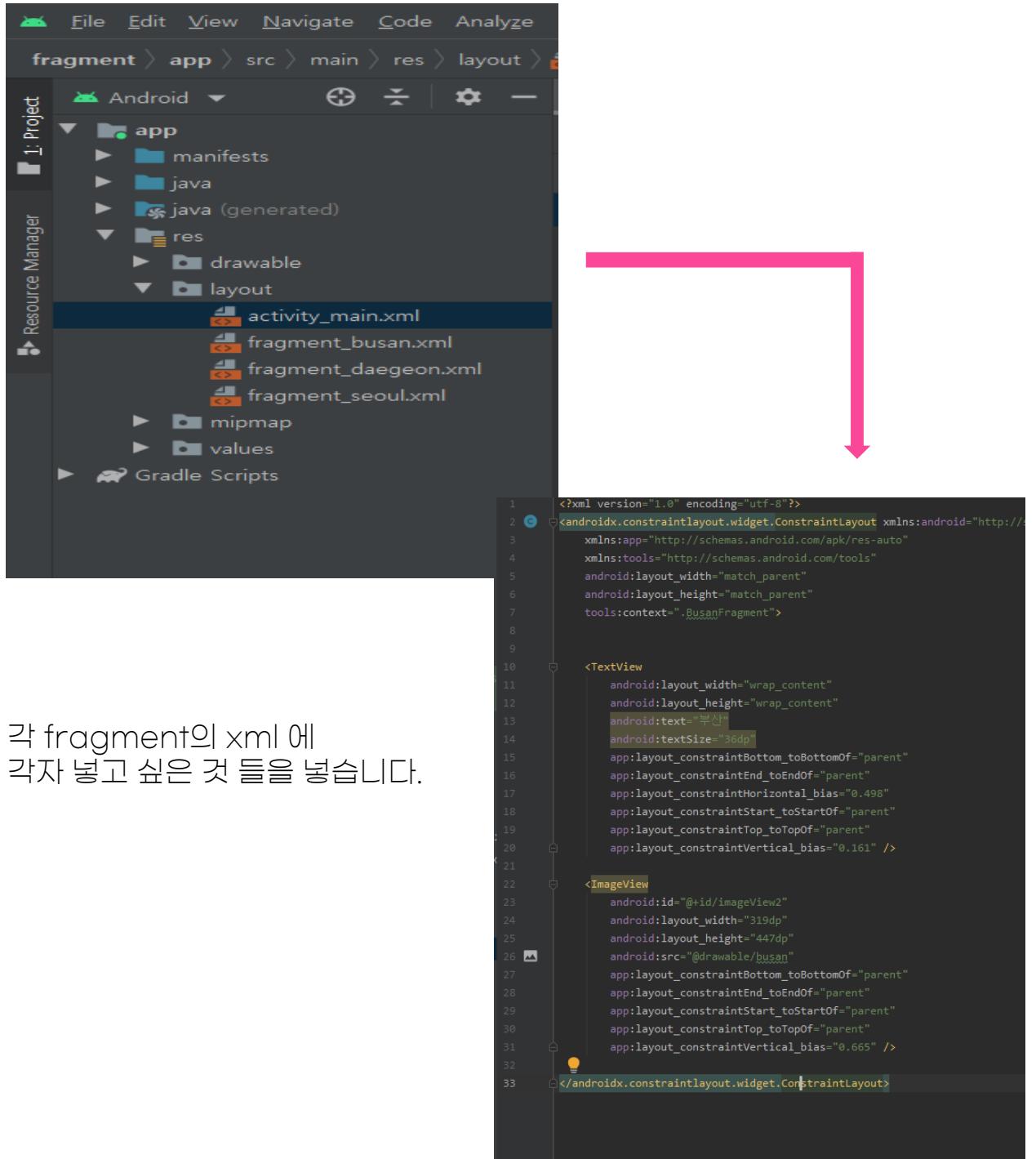
ViewPagerAdapter가 이제 page를 바꾸는 기능을 넣어주는 java 인데

```
1  package com.example.a1;
2
3  import ...
4
5
6  public class ViewPagerAdapter extends FragmentPagerAdapter {
7      public ViewPagerAdapter(@NotNull FragmentManager fm, int i) { super(fm); }
8
9
10     @Override
11     public Fragment getItem(int position) {
12         Fragment returnValue = null;
13         switch(position){
14             case 0 :
15                 returnValue = new seoulFragment();
16                 break;
17             case 1 :
18                 returnValue = new DaegeonFragment();
19                 break;
20             case 2 :
21                 returnValue = new BusanFragment();
22                 break;
23         }
24         return returnValue;
25     }
26
27     @Override
28     public int getCount() { return 3; }
29
30
31     @Override
32     public CharSequence getPageTitle(int position){
33         switch(position){
34             case 0:
35                 return "Seoul";
36             case 1:
37                 return "Daegu";
38             case 2:
39                 return "Busan";
40         }
41         return super.getPageTitle(position);
42     }
43
44
45
46
47
48
49 }
```

Switch ~case 문을 사용하여서 page를 바꾸어 주었습니다.

# 8-3 프래그먼트 만들기

먼저 각 fragment의 xml 을 만들어 줍니다

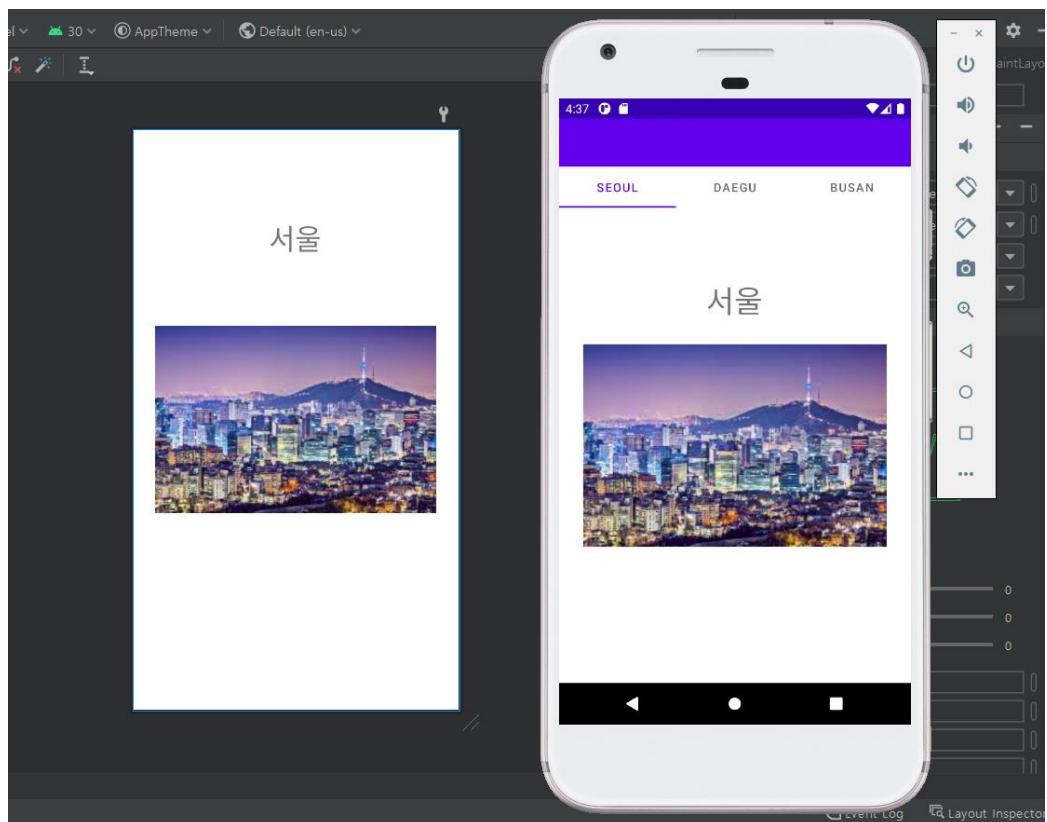


각 fragment의 xml 에  
각자 넣고 싶은 것 들을 넣습니다.

하지만 ViewPagerAdapter가 작동하기 위해선 MainActivity 가 일을 해야하는데

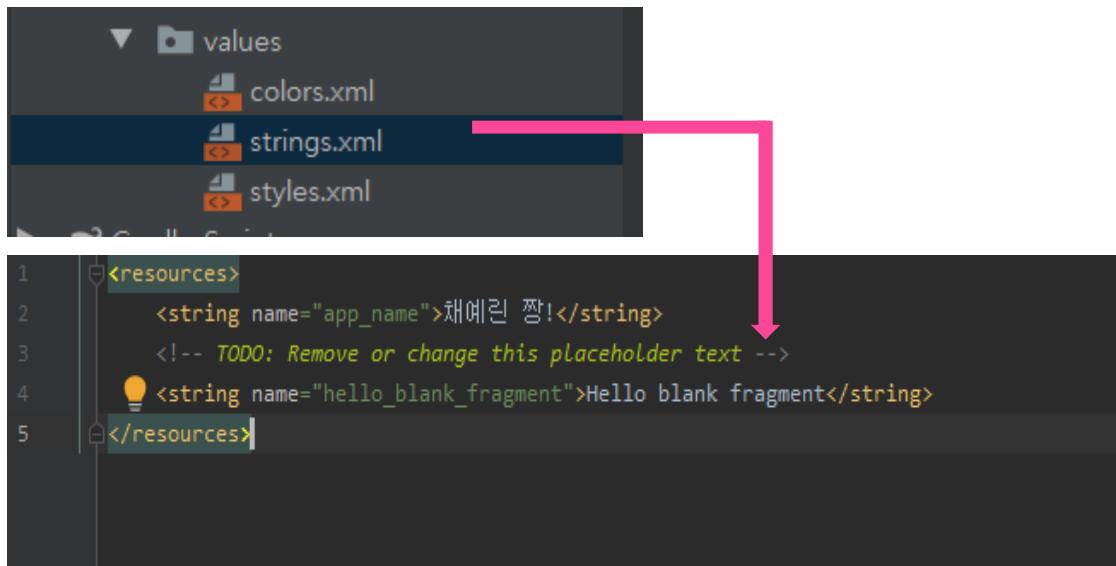
```
1 package com.example.a1;
2
3 import ...
4
5
6 public class MainActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10        super.onCreate(savedInstanceState);
11        setContentView(R.layout.activity_main);
12        ViewPager viewPager = findViewById(R.id.viewPager);
13        viewPager.setAdapter(new ViewPagerAdapter(getSupportFragmentManager());
14        TabLayout tabLayout = findViewById(R.id.tabLayout);
15        tabLayout.setupWithViewPager(viewPager);
16    }
17
18 }
19
20
21
22 }
```

이 코드를 작성해주셔야 합니다.



이렇게 완성이 됐습니다!

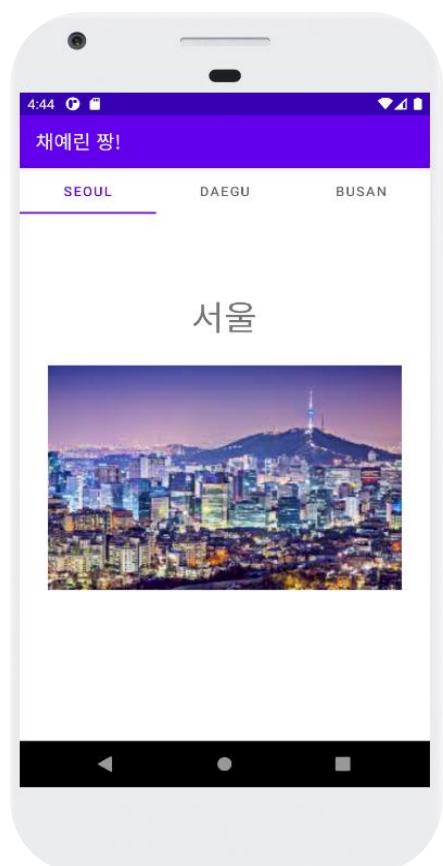
그리고 팁을 알려 드리자면



The screenshot shows the Android Studio project structure. Under the 'values' folder, there are three files: 'colors.xml', 'strings.xml', and 'styles.xml'. A pink arrow points from the 'strings.xml' file in the file tree to the code editor below. The code editor displays the following XML:

```
1 <resources>
2     <string name="app_name">채예린 짱!</string>
3     <!-- TODO: Remove or change this placeholder text -->
4     <string name="hello_blank_fragment">Hello blank fragment</string>
5 </resources>
```

Values에 들어가서 strings를 누르고 string 사이에  
원하는 말을 쓰면 이렇게 나타낼 수  
있습니다.



# 9

## Permission

Permission에 대해 알아보고 활용할 수 있다.

by : 김우진

## 위험 권한

Android API 22 이하에서는 manifest.xml에 넣어둔 앱에서 필요한 권한을 앱을 설치할 때 사용자가 허용하면 한꺼번에 권한이 부여되었습니다. 하지만 사용자들은 그 권한이 무엇을 의미하는지도 잘 모르는 경우가 많아서 권한에 대한 설명을 읽어보지도 않고 그냥 설치하는 경우가 많았습니다. 이를 악용하여 악성 앱을 배포하여 피해를 보는 사람들이 생겨났습니다. 이런 문제점을 해결하고자 구글은 API 23(마시멜로) 버전부터는 권한을 일반 권한과 위험 권한으로 나누었습니다.

## 일반 권한과 위험 권한의 차이

그렇다면 일반 권한과 위험 권한의 차이는 무엇일까요? 일반 권한의 경우 앱을 설치할 때 사용자에게 권한이 부여되어야 함을 알려주고 설치할 것인지를 물어보고 수락되면 앱에 권한이 부여됩니다. 그러나 위험 권한의 경우에는 설치 시에 부여한 권한은 의미가 없으며 실행 시에 사용자에게 권한을 부여할 것인지를 물어보게 됩니다. 만약 사용자가 권한을 부여하지 않으면 해당 기능은 동작하지 않게 됩니다. 즉 앱을 설치했더라도 위험 권한에 대해서 제약이 생기는 것입니다. 위험 권한으로 분류된 주요 권한들은 대부분 개인정보가 담겨있는 정보에 접근하거나 단말의 장치에 접근할 때 부여됩니다.

# 세부 권한

권한 그룹	권한
CALENDAR	<ul style="list-style-type: none"><li>• READ_CALENDAR</li><li>• WRITE_CALENDAR</li></ul>
CAMERA	<ul style="list-style-type: none"><li>• CAMERA</li></ul>
CONTACTS	<ul style="list-style-type: none"><li>• READ_CONTACTS</li><li>• WRITE_CONTACTS</li><li>• GET_ACCOUNTS</li></ul>
LOCATION	<ul style="list-style-type: none"><li>• ACCESS_FINE_LOCATION</li><li>• ACCESS_COARSE_LOCATION</li></ul>
MICROPHONE	<ul style="list-style-type: none"><li>• RECORD_AUDIO</li></ul>
PHONE	<ul style="list-style-type: none"><li>• READ_PHONE_STATE</li><li>• CALL_PHONE</li><li>• READ_CALL_LOG</li><li>• WRITE_CALL_LOG</li><li>• ADD_VOICEMAIL</li><li>• USE_SIP</li><li>• PROCESS_OUTGOING_CALLS</li></ul>
SENSORS	<ul style="list-style-type: none"><li>• BODY_SENSORS</li></ul>
SMS	<ul style="list-style-type: none"><li>• SEND_SMS</li><li>• RECEIVE_SMS</li><li>• READ_SMS</li><li>• RECEIVE_WAP_PUSH</li><li>• RECEIVE_MMS</li></ul>
STORAGE	<ul style="list-style-type: none"><li>• READ_EXTERNAL_STORAGE</li><li>• WRITE_EXTERNAL_STORAGE</li></ul>

# 9-2 Permission 부여하기

## 위험 권한 부여 방법

위험 권한을 부여하는 방법을 알아봅시다. 먼저 AndroidManifest.xml 파일에 사용할 위험 권한을 추가해야 합니다.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.a1121">
    <uses-permission android:name="android.permission.CALL_PHONE"/>
    <application...>
</manifest>
```

기본 권한을 부여할 때는 <uses-permission>이라는 태그를 사용합니다. 앱 권한을 허가했더라도, 사용자가 나중에 허용을 취소할 수도 있기 때문에 권한 요청이 필요한 경우 매번 권한이 허가되었는지 확인하는 것이 좋습니다. 또한, 사용자가 수락했는지 아니면 거부 했는지의 여부를 콜백 메서드로 받아 확인하는 것이 필요합니다.

