

# 시스템 분석 설계 (포트폴리오)

20202269 노연건 (컴퓨터정보공학과)

# 목차

# 안드로이드 개요

# 안드로이드 앱 개발 환경

# 안드로이드 앱 개요 [안드로이드 스튜디오]

# 안드로이드 앱 실행

# + 추가 공부

# 작업 물(로그인&회원가입 화면 구성)

# 안드로이드 개요

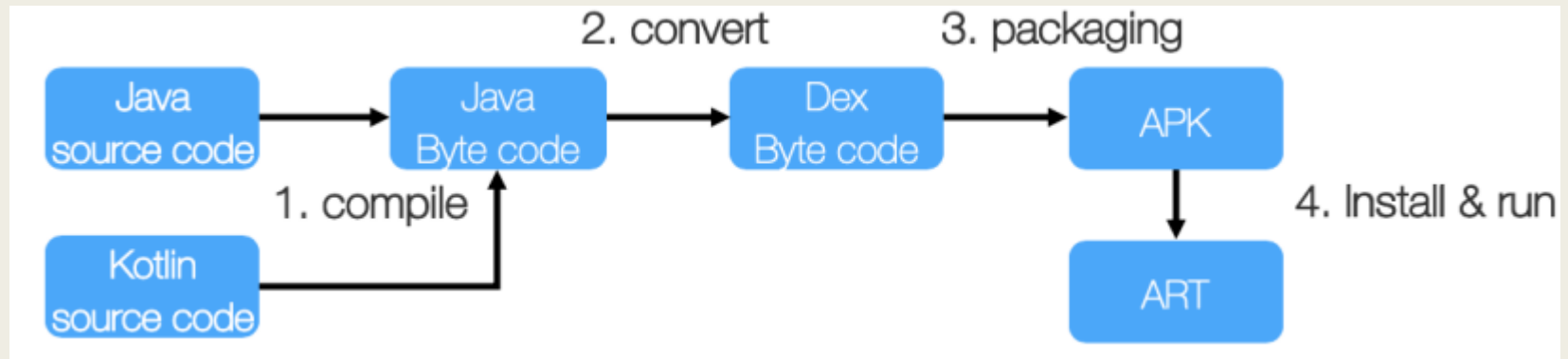
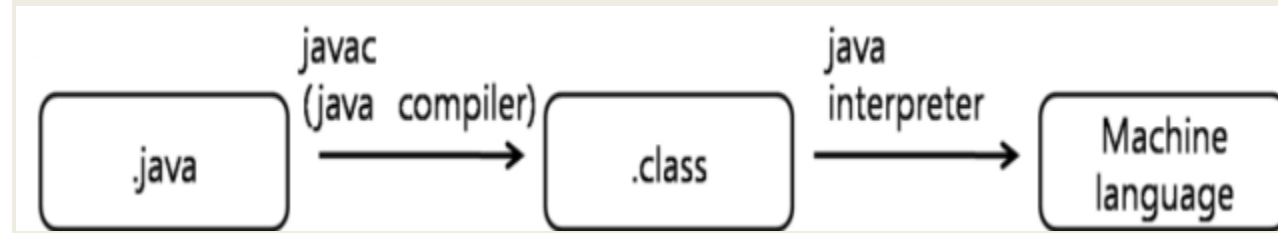
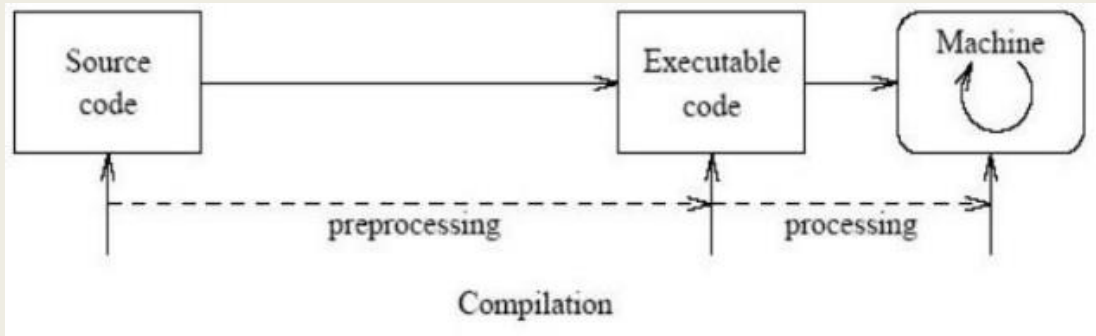
# 안드로이드