```
if (ContextCompat.checkSelfPermission(thisActivity,
    Manifest.permission.WRITE_CALENDAR) == PackageManager.PERMISSION_GRANTED) {
    // Permission is granted
}

if (ContextCompat.checkSelfPermission(thisActivity,
    Manifest.permission.READ_CONTACTS) == PackageManager.PERMISSION_GRANTED) {
    // 거절당했는가?
    if (ActivityCompat.shouldShowRequestPermissionRationale(thisActivity,
        Manifest.permission.READ_CONTACTS)) {
        // 설명 필요함: 권한을 수동 허가해 주어야 함
    } else {
        // 설명 필요 없이 권한을 요청합니다.
        ActivityCompat.requestPermissions(thisActivity,
            new String[]{Manifest.permission.READ_CONTACTS},
            MY_PERMISSIONS_REQUEST_READ_CONTACTS);
        // MY_PERMISSIONS_REQUEST_READ_CONTACTS는 요청 코드입니다.
    }
} else {
    // Permission 이미 허가됨
}

@Override
public void onRequestPermissionsResult(int requestCode,
    String[] permissions, int[] grantResults) {
    switch (requestCode) {
        case MY_PERMISSIONS_REQUEST_READ_CONTACTS: {
            // 요청을 취소했다면 배열은 비어서 올 것임.
            if (grantResults.length > 0
                && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                // 권한 허가됨
            } else {
                // 권한 거부됨.
            }
            return;
        }
        // 다른 퍼미션에 대한 체크
    }
}
```

# 9-3

# Permission 예제

## UI

(위에서부터)

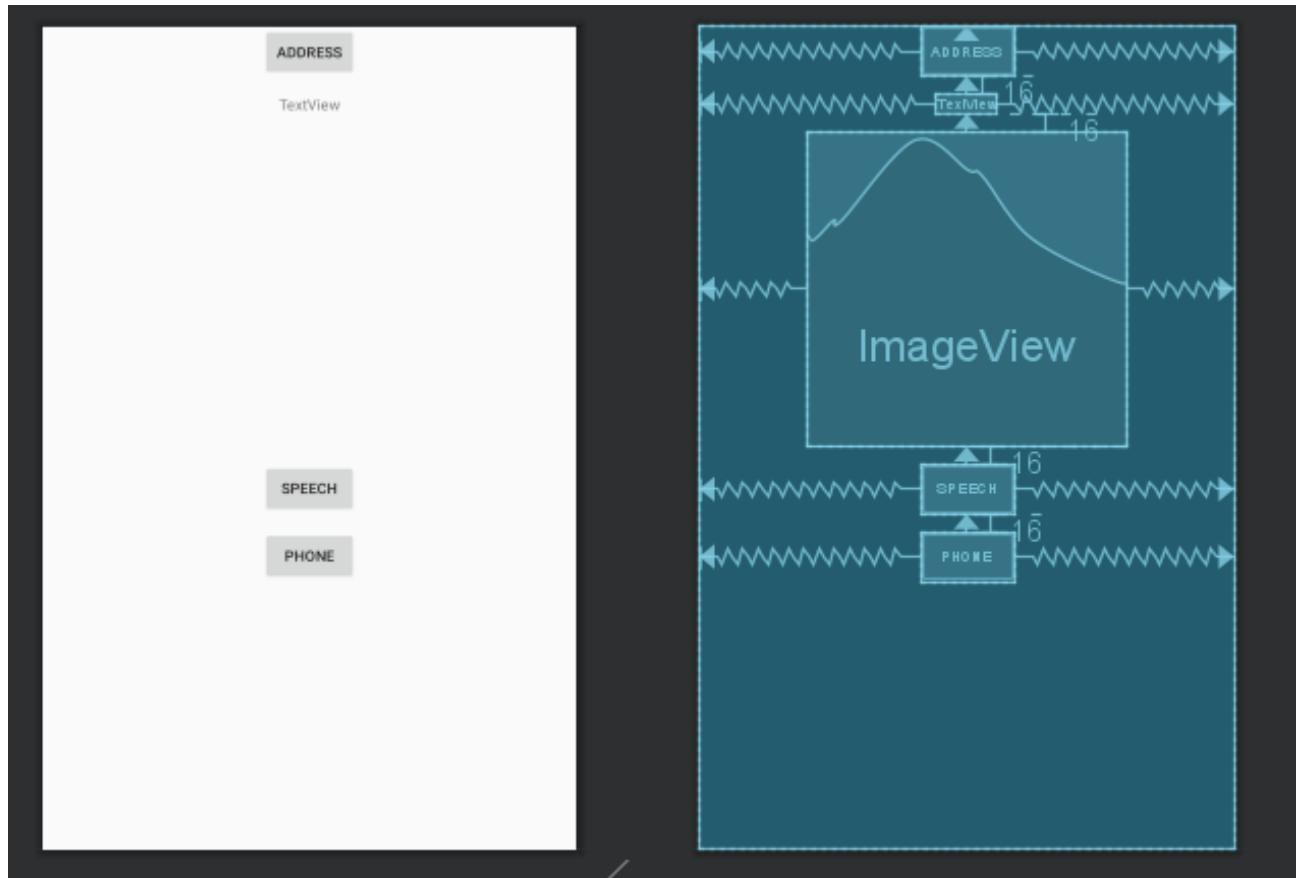
Button id : btnAddress

TextView id : txt

ImageView id : imageView

Button id : btnSpeech

Button id : btnPhone



# AndroidManifest.xml

권한 요청

```
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.a1121">

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

    <application...>

</manifest>
```

# MainActivity.java

```
public class MainActivity extends AppCompatActivity {

    Button btnAddress;
    Button btnSpeech;
    Button btnPhone;
    TextView txt;
    ImageView imageView;

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

        btnAddress = findViewById(R.id.btnAddress);
        btnPhone = findViewById(R.id.btnPhone);
        btnSpeech = findViewById(R.id.btnSpeech);
        txt = findViewById(R.id.txt);
        imageView = findViewById(R.id.imageView);
```

# MainActivity.java

주소록 선택 인텐트  
주소록을 호출

```
btnAddress.setOnClickListener(v-> startActivityForResult(  
    new  
    Intent(Intent.ACTION_PICK).setData(ContactsContract.Contacts  
.CONTENT_URI), 1001));
```

주소록 실행  
(onCreate 뒷, MainActivity안에 작성)

```
@Override  
protected void onActivityResult(int requestCode, int  
resultCode, @Nullable Intent data) {  
    super.onActivityResult(requestCode, resultCode, data);  
    if (requestCode==1001){  
        txt.setText(data.getDataString());  
        startActivity(new  
        Intent(Intent.ACTION_VIEW).setData(Uri.parse(data.getDataStr  
ing())));  
    }  
}
```

갤러리 이미지 가져오기  
갤러리 호출

```
imageView.setOnClickListener(v ->  
startActivityForResult(  
    new Intent(Intent.ACTION_GET_CONTENT)  
    .setType("image/*"),1002));
```

# MainActivity.java

갤러리 실행

```
else if(requestCode == 1002)
{
    try {
        InputStream i =
getContentResolver().openInputStream(data.getData());
        Bitmap img = BitmapFactory.decodeStream(i);
        i.close();
        imageView.setImageBitmap(img);
    } catch (Exception ignored) {}
}
```

음성인식 Intent

```
btnSpeech.setOnClickListener(v -> startActivityForResult(
    new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH)

    .putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
    RecognizerIntent.LANGUAGE_MODEL_FREE_FORM)
        .putExtra(RecognizerIntent.EXTRA_PROMPT,
    "음성인식"), 1003));
```

```
else if(requestCode == 1003)
{
    ArrayList<String> results =
data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
    txt.setText(results.get(0));
}
```

# MainActivity.java

btnPhone 클릭시

```
btnPhone.setOnClickListener(v -> callphone());
```

callphone 메서드

```
private void callphone() {
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.CALL_PHONE) != PackageManager.PERMISSION_GRANTED)
        ActivityCompat.requestPermissions(this,new String[]{Manifest.permission.CALL_PHONE},1000);
    else
        startActivity(new Intent(Intent.ACTION_CALL,
Uri.parse("tel:02-120")));
}
```

권한요청

```
@Override
public void onRequestPermissionsResult(int requestCode,
@NonNull String[] permissions, @NonNull int[] grantResults)
{
    super.onRequestPermissionsResult(requestCode,
permissions, grantResults);
    if(requestCode == 1000)
        if(grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED)
            callphone();
}
```

# 10

## Firebase

Android Studio에 Firebase를 추가해봅시다

by : 이현주, 이예홍

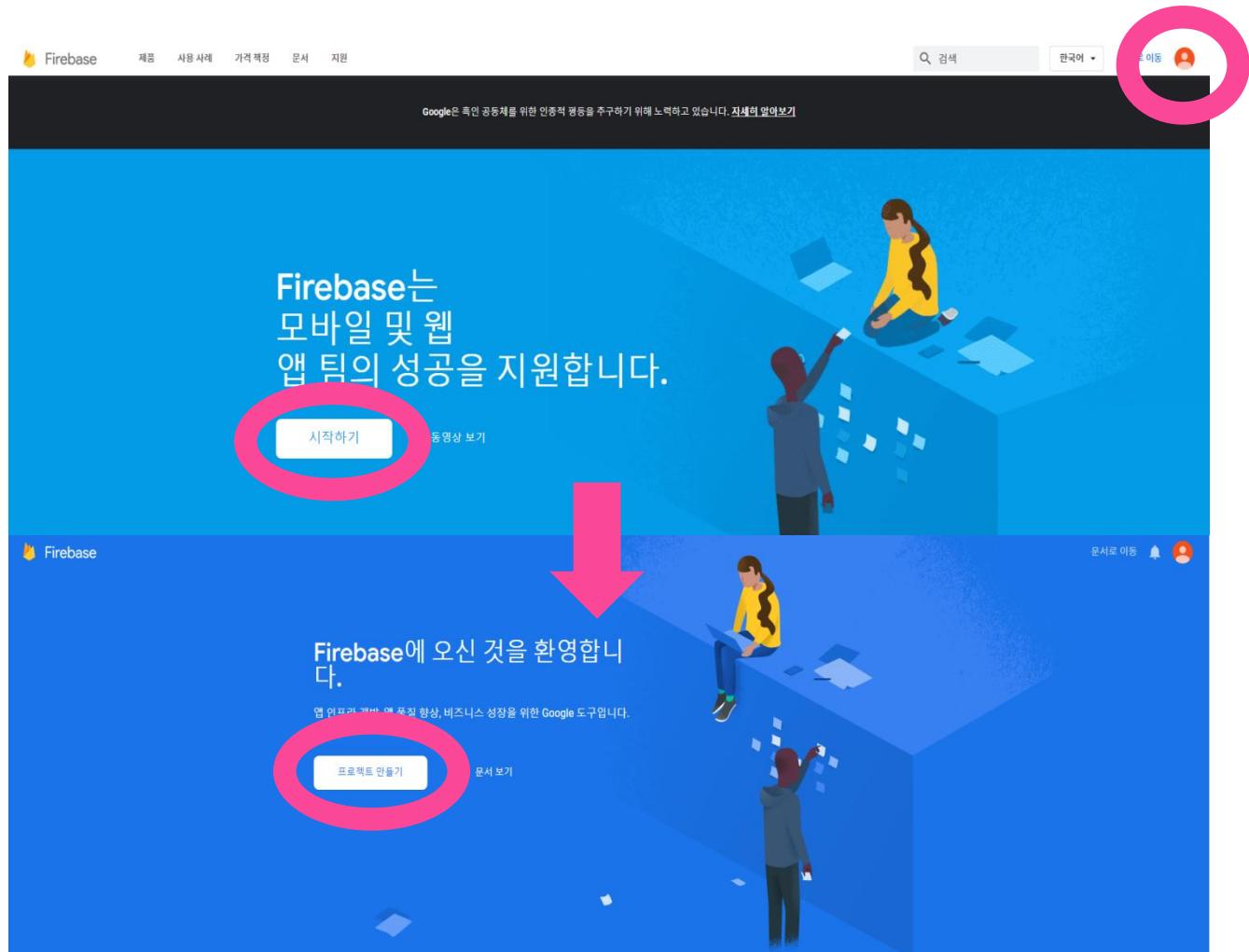
# 10-1 Firebase 개요 및 설치

## Firebase란?

모바일 서버를 지원하는 구글의 플랫폼입니다. 지금까지는 오프라인 사용을 염두에 둔 앱만 개발해 왔으나, 이번에는 Firebase의 기능 중 하나인 데이터베이스와 Android Studio를 연동하여 인터넷을 본격적으로 활용하는 앱을 개발해보도록 하겠습니다.

## Firebase 프로젝트 생성 및 연결

1. 브라우저 검색창에 [firebase.google.com](https://firebase.google.com)를 입력하여 Firebase 홈페이지에 들어간 후, 개인 구글 계정으로 로그인합니다. 이후 시작하기, 프로젝트 만들기 버튼을 차례대로 눌러 프로젝트를 생성합니다.



2. 원하는 이름을 작성한 후, 다음 화면으로 넘어가 구글 애널리틱스 사용 설정을 해제합니다. 이후 프로젝트 만들기 버튼을 클릭합니다.

프로젝트® 이름을 지정하여 시작하기

프로젝트 이름  
fb-test

fb-test-32d9b

계속

× 프로젝트 만들기(1/3단계)

Firebase 프로젝트를 위한 Google 애널리틱스

무제한 무료 애널리틱스 솔루션인 Google 애널리틱스를 사용하면 Firebase Crashlytics, 플랫폼 메시징, 인앱 메시지, 원격 구성, A/B 테스팅, 예측, Cloud Functions에서 타겟팅, 보고 등을 이용할 수 있습니다.

Google 애널리틱스를 통해 다음 기능을 이용할 수 있습니다.

- × AI용 머신러닝 ①
- × 장애가 발생하지 않은 사용자 ①
- × Firebase 제품 전반에서 사용자 세분화 ①
- × 이벤트 기반 Cloud Functions 트리거 ①
- × 차환 막는 무결성 보고 ①
- × 사용자 행동 예측 ①

이전

이전 프로젝트에서 Google 애널리틱스 사용 설정

프로젝트 만들기

× 프로젝트 만들기(2/2단계)

3. 새 프로젝트가 준비되었다는 코멘트가 뜨면 계속을 누르고, Android Studio에 Firebase를 추가할 것이므로 안드로이드 버튼을 클릭합니다.

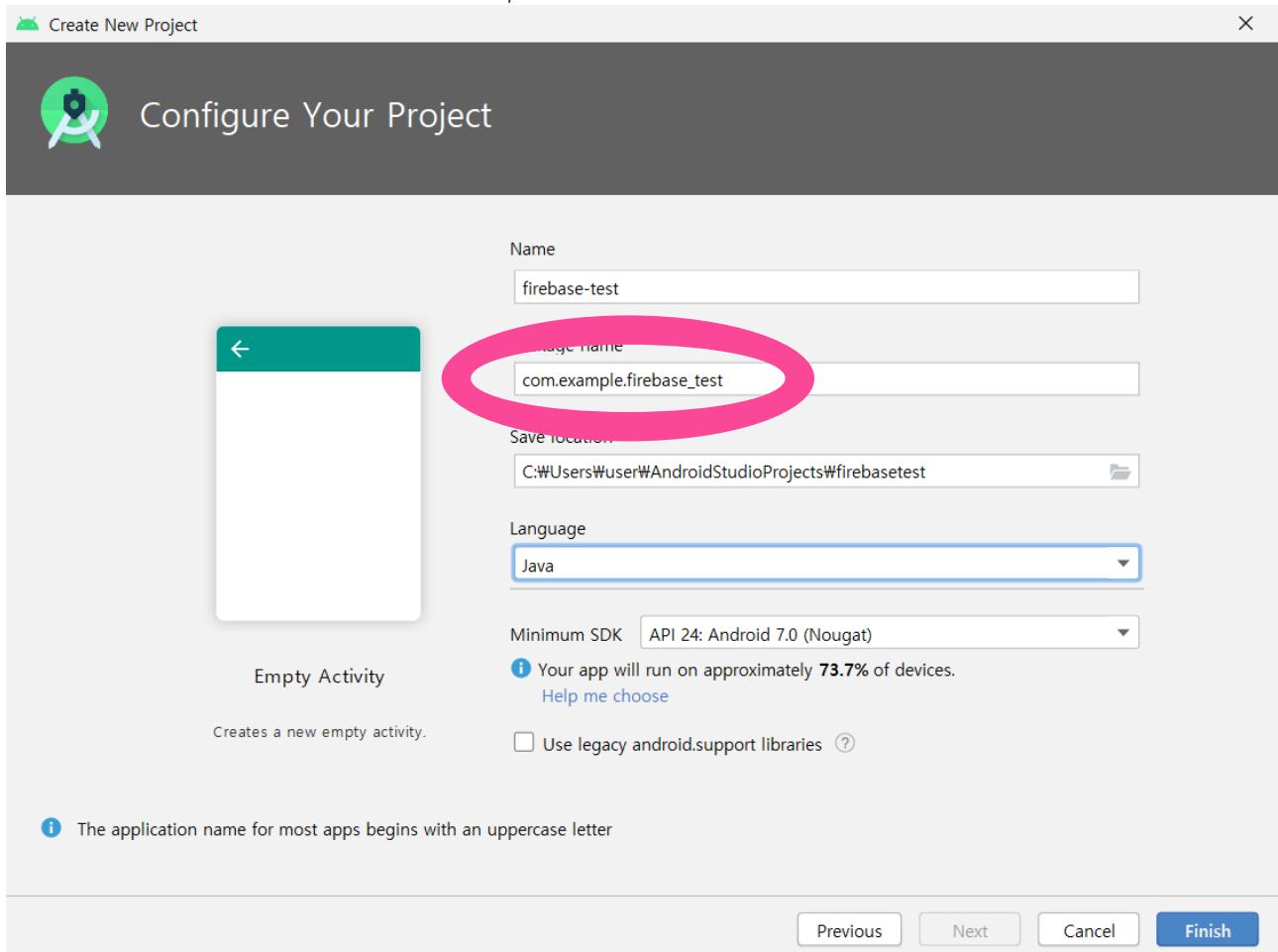
fb-test Spark 요금제

새 프로젝트가 준비되었습니다!

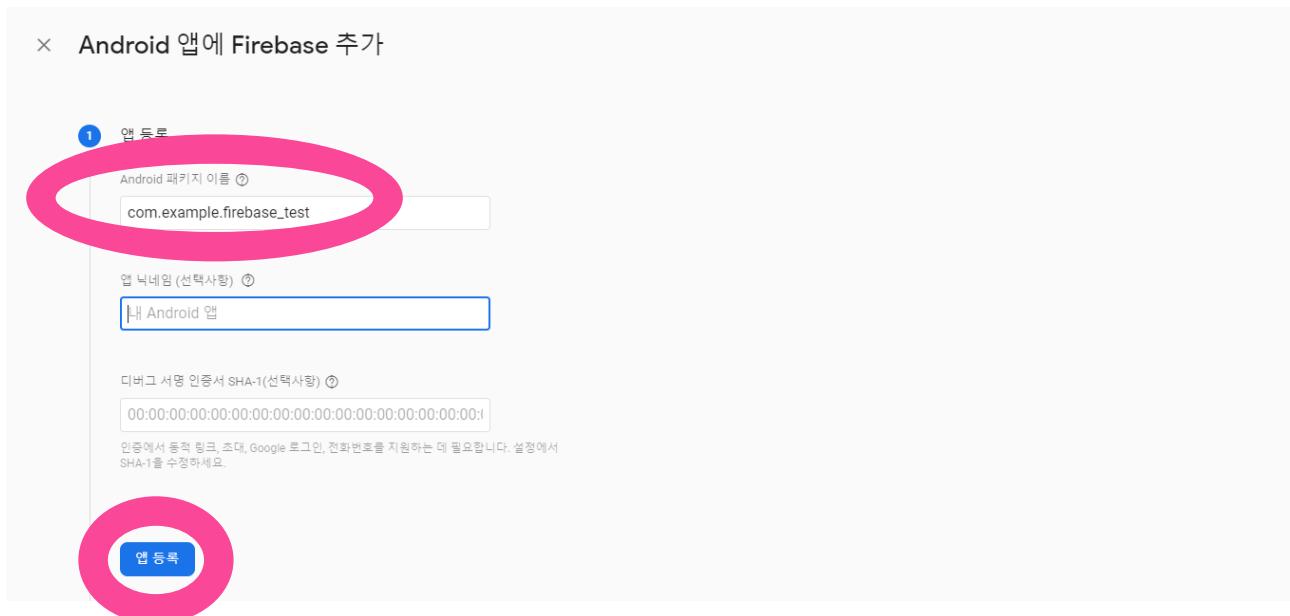
앱에 Firebase를 추가하여 시작하기

안드로이드 추가하기

4. Firebase 창을 대기 상태로 두고, Android Studio 프로그램으로 이동하여 새 프로젝트를 생성합니다. 이때, 패키지 명을 미리 복사해둡니다.



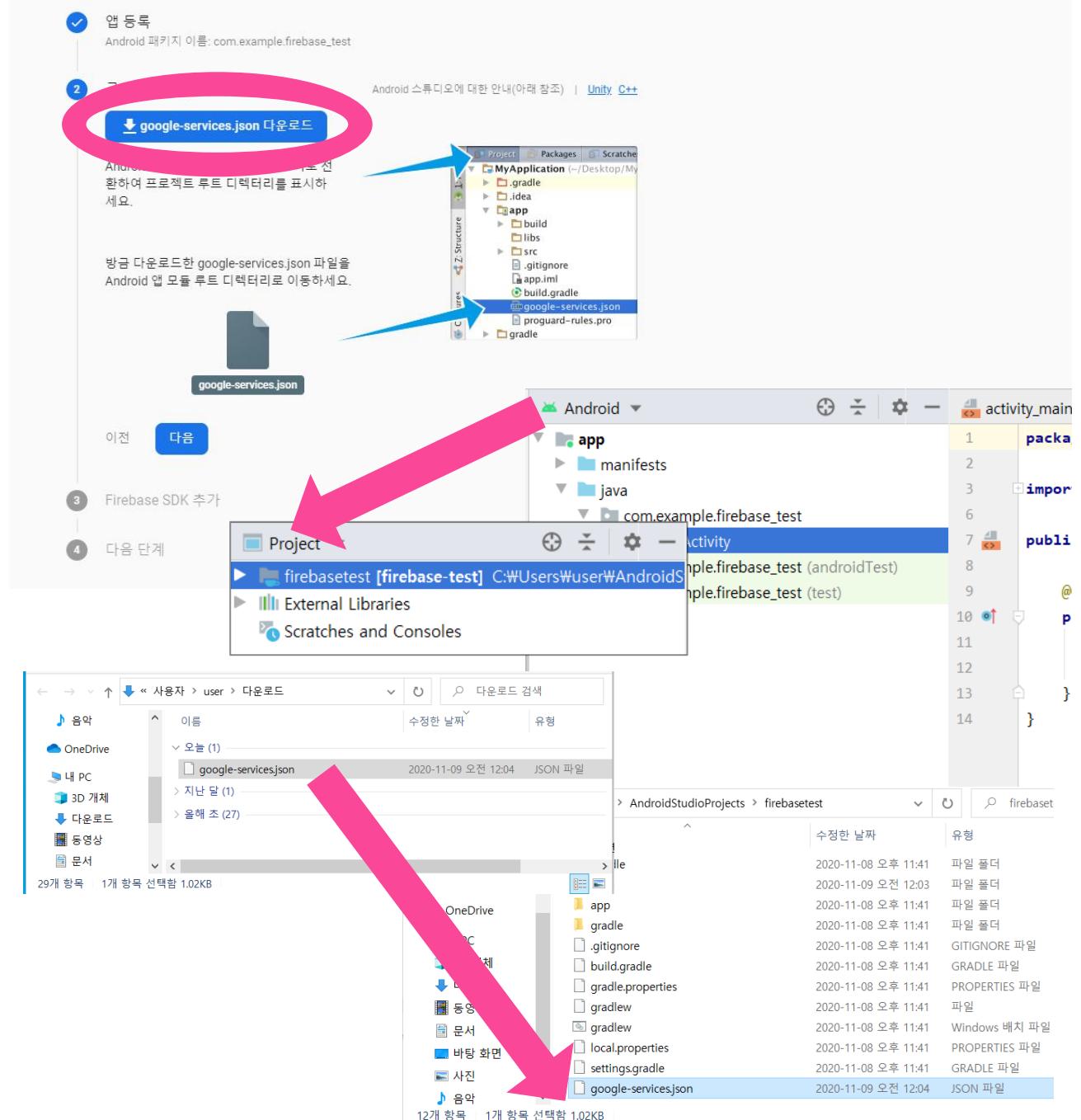
5. 다시 Firebase 창으로 돌아와 Android 패키지 이름 창에 이전에 복사해둔 패키지 명을 붙여넣기하고, 나머지 부분은 선택사항이므로 넘어가서 앱 등록을 누릅니다.



6. google-services.json 다운로드 버튼을 클릭하여 다운받고, android studio 프로그램으로 돌아가 좌측 상단의 android 보기 정렬을 project 보기 정렬로 변경합니다. 이후 나타난 프로젝트 명을 파일 탐색기에 검색하여 android 프로젝트 폴더를 찾아 다운받은 google-services.json을 프로젝트 파일 안에 그대로 붙여넣기합니다. 만약 여러 개를 다운받아 파일 이름이 달라졌다면 원래 이름으로 나타나도록 이름을 변경합니다.

→ 간혹 Build 오류가 나는 경우는 프로젝트 폴더 안의 'app' 폴더에 google-services.json을 붙여넣기합니다.

#### × Android 앱에 Firebase 추가



7. 웹페이지의 설명에 따라 classpath 'com.google.gms:google-services:4.3.4' 부분의 복사버튼을 눌러 복사하고, android studio 프로젝트의 Gradle Script의 프로젝트 수준 build.gradle에 붙여넣습니다.

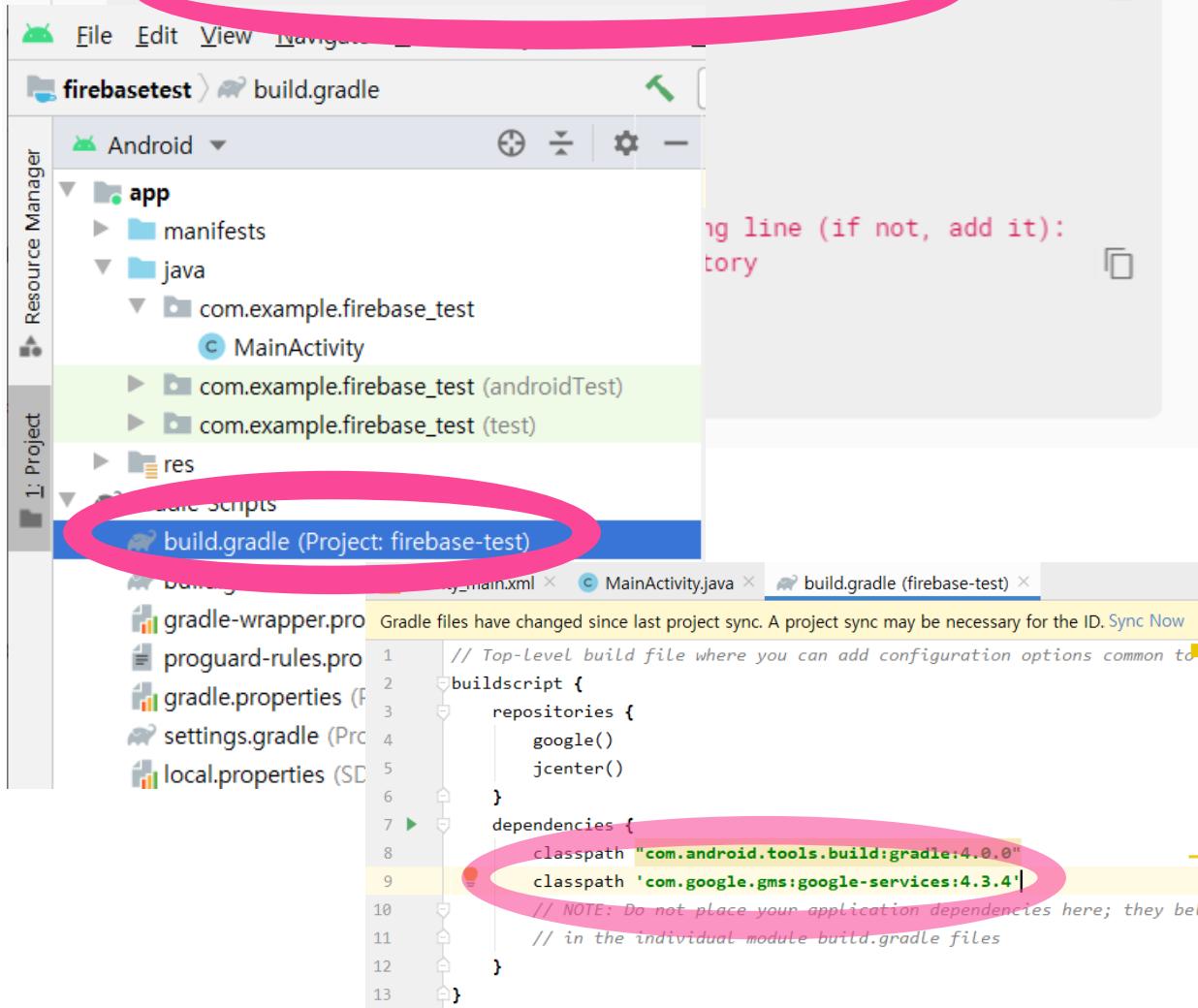
### 3 Firebase SDK 추가

Gradle 안내 | Unity C++

Gradle 을 Google 서비스 플러그인에서 방금 다운로드한 google-services.json 파일을 로드합니다. build.gradle 파일을 수정하여 플러그인을 사용하세요.

프로젝트 수준의 build.gradle (<project>/build.gradle):

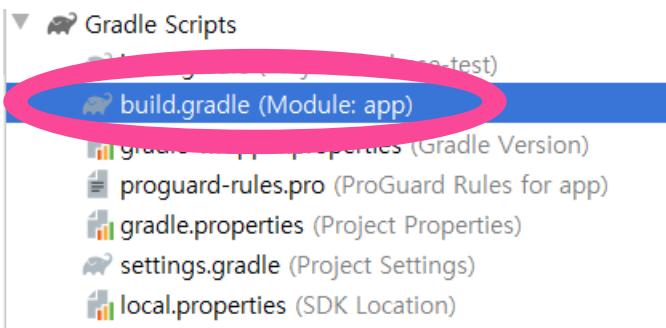
```
buildscript {  
    repositories {  
        // Check that you have the following line (if not, add it):  
        google() // Google's Maven repository  
    }  
    dependencies {  
        ...  
        // Add the following dependency:  
        classpath 'com.google.gms:google-services:4.3.4'  
    }  
}
```



8. 마찬가지로 웹페이지의 설명에 따라 apply plugin:  
'com.google.gms.google-services' 을 복사한 뒤, Gradle Script에서  
앱 수준의 build.gradle를 찾아 붙여넣습니다. 이후 코드 맨 위쪽의 sync Now  
를 클릭하여 동기화를 시켜줍니다.

앱 수준의 build.gradle (<project>/<app-module>/build.gradle):

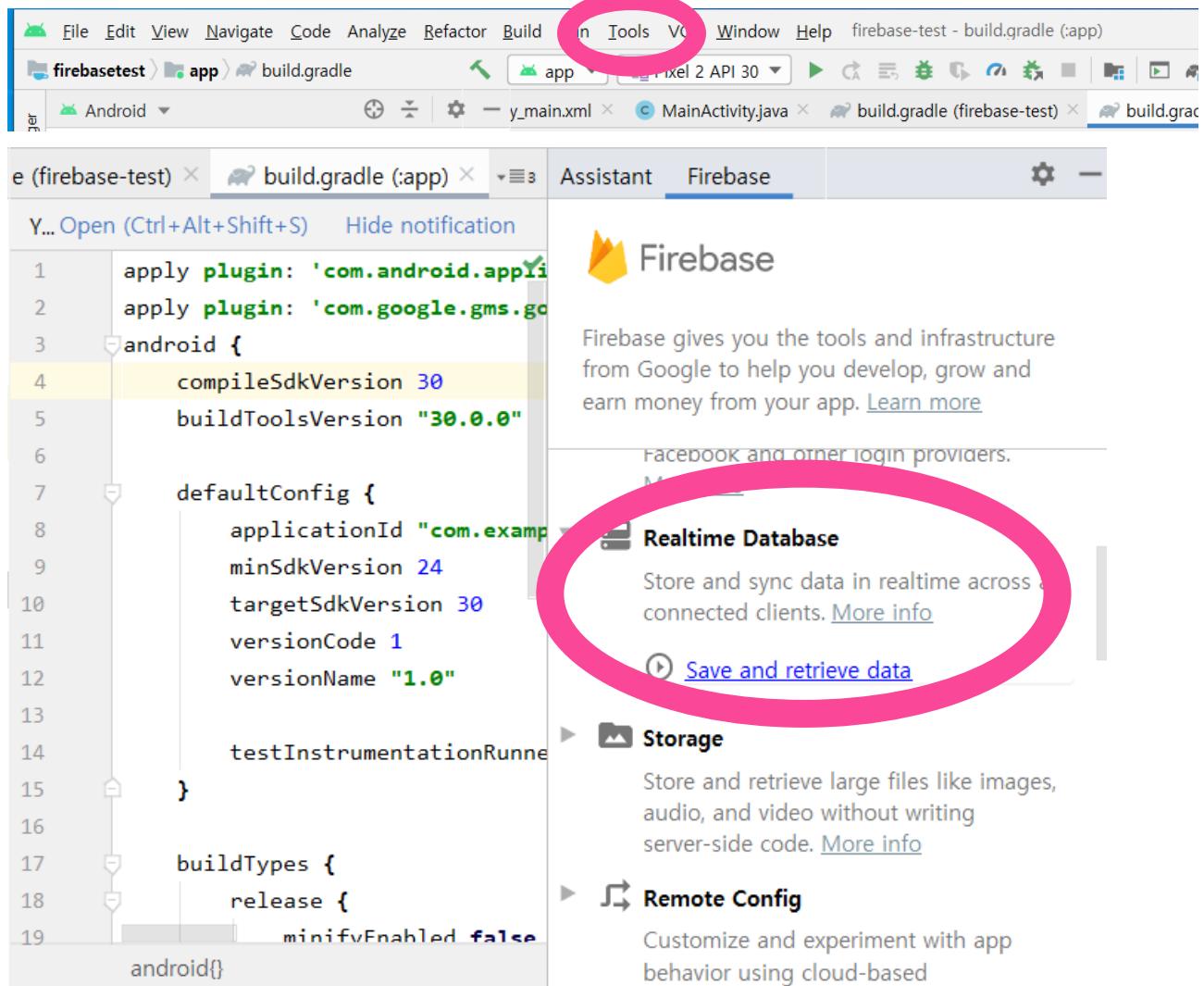
```
apply plugin: 'com.android.application'  
// ...  
apply plugin: 'com.google.gms.google-services'  
  