- 리눅스 커널을 기반으로 하는 휴대전화 등에 사용하는 운영체제의 한 종류
- 휴대전화를 비롯한 휴대용 장치를 위한 운영 체제와 미들웨어 그리고 핵심 어플리케이션을 포함하는 소프트웨어 집합
- 앱 개발 언어 : Java, Kotlin, C, C++
- 오픈 소스 기반 (AOSP : Android Open Source Project)
  - ☞ 커널
  - ☞ IDE(Integrated Development Environment) : [ 안드로이드 스튜디오 ]
  - ☞ SDK(Software Development Kit)
  - ☞ AVD(Android Virtual Device)
  - ☞ 외부 앱의 기능 사용 : [ 인텐트(Intent)객체 사용 ]
- Android Auto / Android TV / Wear OS / Android Things 등



# 안드로이드

- 컴파일된바이트코드를구동할수있는런타임라이브러리 (달빅가상머신 (DalvikVirtualMachine)혹은ART(AndroidRuntime))를 제공



# 안드로이드의 역사

이미지	코드명	버전	API 레벨	발표 일자
	알파(Alpha)	1.0	1	2008년 9월
	베타(Beta)	1.1	2	2009년 2월
	컵케이크(Cupcake)	1.5	3	2009년 4월
	도넛(Donut)	1.6	4	2009년 9월
	아클레어(Eclair)	2.1	7	2010년 1월
	프로요(Froyo)	2.2	8	2010년 5월
	진저브레드(Gingerbread)	2.3	10	2010년 12월
	허니콤(Honeycomb)	3.0	11	2011년 2월
		3.1	12	2011년 5월
		3.2	13	2011년 7월
	아이스크림 샌드위치 (Ice Cream Sandwich)	4.0 4.0.3	14 15	2011년 10월 2011년 11월
	젤리빈(Jelly Bean)	4.1	16	2012년 7월
		4.2	17	2012년 12월
		4.3	18	2013년 6월

이미지	코드명	버전	API 레벨	발표 일자
	킷캣(KitKat)	4.4	19	2013년 10월
		4.4w	20	2014년 6월
	롤리팝(Lollipop)	5.0 5.1	21 22	2014년 11월 2015년 3월
	마시멜로(Marshmallow)	6.0	23	2015년 10월
	누가(Nougat)	7.0	24	2016년 8월
		7.1	25	2016년 10월
	오레오(Oreo)	8.0	26	2017년 8월
		8.1	27	2017년 10월
	파이(Pie)	9.0	28	2018년 8월
	Android Q	10.0	29	2019년 9월
	Android R	11.0	30	2020년 9월

# 안드로이드의 역사

## ■ Android12

- 코드네임:S(사브리나)
- Sdk버전:API레벨31
- 2021년10월 출시
- <https://developer.android.com/about/versions/12>

## ■ Android13

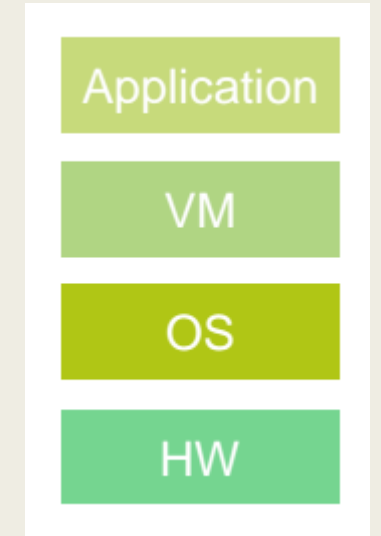
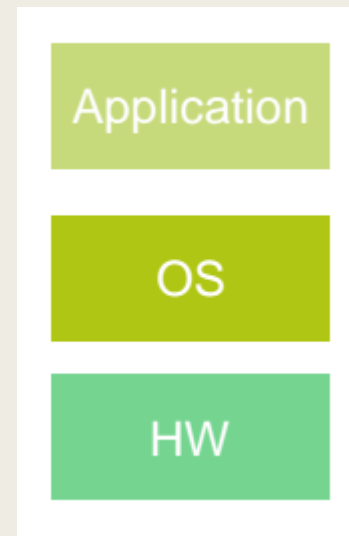
- 코드네임:T(티라미수)
- Sdk버전:API레벨32(베타이후API레벨33)
- 2022년2월10일개발자프리뷰버전/3월중베타버전/2022년하반기 정식버전출시예정
- <https://developer.android.com/about/versions/13>

# 앱



## [ 앱 ]

- 애플리케이션(Application) : PC나 스마트폰 등의 기기에서 구동되는 모든 프로그램을 일컫는 말
- 앱 : 모바일 기기에서 수행되는 애플리케이션 (App Application)