dependencies {  
    // Import the Firebase BoM  
    implementation platform('com.google.firebase:firebase-bom:26.0.0')  
  
    // Add the dependencies for the desired Firebase products  
    // https://firebase.google.com/docs/android/setup#available-libraries  
}
```



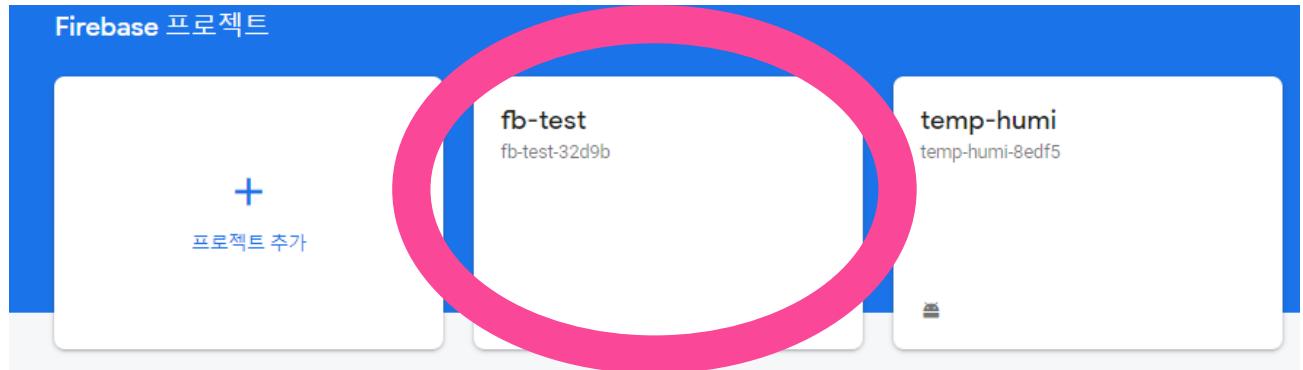
Gradle files have changed since last project sync. A project sync may be necessary for the following changes: [ID. Sync Now](#)

```
1 apply plugin: 'com.android.application'  
2 apply plugin: 'com.google.gms.google-services'  
3 android {  
4     compileSdkVersion 30  
5     buildToolsVersion "30.0.0"  
6  
7     defaultConfig {  
8         applicationId "com.example.firebaseio_test"
```

9. 이제 Android Studio와 Firebase를 연동해보겠습니다. Android Studio 상단의 Tools에서 firebase를 찾아 클릭하고, 프로그램 우측에 나타나는 목록들 중 Realtime Database의 Save and retrieve data를 선택합니다.



10. 1번의 Connecting to firebase 버튼을 클릭하고, firebase 프로젝트 창으로 돌아가 생성한 프로젝트를 선택하면 연결이 완료됩니다.



## Save and retrieve data

Our cloud database stays synced to all connected clients. Data remains available when your app goes offline. Data is stored in a JSON-like structure rather than a table, eliminating the need for schema design.

[Launch in browser](#)

### ① Connect your app to Firebase

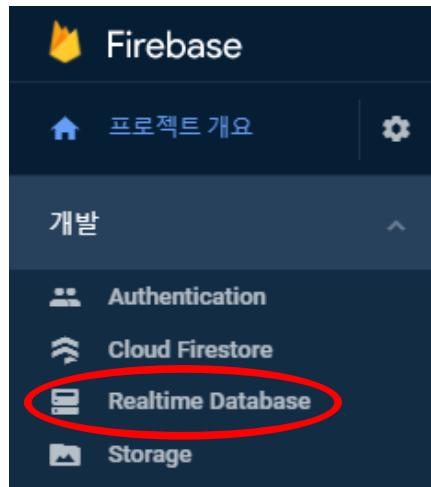
✓ Connected

11. 2번의 Add the Realtime Database to your app 버튼을 클릭한 뒤 뜨는 팝업창의 Accept Changes 버튼을 눌러줍니다.

### ② Add the Realtime Database to your app

✓ Dependencies set up correctly

12. 다시 Firebase 설정 창으로 돌아와 콘솔로 이동 버튼을 눌러주고, 왼쪽 메뉴 중 Database를 선택해줍니다.



13. 화면에 뜨는 데이터베이스 만들기 버튼을 바로 클릭하여 Realtime database를 생성하고, 잠금 모드로 실행해줍니다. 이후 사용 설정 버튼을 누르면 나오는 창의 규칙 탭에서 read와 write창을 모두 false에서 true로 바꾸고 게시 버튼을 눌러줍니다(나오는 경고창은 무시해도 됩니다).

Realtime Database

실시간으로 데이터 저장 및 동기화

데이터베이스 만들기

실시간 데이터베이스 보안 규칙

데이터 구조를 정의한 후 데이터의 보안을 강화하는 규칙을 작성해야 합니다.

[자세히 알아보기](#)

잠금 모드에서 시작

모든 읽기 및 쓰기를 거부하여 데이터베이스를 비공개로 설정합니다.

테스트 모드에서 시작

데이터베이스에 대한 모든 읽기 및 쓰기를 허용하여 빠르게 설정합니다. 보안 규칙이 업데이트되지 않으면 30일 후 클라이언트 읽기/쓰기 액세스가 거부됩니다.

```
{  
  "rules": {  
    ".read": false,  
    ".write": false  
  }  
}
```

모든 제3자 읽기 및 쓰기가 거부됩니다.

취소 사용 설정

Realtime Database

데이터 벡업 사용량

규칙

규칙 수정 규칙 모니터링

게시되지 않은 변경사항

게시 삭제

규칙 플레이그라운드

```
1  {  
2   "rules": {  
3     ".read": true,  
4     ".write": true  
5   }  
6 }
```

# 10-2 Firebase 데이터 수신

## Firebase에서 데이터 가져오기

10단원(Firebase)에 이어서 Android Studio에서 제공하는 예제를 계속 만들어봅시다.

Database 툴(Tool)에서 5번 예제에 있는 코드를 그대로 복사해서 아래 사진처럼 MainActivity.java에 붙여 넣어 주세요.

```
public class MainActivity extends AppCompatActivity {

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

        // Write a message to the database
        FirebaseDatabase database = FirebaseDatabase.getInstance();
        DatabaseReference myRef = database.getReference( path: "message");

        myRef.setValue("Hello, World!");

        // Read from the database
        myRef.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                // This method is called once with the initial value and again
                // whenever data at this location is updated.
                String value = dataSnapshot.getValue(String.class);
                Log.d(TAG, "Value is: " + value);
            }

            @Override
            public void onCancelled(DatabaseError error) {
                // Failed to read value
                Log.w(TAG, "Failed to read value.", error.toException());
            }
        });
    }
}
```

그리고 다음 코드는 이 예제에선 필요없는 부분이므로 삭제해주세요.

Log.d(TAG, "Value is: " + value);

Log.w(TAG, "Failed to read value", error.toException());

늘 그랬듯이 빨갛게 에러가 뜨는 코드는 Alt+Enter를 눌러 import 해줍니다.

앞선 과정을 모두 수행하였다면, 최종적으로 아래 사진과 같이 결과물이 완성되었을 것입니다.

```
public class MainActivity extends AppCompatActivity {

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

        // Write a message to the database
        FirebaseDatabase database = FirebaseDatabase.getInstance();
        DatabaseReference myRef = database.getReference( path: "message");

        myRef.setValue("Hello, World!");

        // Read from the database
        myRef.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                ① // This method is called once with the initial value and again
                  // whenever data at this Location is updated.
                String value = dataSnapshot.getValue(String.class);
            }
        });

        ② @Override
        public void onCancelled(DatabaseError error) {
            // Failed to read value
        }
    });
}

}
```

위 사진에서

①은 onDataChange() 메소드로, 데이터의 값이 변했을 때마다 작동하고,  
②는 onCancelled() 메소드로, 함수 내에서 에러가 발생했을 때 작동합니다.

이제 여기에 toast(토스트)를 추가하여 Firebase와 Android Studio가 잘 연결되었는지 확인하는 예제를 만들어보겠습니다.

# toast 추가하기

이제 아래 사진과 같이 onDataChange 메소드 안에 코드 한 줄을 추가해주세요.

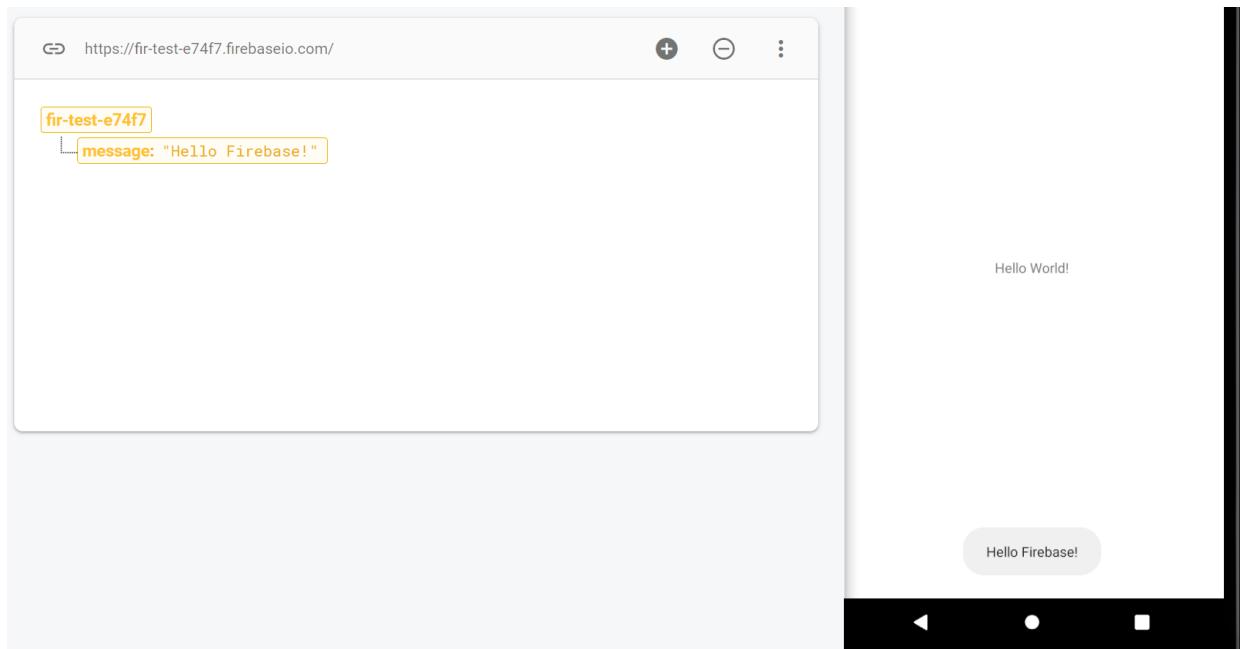
```
// Read from the database
myRef.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        // This method is called once with the initial value and again
        // whenever data at this location is updated.
        String value = dataSnapshot.getValue(String.class);
        Toast.makeText(context: MainActivity.this, value, Toast.LENGTH_LONG).show();
    }

    @Override
    public void onCancelled(DatabaseError error) {
        // Failed to read value
    }
});
```

위 코드는 Firebase의 데이터 내용이 바뀔 경우 바뀐 내용을 toast로 띄워줍니다.

## 실행하기

정확히 작성하였다면, 아래 사진처럼 firebase의 데이터를 수정했을 때, 수정된 내용이 toast로 출력되는 것을 볼 수 있습니다.



# 10-3 Firebase 데이터 송신

## Firebase 송신 예제

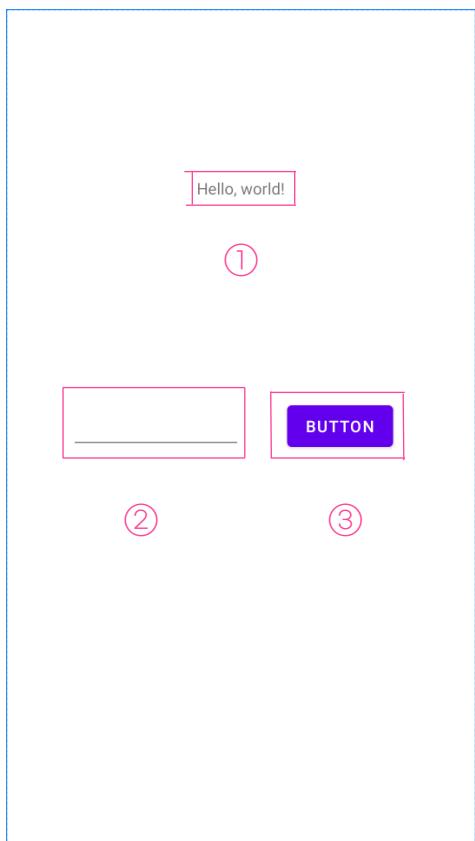
마지막 단계까지 도착하신 여러분을 진심으로 환영합니다!

이제 Firebase에 데이터를 보내주는 예제를 만들어봅시다.

이 예제 역시 10단원(Firebase)와 11-1단원(Firebase 데이터 수신)에 이어서 같은 프로젝트로 진행하겠습니다.

### Activity 구성

먼저 activity\_main.xml에 아래 사진과 같이 액티비티를 구성해주세요.



위 사진에서

①은 TextView로, id는 textView로 해주세요.

②는 EditText로, id는 editText로 해주세요.

③은 Button로, id는 button로 해주세요.

※activity 구성이 어려우면 2단원(Activity)와 3단원(View Components)를 참고해주세요.

# MainActivity.java 구성

일단 지금까지 썼던 firebase 기본 예제 코드는 전부 지워주시고 처음 상태로만 들어주세요. 그리고 아래 사진과 같이 activity의 요소들부터 선언해줍시다.