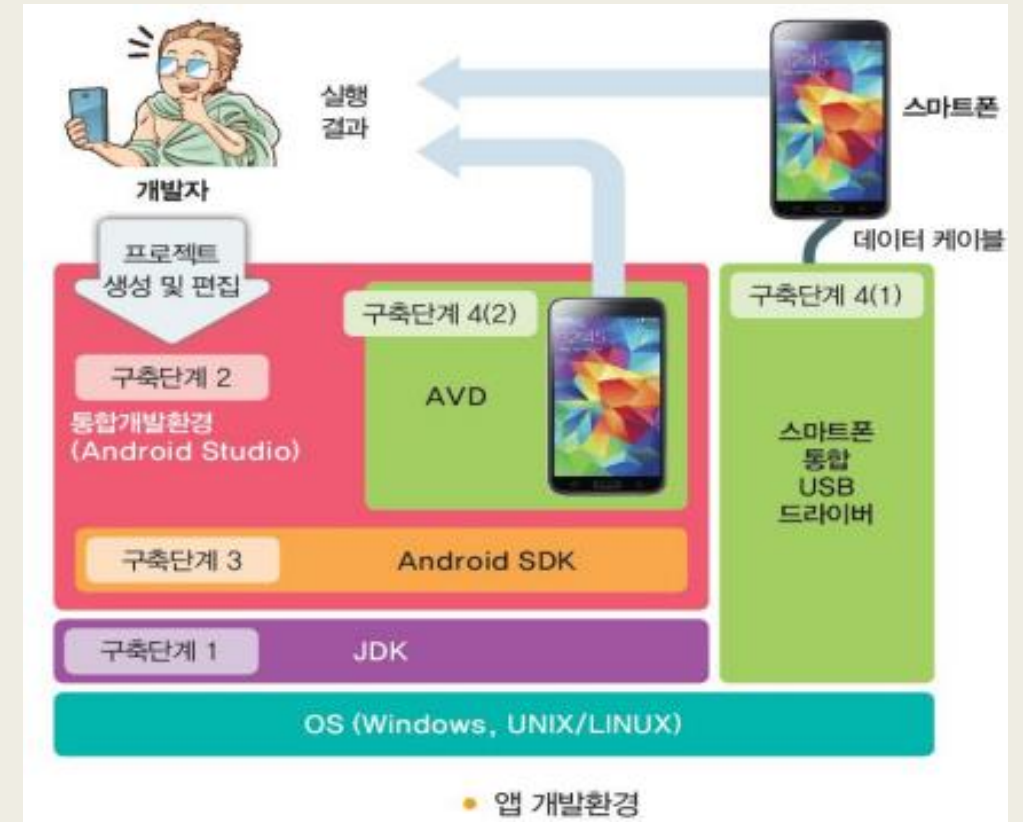




# 안드로이드 앱 개발 환경

# 안드로이드 앱 개발 환경

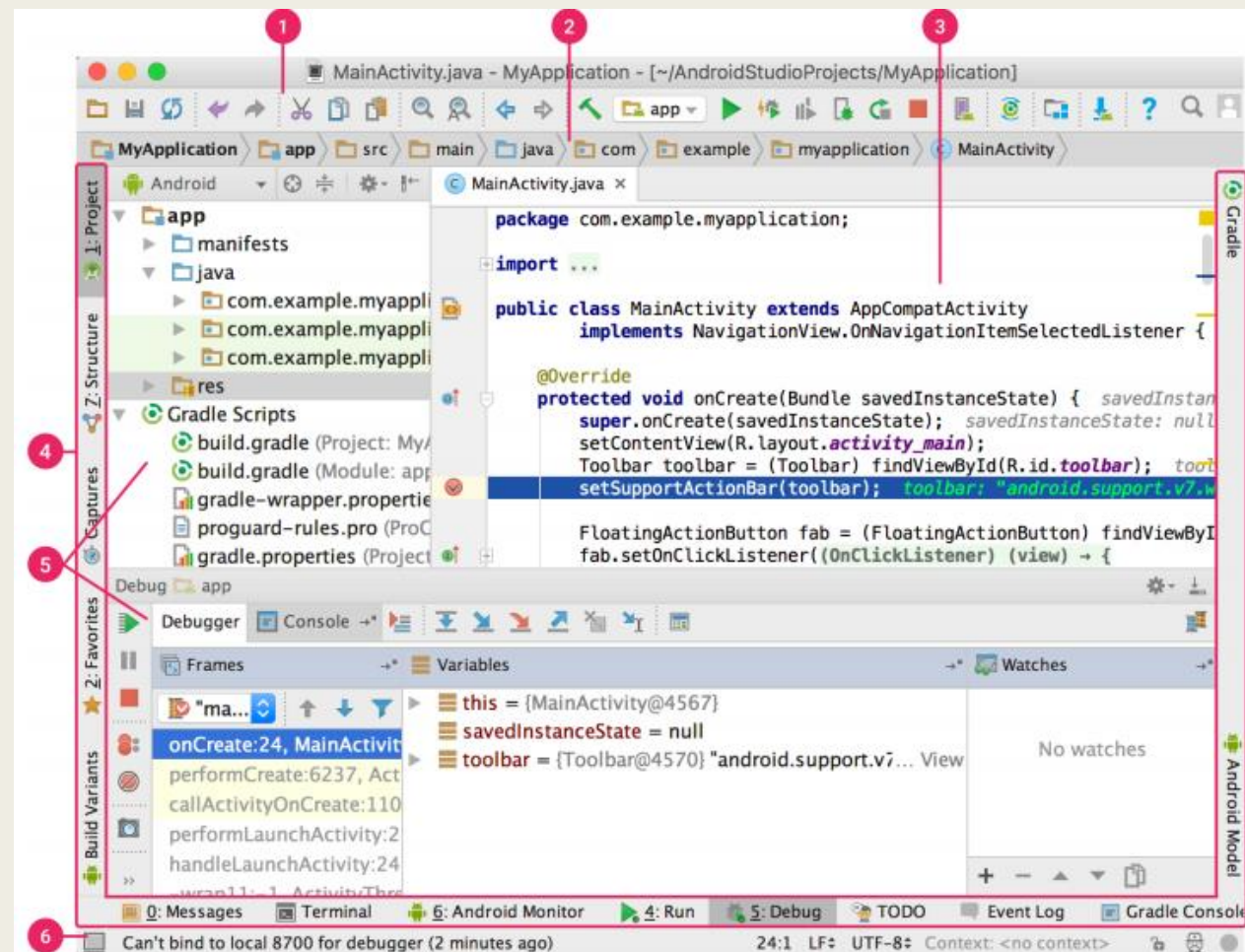
- AndroidStudio (IDE : 통합개발환경)
- JDK (Java Development Kit)
- 안드로이드 SDK (API)
- 실행
  - ☞ AVD(Android Virtual Device)
  - ☞ 안드로이드 지원 기기 사용시 통합 USB 드라이버 설치 필요