```
package com.example.firebaseio_test;

import ...

public class MainActivity extends AppCompatActivity {
    private TextView textView;
    private EditText editText;
    private Button button;
    DatabaseReference mRootRef = FirebaseDatabase.getInstance().getReference();
    DatabaseReference conditionRef = mRootRef.child("text");

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

        textView = (TextView) findViewById(R.id.textView);
        editText = (EditText) findViewById(R.id.editText);
        button = (Button) findViewById(R.id.button);
    }
}
```

데이터베이스에서 데이터를 읽고 쓰려면 DatabaseReference가 필요합니다. 그리고 .child()는 간단히 말하면 데이터가 있을 위치의 이름을 정해주는 것입니다.

빨간 줄로 에러가 뜨면 Alt+Enter로 import 해주시면 됩니다.

이제 Firebase와 연동할 수 있도록 코드를 작성해보겠습니다.  
아래 사진과 같은 코드를 **public class** 안에 작성해주세요.

```
|  
}  
protected void onStart() {  
    super.onStart();  
  
|  
}  
}
```

onStart 메소드는 데이터의 변화를 알기 위해 사용하는 것이라고 생각하시면 되겠습니다.

그리고 onStart 내부에 다음 사진과 같이 코드를 작성해주세요.

```
protected void onStart() {  
    super.onStart();  
  
    conditionRef.addValueEventListener(new ValueEventListener() {  
        |  
    });  
  
}  
}
```

.addValueEventListener() 메소드는 이전에 child로 설정해준 경로 안의 데이터가 변화됐을 때 작동합니다.

이제 아래 사진처럼 onDataChange() 메소드와 onCancelled() 메소드를 추가해주세요.

```
protected void onStart() {  
    super.onStart();  
  
    conditionRef.addValueEventListener(new ValueEventListener() {  
        @Override  
        public void onDataChange(DataSnapshot dataSnapshot) {  
            |  
        }  
  
        @Override  
        public void onCancelled(DatabaseError databaseError) {  
            |  
        }  
    });  
  
}
```

onDataChange 메소드 안에 아래 사진과 같이 코드를 추가해주세요.

```
protected void onStart() {
    super.onStart();

    conditionRef.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            String text = dataSnapshot.getValue(String.class);
            textView.setText(text);
        }

        @Override
        public void onCancelled(DatabaseError databaseError) {

        }
    });
}
```

위 코드는 text 변수 안에 받아온 데이터 문자열을 넣고, textView의 텍스트를 바꿔주는 코드입니다.

바로 아래에 버튼이 눌렸을 때 작동하는 메소드를 만들어 줍시다.

```
}

button.setOnClickListener(new View.OnClickListener() {
    |
});

});
```

그 안에는 conditionRef에 setValue를 해주어  
버튼을 누를 때마다 설정된 경로로 editText의 값이 업데이트 되게 하는 코드를  
아래 사진과 같이 넣어주세요.

```
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        conditionRef.setValue(editText.getText().toString());
    }
});
```

## 최종적으로 이렇게 코드가 완성이 됩니다.

```
package com.example.firebaseio_test;

import ...

public class MainActivity extends AppCompatActivity {
    private TextView textView;
    private EditText editText;
    private Button button;
    DatabaseReference mRootRef = FirebaseDatabase.getInstance().getReference();
    DatabaseReference conditionRef = mRootRef.child("text");

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

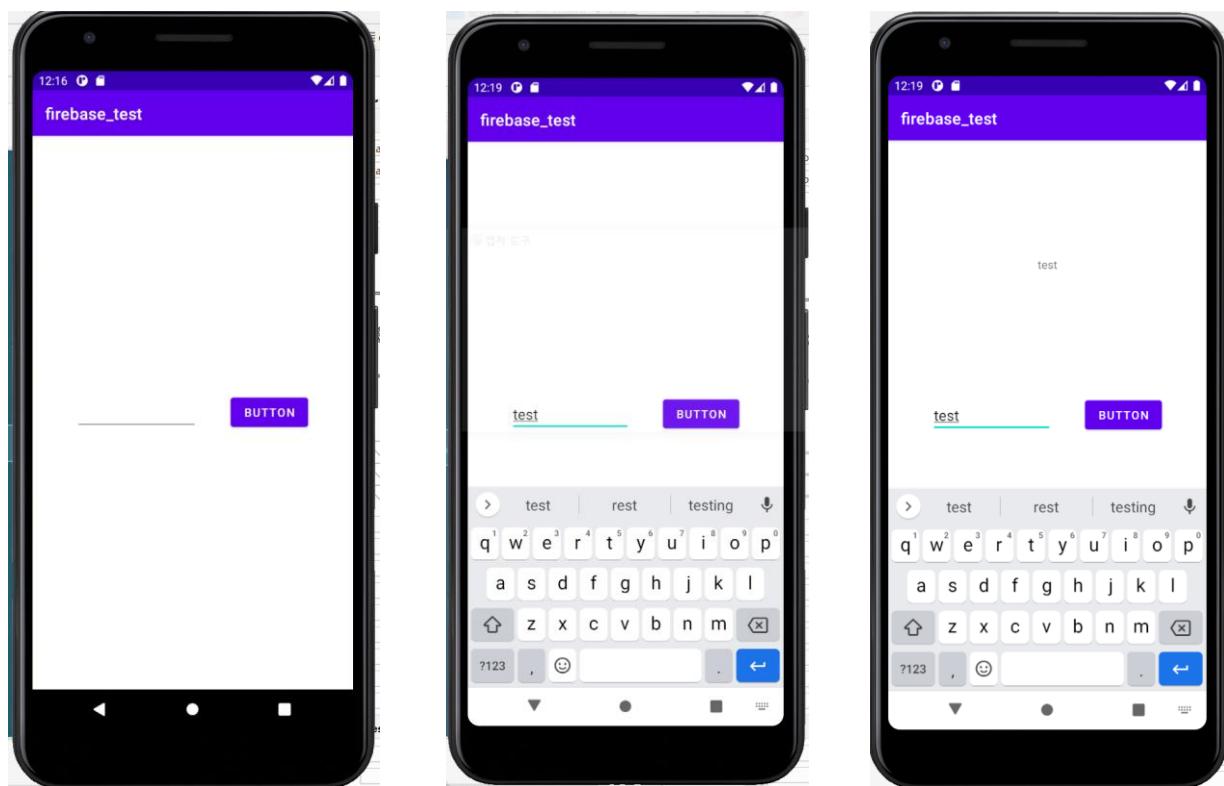
        textView = (TextView) findViewById(R.id.textView);
        editText = (EditText) findViewById(R.id.editText);
        button = (Button) findViewById(R.id.button);
    }