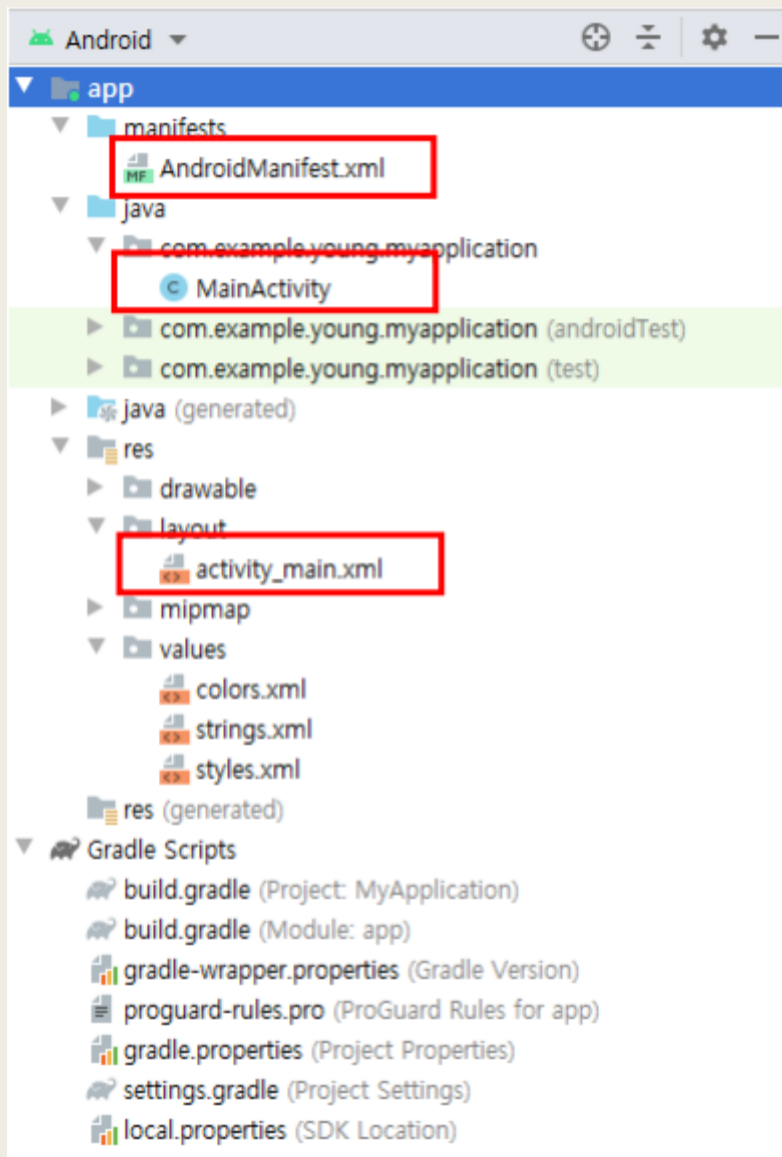
# 안드로이드 앱 개요

# 안드로이드 스튜디오 UI