    protected void onStart() {
        super.onStart();
        conditionRef.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                String text = dataSnapshot.getValue(String.class);
                textView.setText(text);
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {
                ...
            }
        });
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                conditionRef.setValue(editText.getText().toString());
            }
        });
    }
}
```

# 실행하기

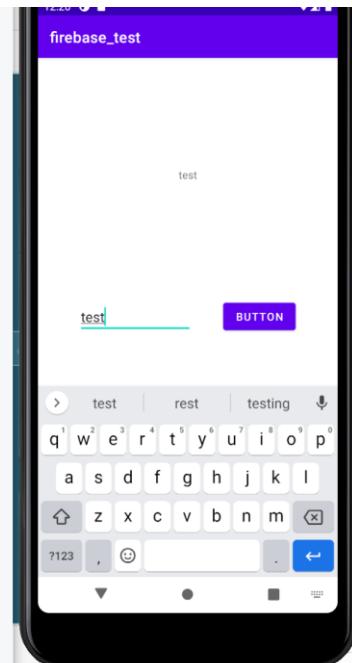
앱을 실행해보면 아래 첫 번째 사진과 같이 실행이 되고, EditText에 아무거나 쳐서 버튼(Button)을 누르면 TextView가 바뀌는 것을 볼 수 있습니다.



그리고 Realtime Database에서 EditText에 쳤던 것이 데이터로 넘어온 것을 확인할 수 있습니다.

## Realtime Database

A screenshot of the Firebase Realtime Database console. At the top, there are tabs for '데이터', '규칙', '백업', and '사용량'. The '데이터' tab is selected. Below the tabs, the URL 'https://fir-test-e74f7.firebaseio.com/' is shown. On the right, there is a search bar and a help icon. The main area displays a tree structure under the node 'fir-test-e74f7'. One branch is expanded, showing the key 'text' with the value 'test'.



# 11

## 깃허브(GitHub)

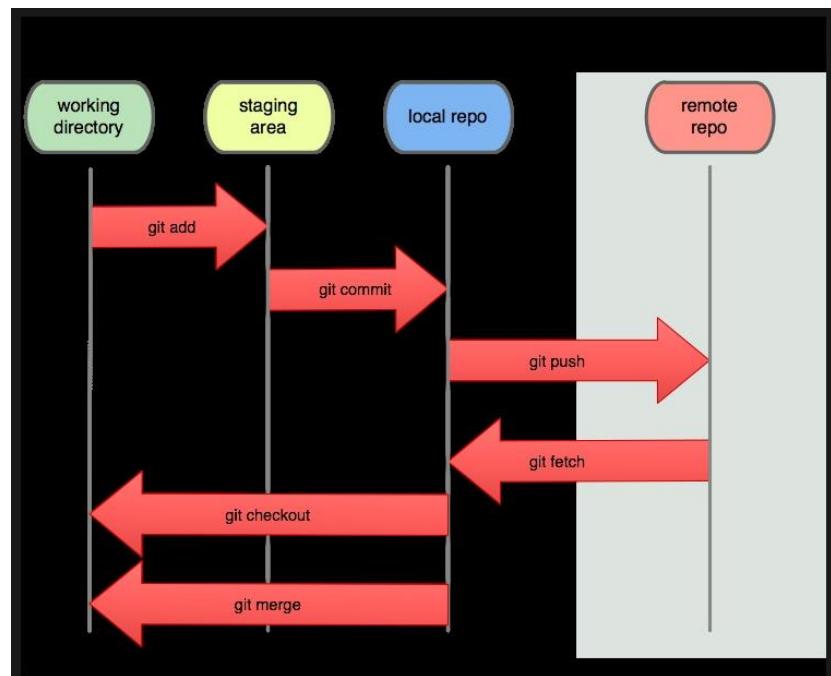
깃허브를 사용하면 소스코드를 간단하게 올리고 받을 수 있  
을 뿐만 아니라 여러 명이 진행되는 프로젝트 협업도 간편  
하게 진행할 수 있습니다.

by : 정해성

구글에 git 검색 후 다운로드  
또는  
<https://git-scm.com/downloads>  
위 주소로 들어가서 다운로드



자신에게 맞는 환경을 선택 (이하 설명은 윈도우 기준으로 합니다)

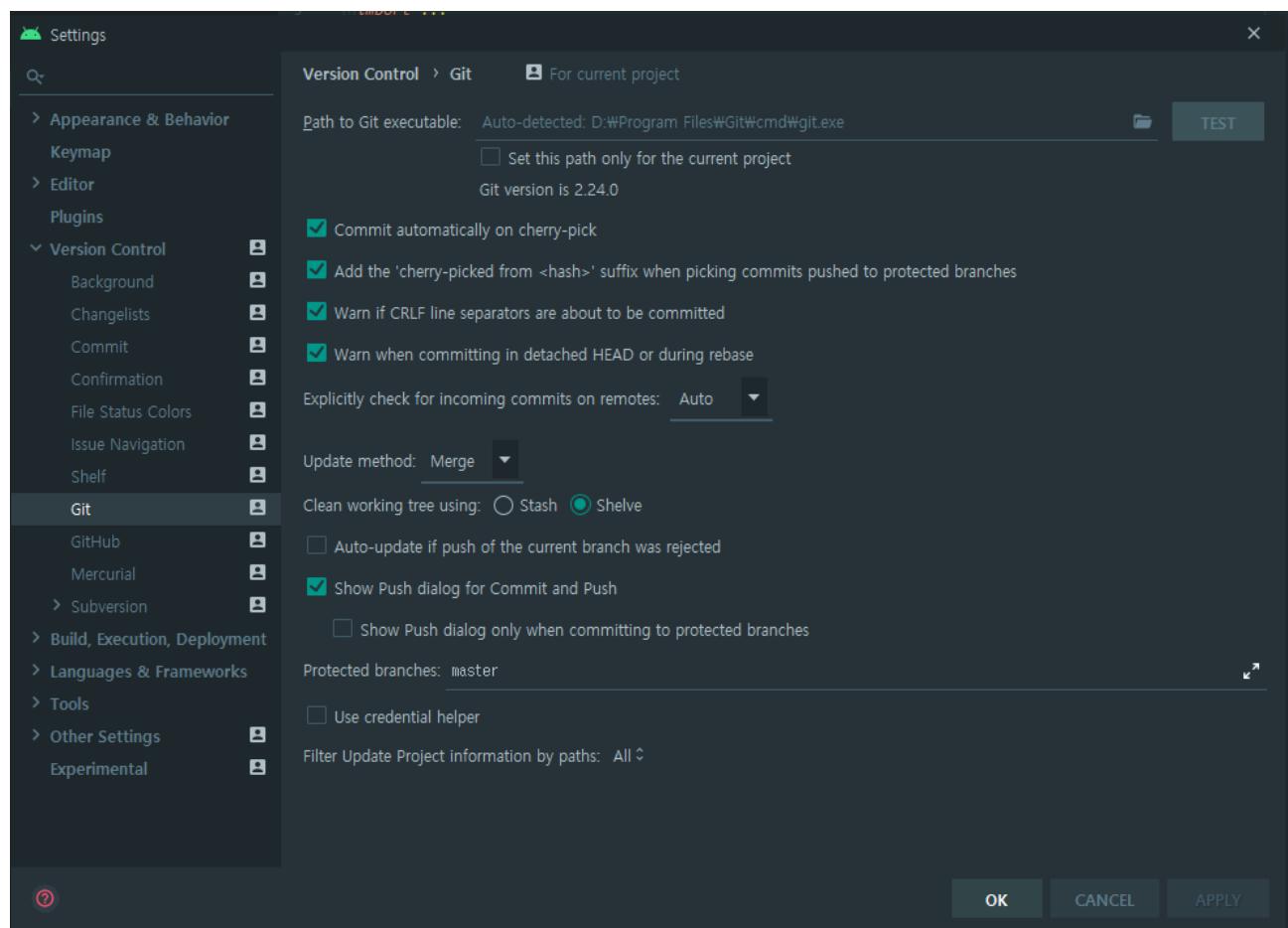


-> 깃의 작동 원리

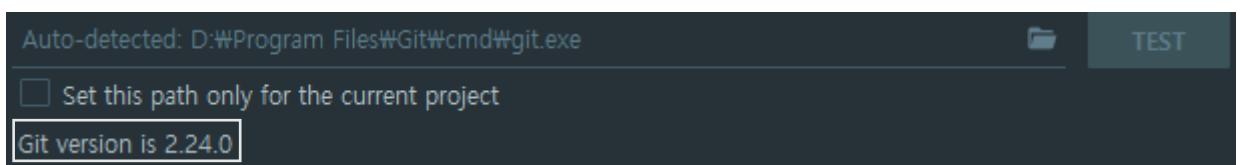
출처 :  
<http://pismute.github.io/whygitisbetter/images/local-remote.png>

Git 설치가 모두 끝났다면  
Android Studio로 들어가 주세요

File > Settings... > Version Control > Git



자신이 Git을 설치한 위치를 확인하고 TEST 버튼을 눌러서



Git version is (깃 버전) 이 뜨면 깃 세팅 끝!

<https://github.com/> 접속 후 New 버튼 클릭

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \*

Start-Android
/
GitHubUploadTest
✓

Repository name \*

Great repository names are short and memorable. Need inspiration? How about [potential-engine](#)?

Description (optional)

깃허브 업로드 테스트

Public  
Anyone on the internet can see this repository. You choose who can commit.

Private  
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

Add a README file  
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore  
Choose which files not to track from a list of templates. [Learn more.](#)

Choose a license  
A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

주의 : 프로젝트를 올리기 전에 다른 파일을 미리 생성하면 안 됨

(체크 해제 후 넘어가기)

Quick setup — if you've done this kind of thing before

복사하고 기억해두기

or

<https://github.com/Start-Android/GitHubUploadTest.git>

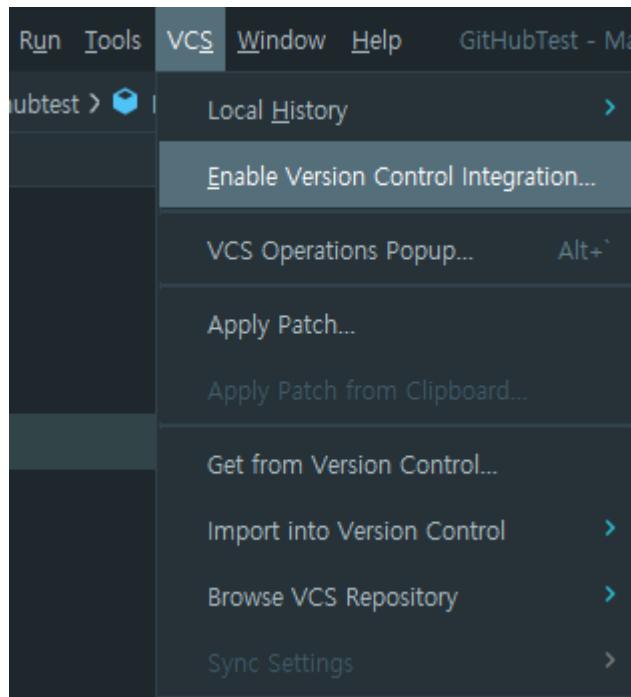
Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a README, LICENSE, and .gitignore.

처음부터 시작하는 Android Studio

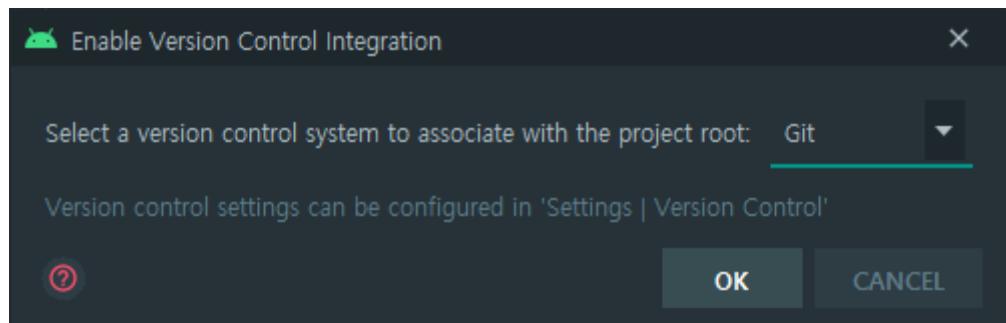
101

# 11-3

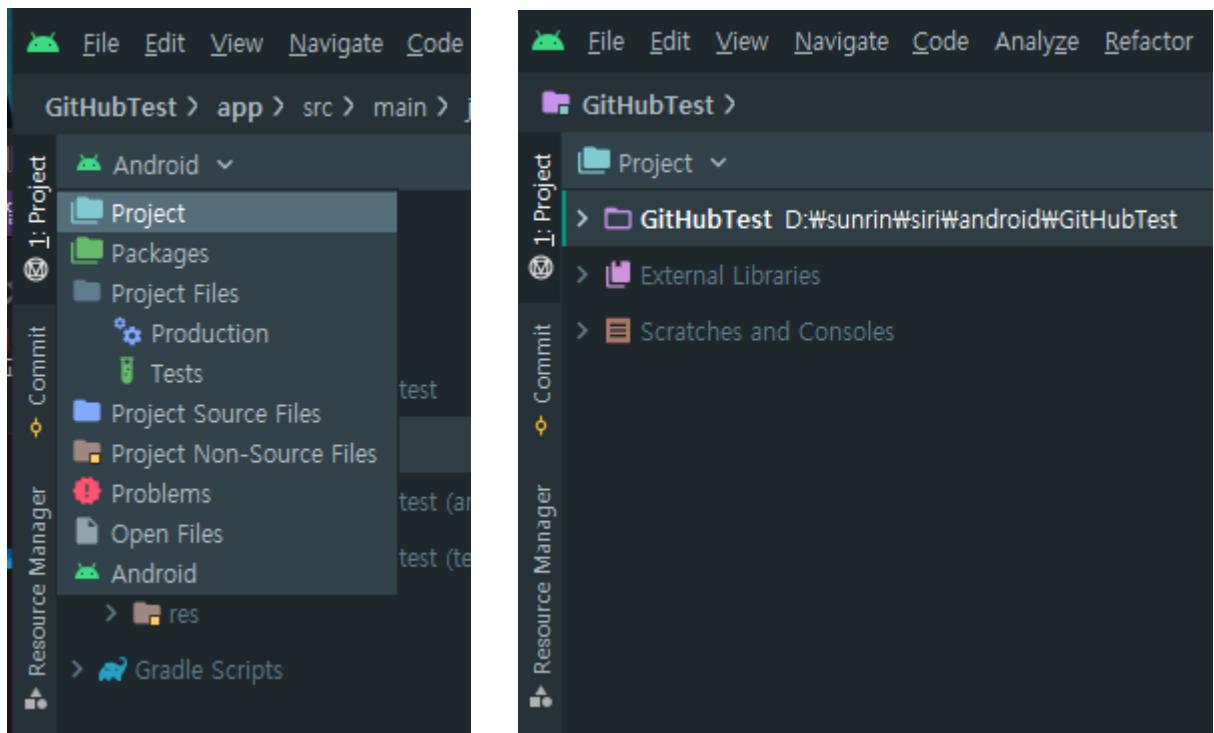
# 깃허브 세팅하기



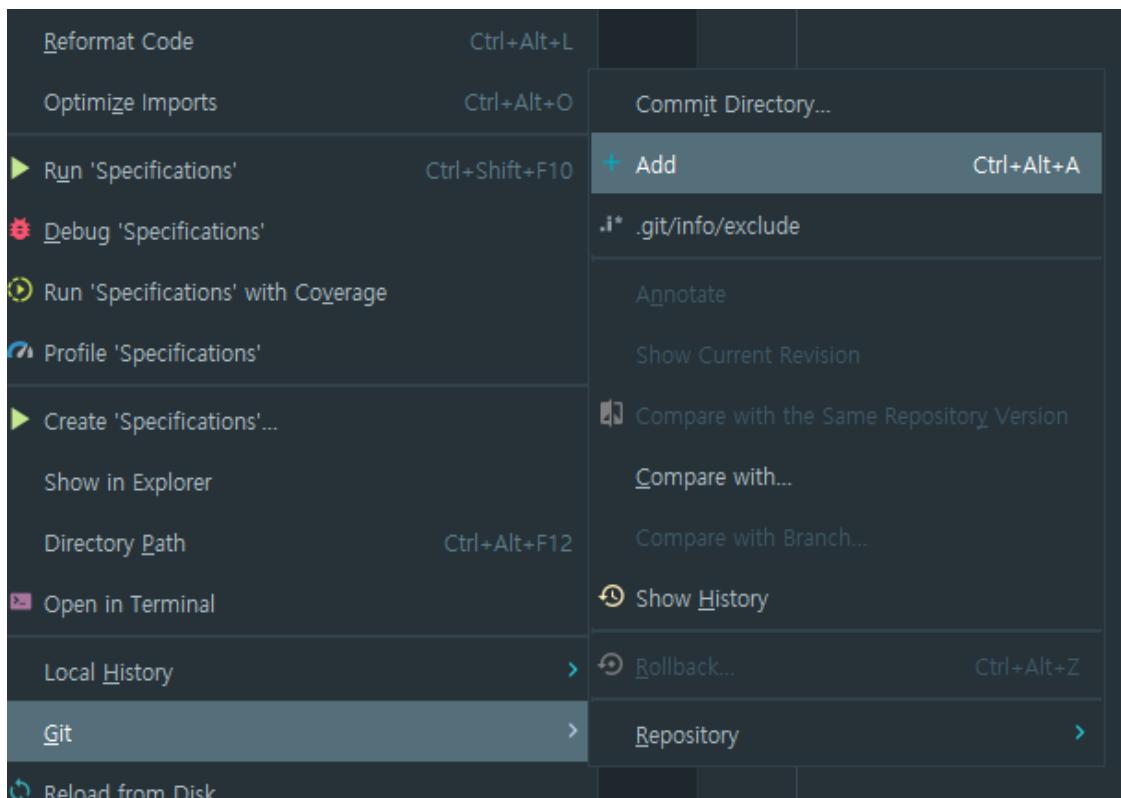
VCS > Enable Version Control..



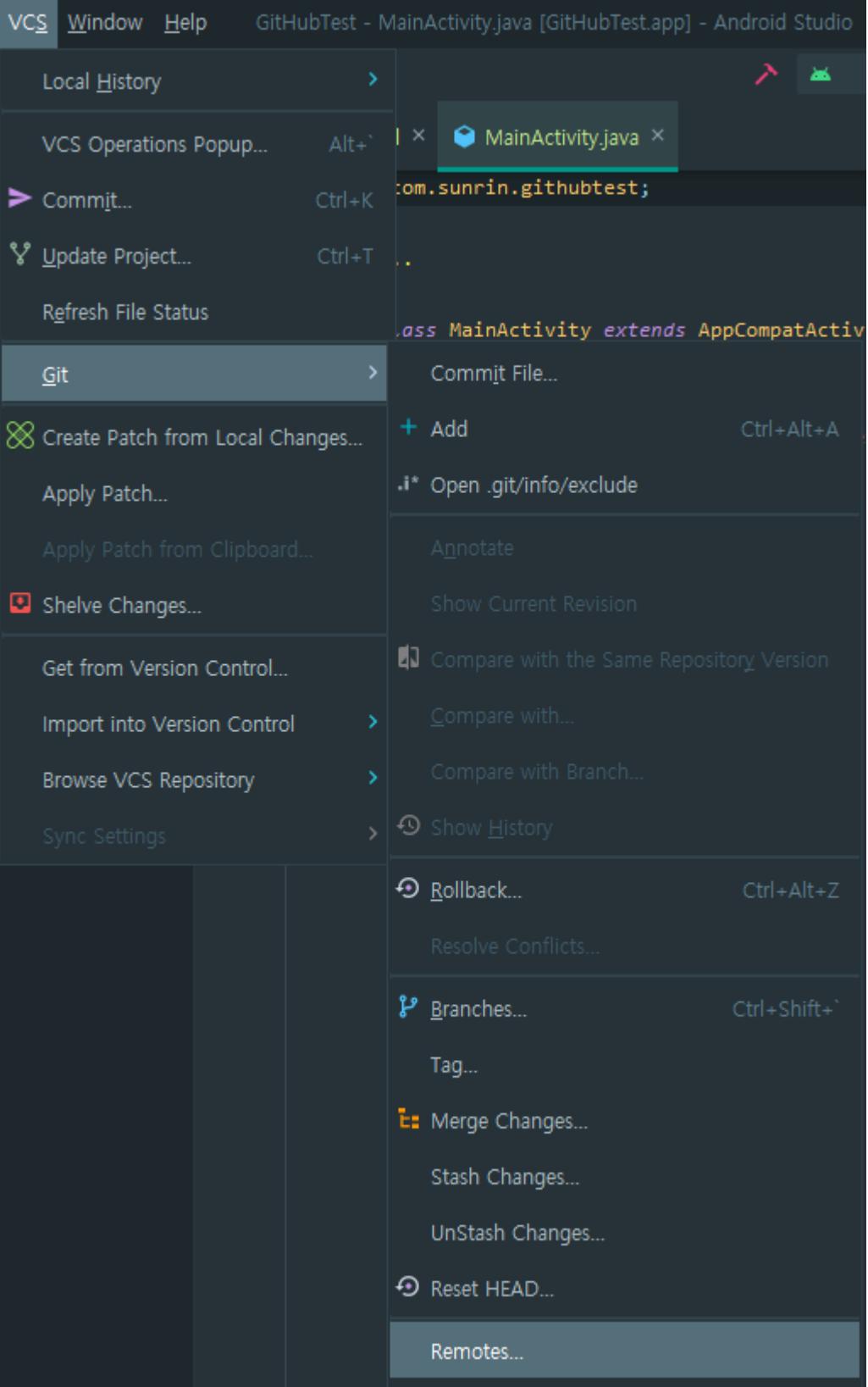
OK



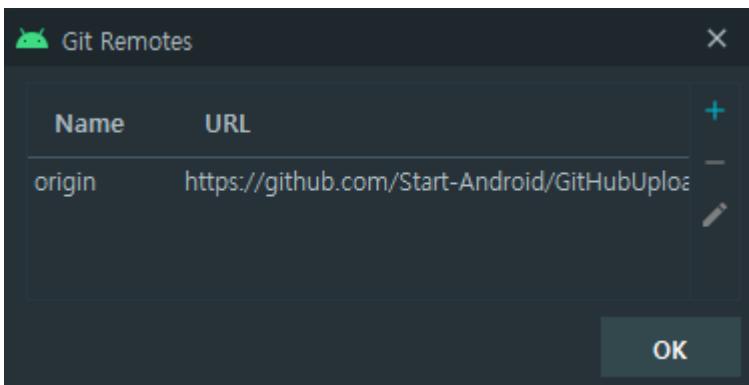
프로젝트 창으로 바꾼 후



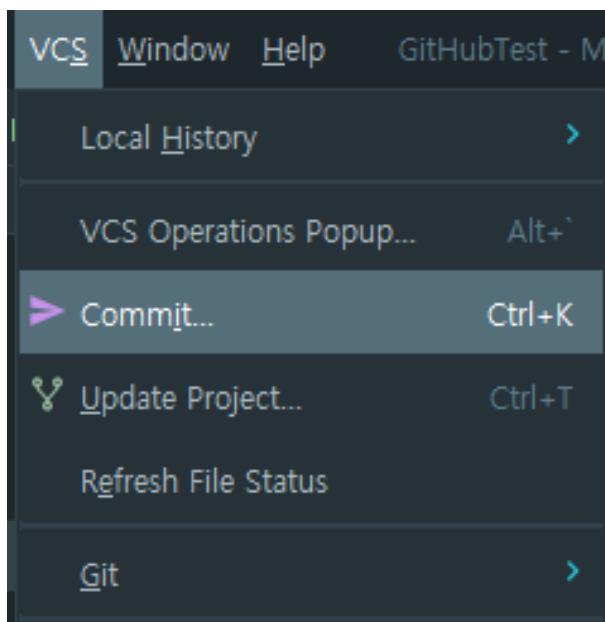
우클릭 > Git > Add



VCS > Git > Remotes..



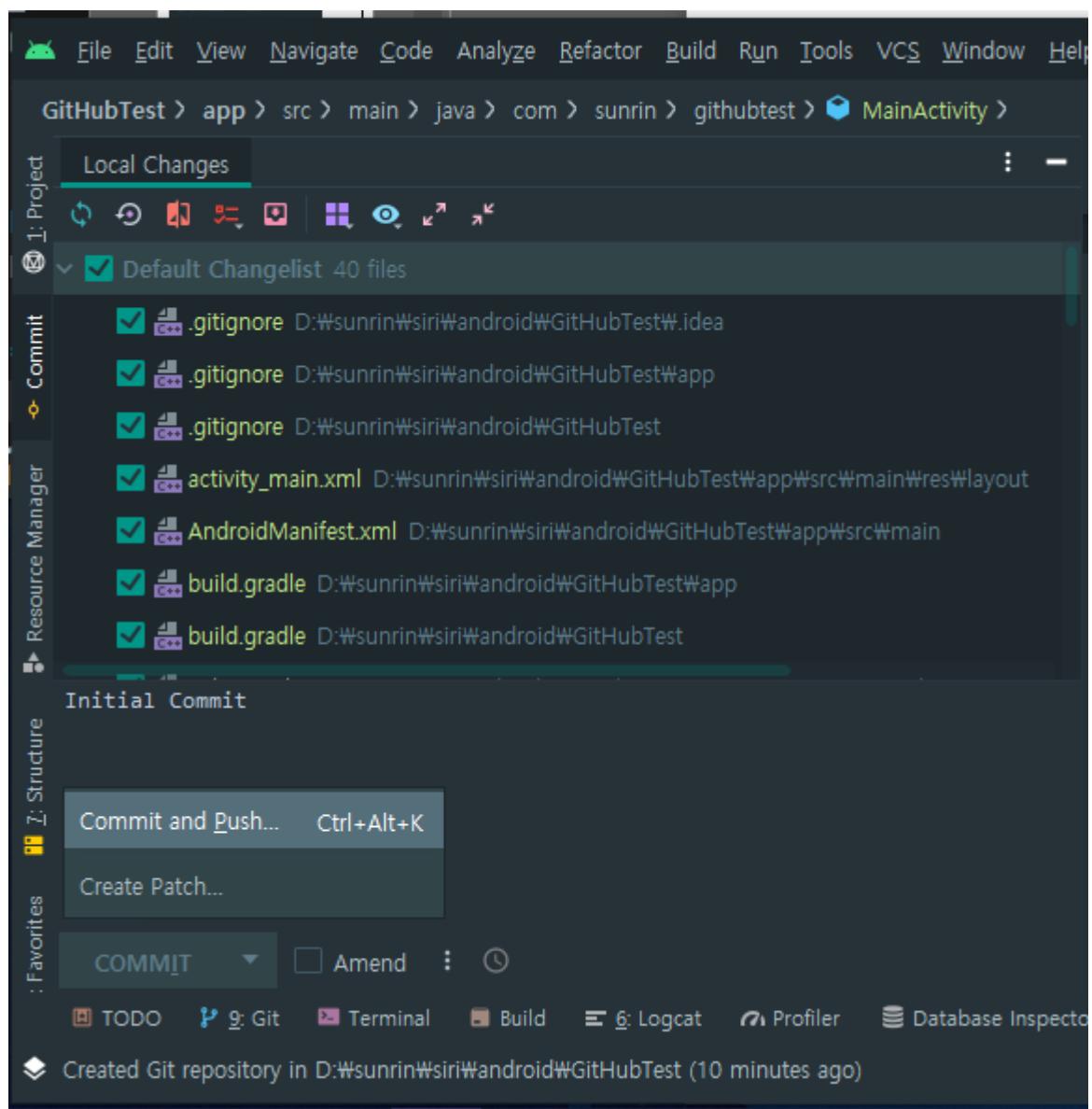
+ > 복사해 두었던 링크 붙여넣기 > OK



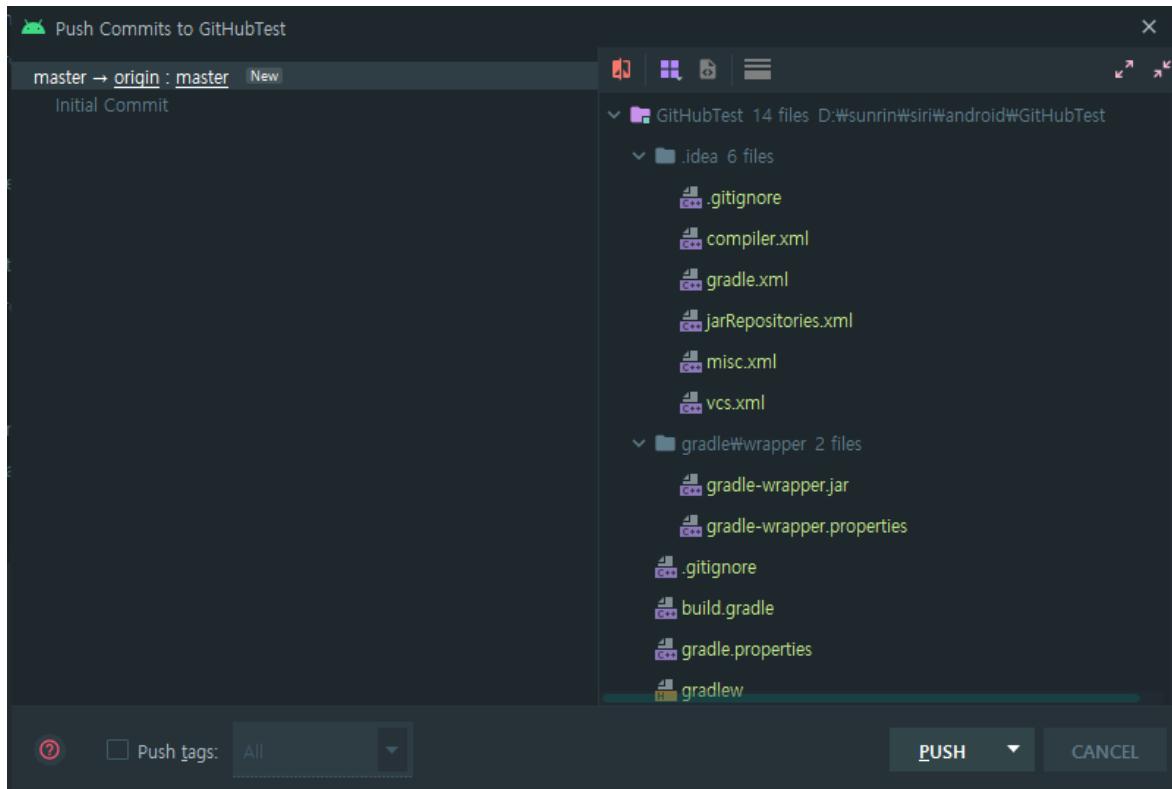
VCS > Commit

UserName과 이메일 입력하기

## 메시지 입력 후 Commit and Push



# Push



Done!

The screenshot shows the GitHub repository page for "Start-Android / GitHubUploadTest". The top navigation bar includes "Pull requests", "Issues", "Marketplace", and "Explore". The repository name is "Start-Android / GitHubUploadTest" with 0 stars, 0 forks, and 0 releases. The "Code" tab is selected. The main content area shows a single commit titled "XxCtrlZx Initial Commit" made by "46a11ef" 2 minutes ago, which contains 1 commit. The commit details show files like ".idea", "app", "gradle/wrapper", ".gitignore", "build.gradle", "gradle.properties", "gradlew", "gradlew.bat", and "settings.gradle" all as "Initial Commit" 2 minutes ago. To the right, there are sections for "About" (깃허브 업로드 테스트), "Releases" (No releases published), "Packages" (No packages published), and "Languages" (Java 100.0%). A note at the bottom encourages adding a README.



# 부록

Android Studio를 사용하면서 알아 두면 유용한 정보를  
정리해 둔 곳입니다.

by : 박찬웅, 이예희, 이종우

# X-1

# manifest

## manifest

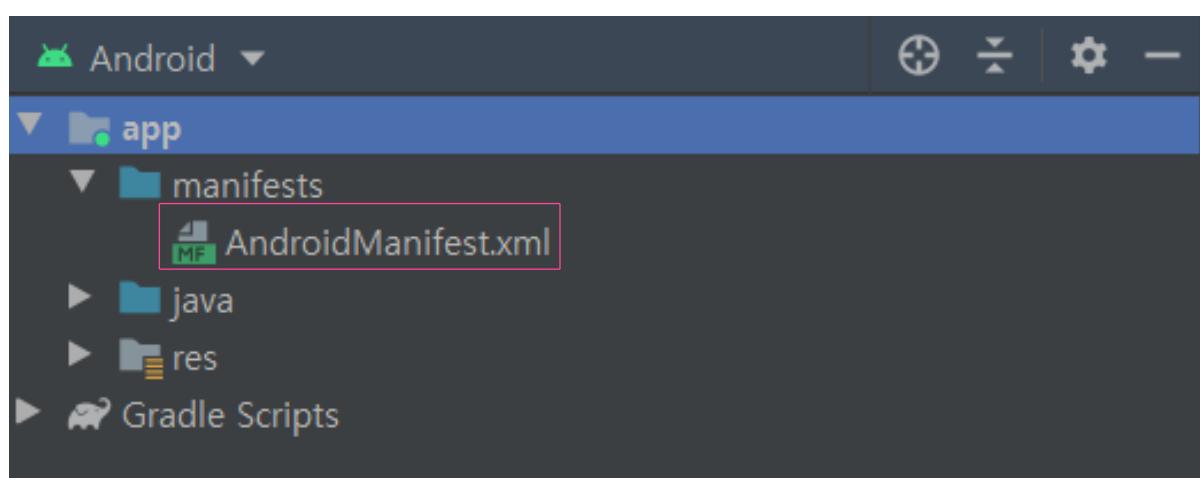
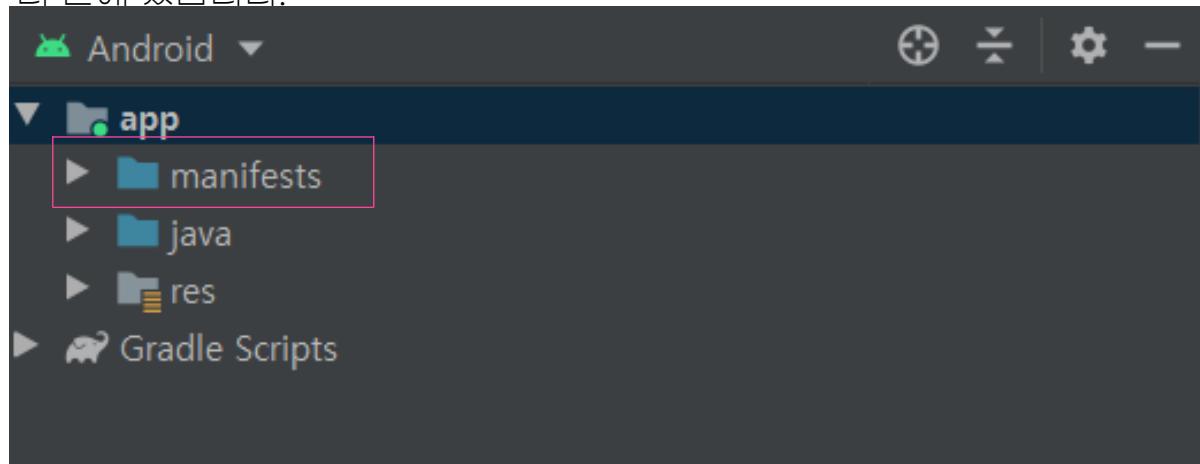
### 앱 매니페스트(App Manifest)란?

모든 애플리케이션에는 `AndroidManifest.xml`이란 파일이 있어야 합니다.  
매니페스트 파일은 Android 시스템에 대한 앱 관련 필수 정보가 들어 있습니다.  
매니페스트가 하는 일은 대표적으로

- 애플리케이션에 대한 Java 패키지의 이름 선언하기
- 애플리케이션의 구성 요소 설명하기
- 애플리케이션의 권한 설정하기
- 애플리케이션에서 요구하는 하드웨어와 소프트웨어의 특징 등이 있습니다.

### AndroidManifest.xml의 위치

`AndroidManifest.xml`은 아래 사진처럼 ‘app’ 안에 속해 있는 ‘manifests’ 폴더 안에 있습니다.



# AndroidManifest.xml의 기본 구성 요소

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

<?xml version="1.0" encoding="utf-8?> (1번 줄)

xml의 버전과 인코딩 형식을 지정합니다.

**<application> (5~19번 줄)**

어플리케이션을 선언합니다. 여기에는 각각의 어플리케이션 컴포넌트들을 선언하고 있는 하위 요소들을 포함되어 있습니다.

**android:allowBackup (6번 줄)**

백업 기능의 사용 여부를 표시합니다.

**android:icon (7번 줄)**

어플리케이션 전체와 각각의 컴포넌트들을 위한 아이콘을 나타내며, 이미지 파일을 포함하고 있는 참조로써 반드시 세팅되어야 합니다.

**android:label (8번 줄)**

어플리케이션과 각각의 컴포넌트들을 위한 기본 라벨을 나타냅니다.

**android:supportsRtl (10번 줄)**

어플리케이션이 RTL(Right-To-Left) 레이아웃 기능을 지원하는지 여부를 나타냅니다.

**android:theme (11번 줄)**

어플리케이션 안의 모든 기능들을 위한 기본 테마를 정의한 스타일을 나타냅니다.

**<activity> (12~18번 줄)**

어플리케이션의 시각적인 사용자 인터페이스 부분을 실현하는 activity를 선언합니다.

**<intent-filter>와 <action> (13~17번 줄)**

각각의 객체들이 필터기능을 사용할 수 있도록 지정합니다.

# X-2

# build.gradle

## build.gradle

### 빌드 자동화란?

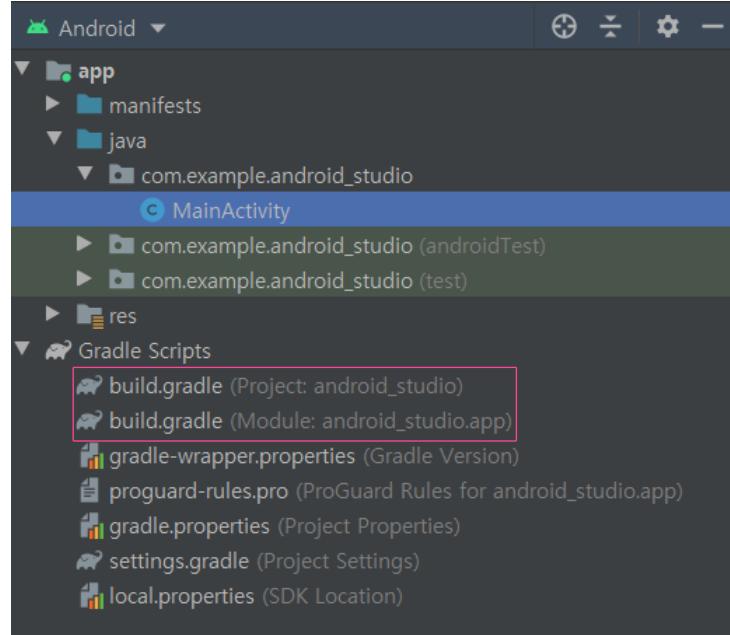
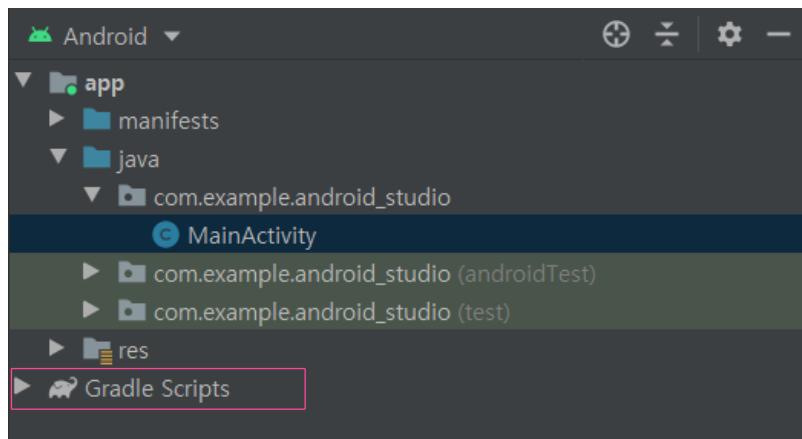
build.gradle은 빌드 자동화를 위한 툴입니다.

프로젝트는 개발 전 미리 빌드 환경 설정을 하는데 이 때 컴파일과 링킹 등을 통해 실행 파일, java, jar과 같이 다양한 라이브러리 파일을 생성합니다.

바로 이런 작업들을 자동화하는 것이 빌드 자동화입니다.

### build.gradle의 위치

build.gradle은 아래 사진과 같이 ‘Gradle Scripts’안에 속해 있고, project 버전과 module 버전으로 나뉘어져 있습니다. 이 중 module 버전에 대해 자세히 알아봅시다.



# build.gradle의 기본 구성 요소

```
1  plugins {
2      id 'com.android.application'
3  }
4  android {
5      compileSdkVersion 30
6      buildToolsVersion "30.0.0"
7      defaultConfig {
8          applicationId "com.example.android_studio"
9          minSdkVersion 28
10         targetSdkVersion 30
11         versionCode 1
12         versionName "1.0"
13         testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
14     }
15     buildTypes {
16         release {
17             minifyEnabled false
18             proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
19         }
20     }
21     compileOptions {
22         sourceCompatibility JavaVersion.VERSION_1_8
23         targetCompatibility JavaVersion.VERSION_1_8
24     }
25 }
26 dependencies {
27     implementation 'androidx.appcompat:appcompat:1.2.0'
28     implementation 'com.google.android.material:material:1.2.1'
29     implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
30     testImplementation 'junit:junit:4.+'
31     androidTestImplementation 'androidx.test.ext:junit:1.1.2'
32     androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'
33 }
```

## **id 'com.android.application' (2번 줄)**

안드로이드 플러그인 사용을 gradle에 적용합니다.

## **android{} (4~25번 줄)**

이 안의 내용은 안드로이드와 관련된 빌드 설정입니다.

## **compileSdkVersion (5번 줄)**

앱 컴파일 시 사용할 SDK 버전을 지정합니다.

## **buildToolsVersion (6번 줄)**

사용할 빌드 툴의 버전을 명시합니다.

## **defaultConfig{} (7~14번 줄)**

AndroidManifest.xml에서 사용하는 설정을 적용합니다.

## **buildTypes{} (15~20번 줄)**

다양한 빌드 타입 종류를 지정합니다.

## **compileOptions{} (21~24번 줄)**

컴파일 시 쓰이는 java의 버전을 명시합니다.

## **dependencies{} (26~33번 줄)**

라이브러리 추가 시 이곳에 작성합니다.

# X-3 Values

## Colors

app>res>values>colors.xml

colors.xml에서 색상 리소스를 관리합니다.

layout의 xml에서 호출이 가능합니다.

## Strings

app>res>values>strings.xml

<resource>요소는 xml 리소스를 추가하는데 있어서 필수로 들어가는 루트요소입니다.

그렇기에 반드시 추가하는 리소스는 내부에 작성되어야 합니다.

<string>요소는 단일 문자열 리소스를 추가할 때 사용하는 요소입니다.

반드시 name 속성을 지정해야 하며 name 속성값은 문자열의 이름과 다른 곳에서 참조해야 할 리소스 ID로 사용됩니다.

## Styles

app>res>values>styles.xml

<resource>요소는 xml 리소스 파일의 루트요소로 모든 리소스 정의는

<resource>의 하위요소로 정의해야 합니다.(strings.xml과 동일)

<style>요소는 스타일 리소스를 추가할 때 사용하는 요소로 name 속성을 지정하여 다른 곳에서 해당 스타일 리소스를 참조할 수 있도록 합니다.

<item>요소는 스타일을 지정하고자 하는 속성에 따라 문자열, 색상, 다른 리소스에 대한 참조를 지정합니다.

## 다크 모드란?

다크 모드는 휴대폰의 소프트웨어에서 밝은 테마 대신에 어두운 테마를 지원하는 것을 뜻합니다. 안드로이드(삼성) 휴대폰 기준으로 파이(9.0) 버전부터 ‘야간 모드’라는 이름으로 지원하기 시작했으며, 현재는 이 글의 제목처럼 ‘다크 모드’라는 이름으로 사용 가능합니다.

여러분의 휴대폰에서 설정 → 디스플레이 → 다크 모드에 들어가 ‘지금 켜기’를 통해 해당 기능을 활성화하면, 화면의 색 계열이 흰색에서 검은색이 되는 것을 확인할 수 있습니다.

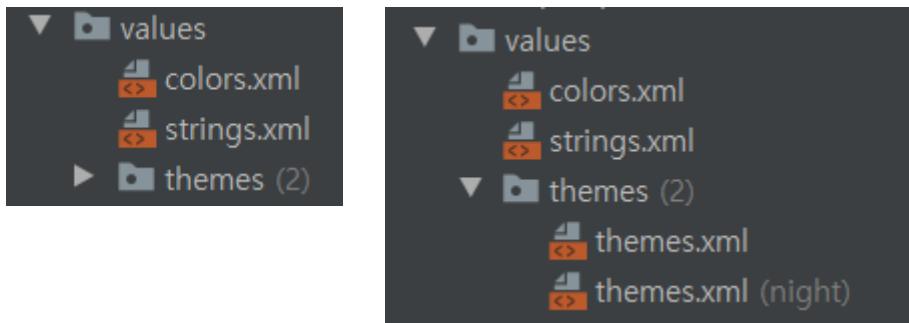
기존에는 일반 모드(흰색)에만 맞춰 디자인을 하면 되었지만, Android Studio 버전 4.1부터 다크 모드를 기본 지원하기 시작하면서 앞으로는 다크 모드, 일반 모드 모두를 고려해 코딩 및 어플리케이션 디자인을 해야 할 것입니다.



적용 전      적용 후

# 다크 모드 적용하기

## 1. themes에서 적용



위 사진처럼 Android Studio 버전 4.1 이상에서 프로젝트를 생성할 경우, themes 폴더 안에 themes.xml이 2개가 생성되어 있는 것을 확인할 수 있습니다.

이 중 (night)라고 되어 있는 것이 다크 모드가 적용되었을 때 적용되는 themes라고 보면 되겠습니다.

이 2개의 themes 파일을 비교하고 앱을 실행해 보면서, 잘 맞는 색상을 선택하면 됩니다.

## 2. colors에서 적용

아래 사진을 참고해서 기존의 colors.xml을 지우지 않고 추가로 colors.xml을 만들어줍니다.



그럼, 이런 식으로 다크 모드에서 실행이 될 colors.xml이 생성이 될 것입니다.

이제 기존에 있었던 colors.xml에 등록되어 있는 색상을 참고해서 다크 모드에서 사용할 색상을 지정해 주면 됩니다.

## 주의할 점

- themes와 colors 둘 중 한 가지 방법으로만 다크 모드를 적용하는 것을 추천합니다. 만약 두 방법을 모두 사용하게 된다면, 색상 관련 코드가 꼬여버려 디버그가 어려워질 수 있습니다.
- 다크 모드, 일반 모드 모두 잘 적용되었는지 꼭 확인해야 합니다. 그렇지 않으면, 아래와 같은 불상사가 발생할 수 있습니다.



적용 전      적용 후

## 다크 모드 적용 해제하기

다크 모드를 적용하게 되면, 개발하는 어플리케이션에 따라 디자인과 코딩, 두 가지 측면에서 모두 번거로워질 수 있습니다.

따라서, 기존처럼 다크 모드를 사용하지 않고 어플리케이션을 개발하고 싶을 때는 themes.xml의 DayNight을 Light로 변경하면 됩니다.

→ themes.xml(night)을 지우거나 두 개의 themes.xml에 동일하게 적용해야 합니다.

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Theme.Intentview" parent="Theme.MaterialComponents.DayNight.DarkActionBar">
        <!-- Primary brand color. -->
        <item name="colorPrimary">@color/purple_500</item>
        <item name="colorPrimaryVariant">@color/purple_700</item>
        <item name="colorOnPrimary">@color/white</item>
        <!-- Secondary brand color. -->
        <item name="colorSecondary">@color/teal_200</item>
        <item name="colorSecondaryVariant">@color/teal_700</item>
        <item name="colorOnSecondary">@color/black</item>
        <!-- Status bar color. -->
        <item name="android:statusBarColor" tools:targetApi="l">?attr/colorPrimaryVariant</item>
        <!-- Customize your theme here. -->
    </style>
</resources>
```