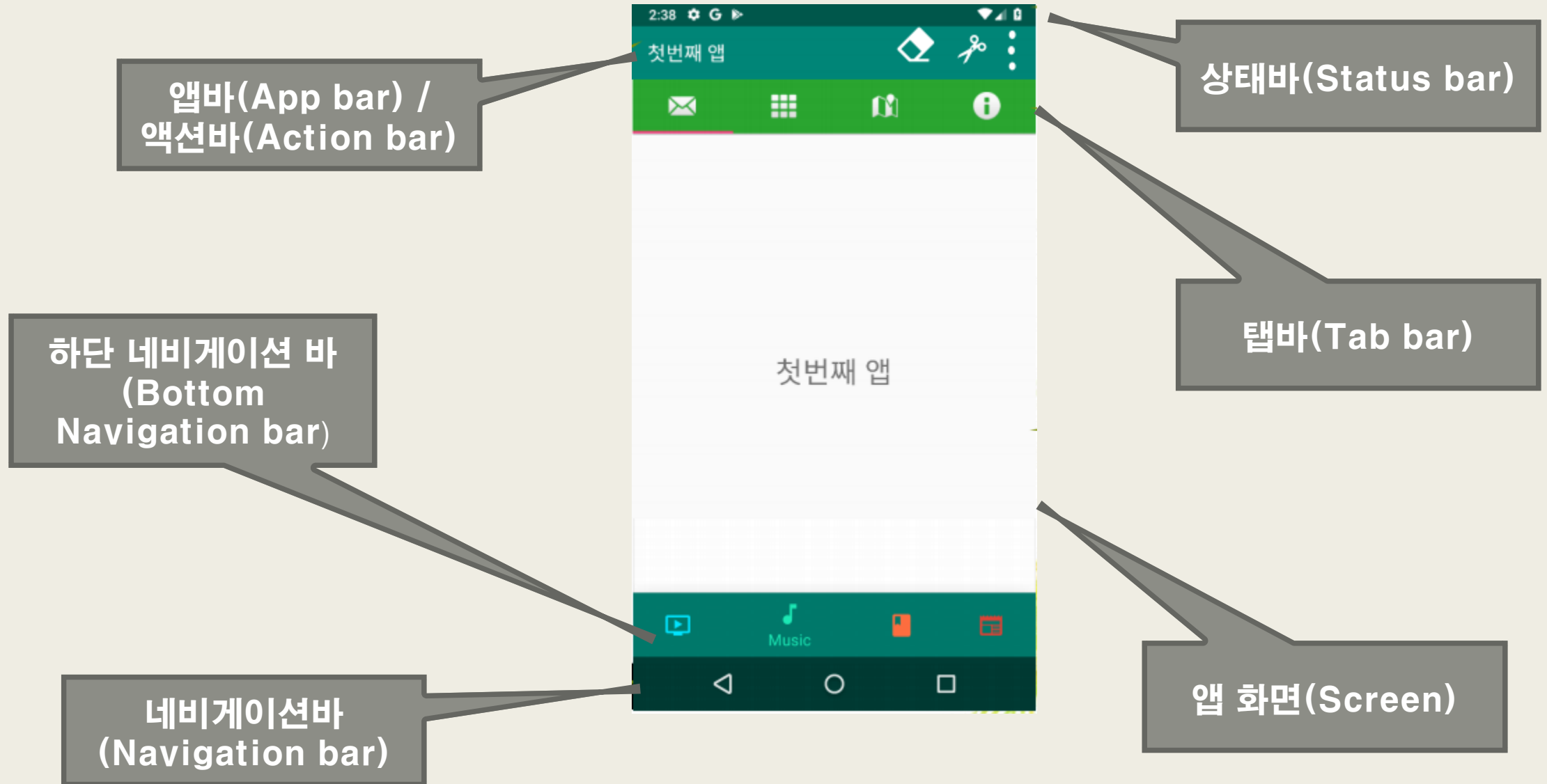
1. 툴바
2. 탐색 메뉴
3. 편집기 창
4. 도구(Tool) 창 모음
5. 도구 창
6. 상태표시줄



# 프로젝트 기본 폴더 구조

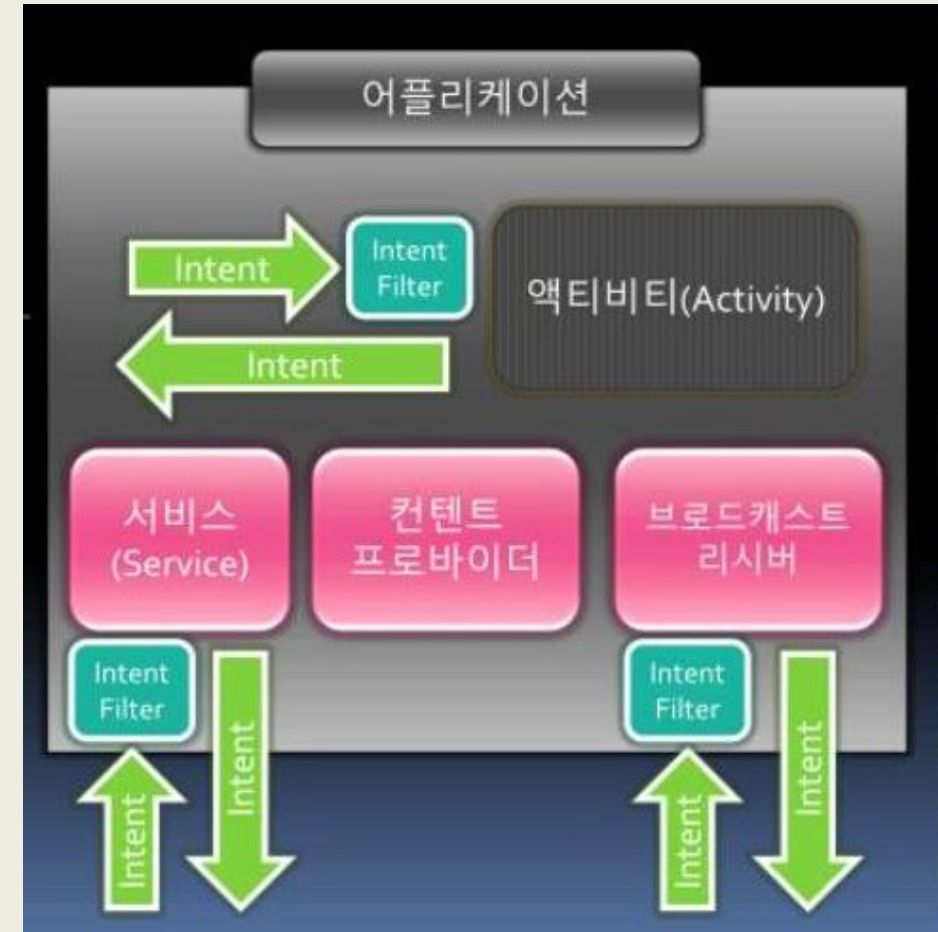


# 안드로이드 앱 화면 구조

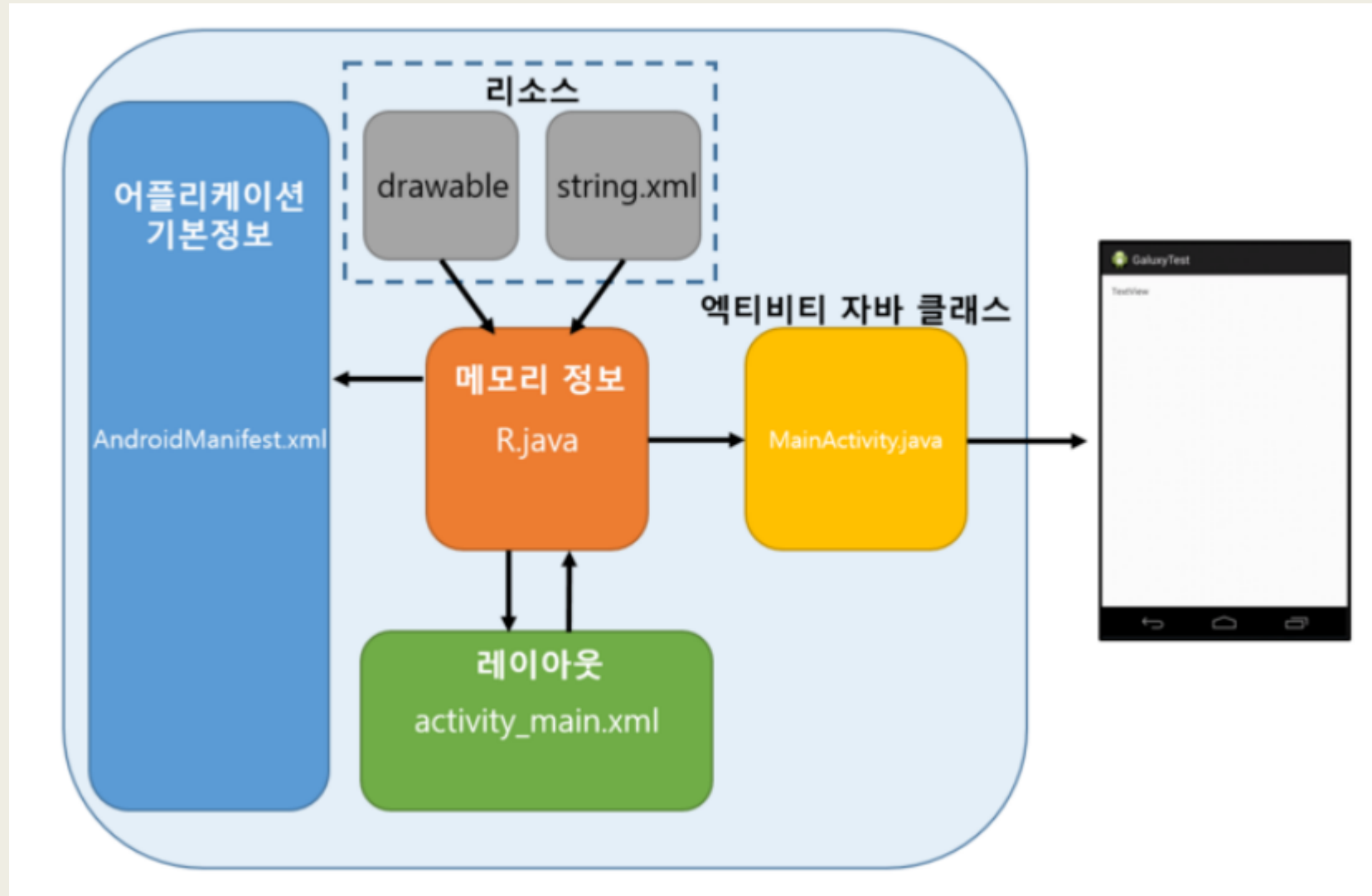


# 안드로이드 앱 구성 컴포넌트

- 액티비티 : UI가 있는 화면을 제공
- 서비스 : 백그라운드로 실행되는 컴포넌트 (예 : 음악재생)
- 브로드캐스트 리시버 : 방송을 받고 반응하는 컴포넌트 (예 : 문자수신)
- 콘텐츠 프로바이더 : 데이터를 관리하고 다른 애플리케이션에게 제공하는 컴포넌트 (예 : 연락처 정보)



# 안드로이드 앱 내부 구조





# 안드로이드 앱 구조

## ■ 액티비티 (Activity)

사용자와 상호작용 할 수 있는 화면을 제공

화면 구성요소들을 조작 및 관리

Java 클래스로 정의 : 확장자가 java인 파일로 저장

## ■ 레이아웃

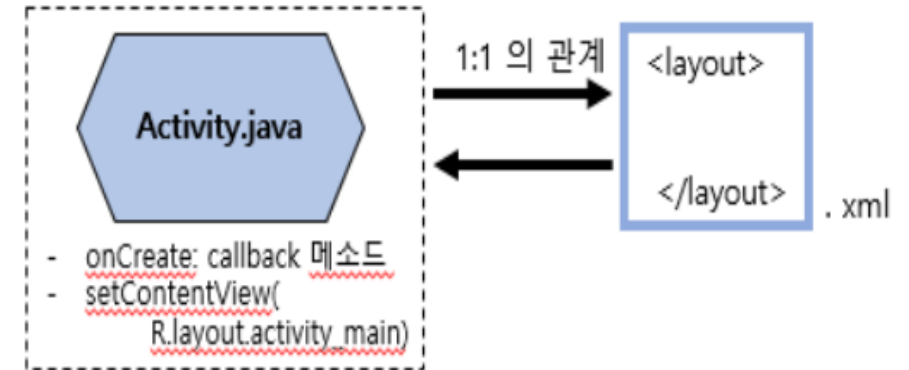
화면구성요소들을 화면에 배치하고 구성하는 방법

XML 형식으로 UI 설정 : 확장자가 xml인 파일로 저장

## ■ 레이아웃 인플레이션

XML 레이아웃에 정의된 화면구성 내용을 읽어 화면구성 요소(뷰, 위젯)들을 생성하여 화면을 구성하는 과정

setContentView() / findViewById()



# 액티비티

소스 | MainActivity.java

F

패키지 이름

01 **package** com.example.yschang.text1;

02

page39

03 **import** android.support.v7.app.AppCompatActivity;

04 **import** android.os.Bundle;

05

06 **public class** MainActivity **extends** AppCompatActivity {

07

AppCompatActivity 클래스에  
대한 패키지명과 클래스명

Bundle 클래스에 대한  
패키지명과 클래스명

MainActivity 클래스는 AppCompatActivity 클래스  
(부모 클래스)로부터 상속 받아 정의함

부모 클래스에 있는 onCreate()  
메소드를 재정의함

08 **@Override**

09 **protected void** onCreate(Bundle savedInstanceState) {

10 **super.onCreate**(savedInstanceState);

11 **setContentView**(R.layout.activity\_main);