# X-5

# 스마트폰 연결

## 에뮬레이터

안드로이드 스튜디오에서는 개발자의 편의를 위해 스마트폰 없이도 어플리케이션을 실행시켜볼 수 있는 에뮬레이터를 무료로 제공합니다.

그래서 .apk를 실행할 수 없는 데스크탑 환경에서, 심지어는 Mac에서도 안드로이드 어플리케이션 개발이 가능하죠.

그런데, 제목을 보면 <스마트폰 연결> 이라고 나와있죠?

왜일까요?

안드로이드 스튜디오의 단점 중 하나이자, 어플리케이션 개발을 하기 쉽게 만드는 원인 중 하나가 바로 이 에뮬레이터이기 때문입니다!

하지만 우리는 늘 그렇듯이 답을 찾아야겠죠?

그래서 나온 채신기술!

무려 자신의 스마트폰을 컴퓨터에 연결해 어플리케이션을 바로 실행해볼 수 있습니다!

어떻게 하냐구요?

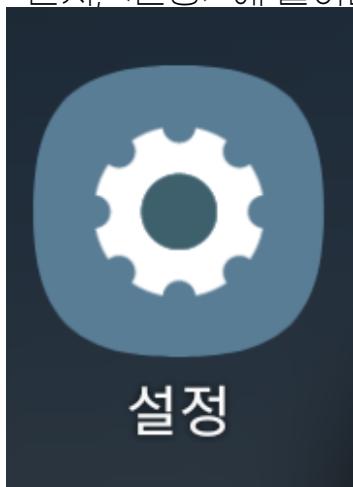
지금부터 따라오시죠!

## 스마트폰 연결

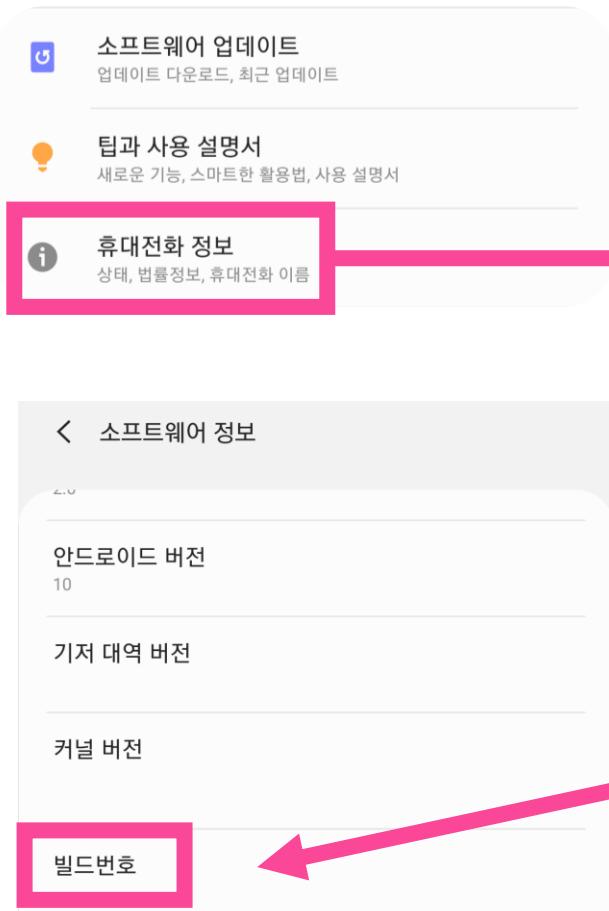
### 개발자 모드 활성화

스마트폰을 컴퓨터와 연결하기 전, 개발자 모드를 활성화해줍니다.

먼저, <설정>에 들어갑니다.



그 다음, 제일 아래로 내려가서 <휴대전화 정보>를 클릭하고, <소프트웨어 정보>로 들어갑니다.



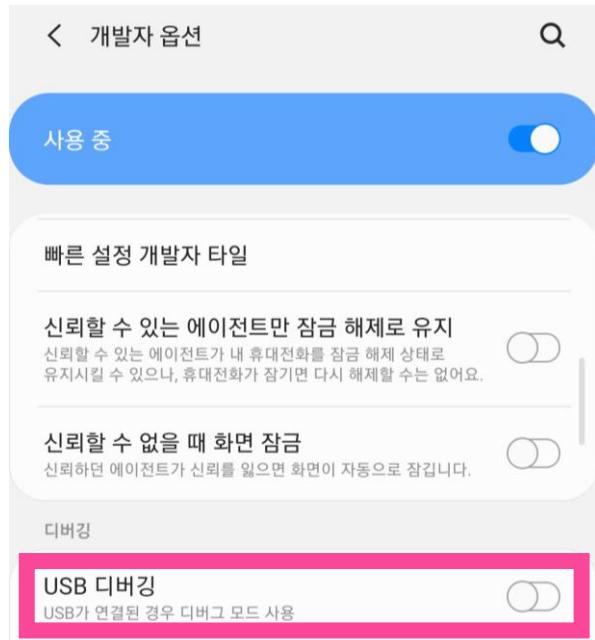
그 다음, <빌드 번호>를 “개발자 모드를 켰습니다.”라는 Toast 메시지가 뜰 때까지 연타해주세요.

그럼 아래 사진처럼 <휴대전화 정보> 아래 <개발자 옵션>이 생길 것입니다.



# USB 디버깅 활성화

<개발자 옵션>에 들어가서, <USB 디버깅>을 활성화합니다.



- 개발자 옵션을 끌 경우, <개발자 옵션> 메뉴가 사라지므로, 다시 사용하려면 앞의 과정을 다시 거쳐야 합니다.
- <USB 디버깅>을 켤 경우, 휴대폰이 보안에 상대적으로 취약해질 수 있으므로 보안이 걱정된다면 항상 켜 두지는 않는 것을 추천합니다.

## 컴퓨터에 연결

이제 컴퓨터에 해당 휴대폰을 연결해 볼시다.

“USB 디버깅을 허용하시겠습니까?”라는 Alert 창이 나오면 “허용”을 눌러줍시다.

USB 디버깅을 허용하시겠습니까?

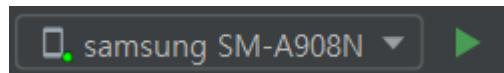
컴퓨터 RSA 키 지문:

이 컴퓨터에서 항상 허용

취소

허용

휴대폰이 제대로 인식되었다면 아래와 같이 본인이 연결한 휴대폰의 모델명이 뜰 것입니다.



이 상태로 옆의 실행(Run) 버튼을 누르면, 어플리케이션이 여러분의 휴대폰에 설치가 되고 자동으로 실행이 될 것입니다.

## 연결이 안 된다면?

혹시라도 위 화면에 휴대폰이 뜨지 않거나, “USB 디버깅을 허용하시겠습니까?” Alert 창이 뜨지 않는다면 케이블에 문제가 있거나 컴퓨터에 USB 드라이버가 깔려 있지 않을 확률이 높으니, 케이블을 교체하거나 USB 드라이버를 재설치하세요.

만약 휴대폰은 뜨는데 “Physical Device”나 에러 표시가 뜬다면 Android Studio에서 개발한 프로그램의 버전이 본인 휴대폰과 맞지 않는 경우일 것입니다. 이 경우는 Android Studio에서 프로젝트를 생성할 때 Minimum SDK를 낮추어야 합니다.

# 부록

출처 및 참고자료

# 출처 및 참고자료

## 글꼴

G마켓 산스

<http://company.gmarket.co.kr/company/about/company/company--font.asp>

## 소스코드

본 교재에서 제공하는 소스코드는 아래 링크에서 다운로드 받을 수 있습니다.

<https://github.com/Start-Android/>

## 동영상 강의

본 교재와 관련된 동영상 강의는 아래 YouTube 링크에서 시청 가능합니다.

[https://www.youtube.com/channel/UCQPD8Bu1JIA\\_FoVlq1P4rOw](https://www.youtube.com/channel/UCQPD8Bu1JIA_FoVlq1P4rOw)