12 }

13 }

MainActivity 자바 클래스가  
호출될 때 처음 실행되는 메소드

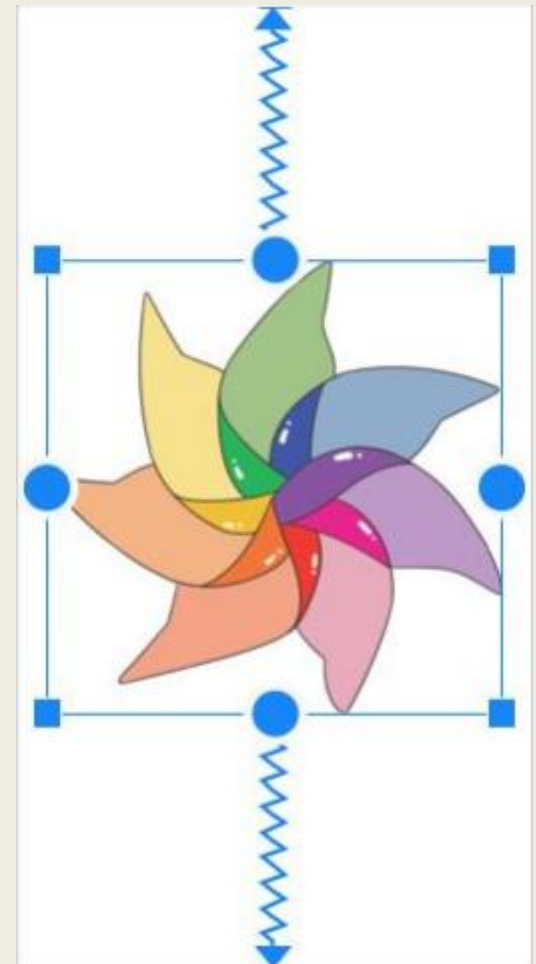
액티비티 생성

E  
activity\_main.xml에서 정의된 화면  
레이아웃을 액티비티에 출력

# Layout Xml

```
<?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"
    tools:context="ac.dongyang.pinwheel.MainActivity">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/pinwheel" />
</android.support.constraint.ConstraintLayout>
```



# 안드로이드 앱 구조

## ■ 매니페스트 파일(Manifestfile)

- Android 시스템이 앱의 코드를 실행하기 전에 확보해야 하는 앱에 대한 필수 정보를 시스템에 제공

- Java 패키지이름

- 애플리케이션구성요소 : 액티비티, 서비스, 프로 바이더, 리시버

- 권한 설정 : 인터넷, 내부 파일 시스템 접근 등

- 애플리케이션 실행 프로세스 : 인텐트 필터

- 앱에서사용하는HW/SW기능:카메라,Bluetooth 사용 등

- XML 형식으로 기술

# 매니페스트 파일

```
01 <?xml version="1.0" encoding="utf-8"?>
02 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
03     package="com.example.yschang.helloandroid">
04
05     <application
06         android:allowBackup="true"
07         android:icon="@mipmap/ic_launcher" ← A
08         android:label="@string/app_name" ← C
09         android:supportsRtl="true"
10         android:theme="@style/AppTheme" ← B
11         <activity android:name=".MainActivity" ← E
12             <intent-filter>
13                 <action android:name="android.intent.action.MAIN" />
14
15                 <category android:name="android.intent.category.LAUNCHER" />
16             </intent-filter>
17         </activity>
18     </application>
19
20 </manifest>
```

어플리케이션 아이콘

어플리케이션 라벨

화면 테마

어플리케이션을 구성하는 액티비티 자바 클래스

앱 실행 시, MainActivity 자바 클래스가 처음 실행됨

# 안드로이드 앱 구조

## ■ 리소스파일(Resources):xml형식으로기술

- layout

- values

```
<resources>  
    <string name="app_name">My Application</string>  
</resources>
```

- strings.xml:앱에서사용되는문자열들을정의

- themes.xml(안드로이드스튜디오4.10이후)

- View의모양과형식을지정하는속성모음

- colors.xml:앱에서사용되는색상들을정의

- drawable:이미지,shape,statelist등

- mipmap:icon

- menu:메뉴

- 리소스이름:자바변수명명규칙을따르고대문자, 한글,공백문자사용하면안됨



# 안드로이드 앱 구조

- R.java: 앱에서 사용되는 리소스들의 ID를 저장하고 있는 클래스 파일
  - 리소스의 종류별로 내부클래스(innerclass)가 정의되고 각 리소스의 ID를 정수 상수로 관리
  - 리소스사용방법
    - Javacode에서참조방법:R.리소스유형.리소스이름(예:R.string.app\_name)
    - XML파일에서참조방법:@리소스유형/리소스이름(예:@string/app\_name)

```
<resources>
    <string name="app_name">My Application</string>
</resources>

package com.example.myapplication;

public final class R {
    public static final class string {
        public static final int app_name = 0x7f0b0027;
        public static final int search_menu_title = 0x7f0b0028;
        public static final int status_bar_notification_info_overflow = 0x7f0b0029;
    }
}
```



# 안드로이드 앱 구조

## ■ 리소스 참조 예제

The image shows a screenshot of an Android Studio IDE with two files open: MainActivity.java and activity\_main.xml. Red boxes and arrows highlight the resource references between the two files.

**MainActivity.java (Left Panel):**

```
1 package com.example.young.myapplication;
2
3 import ...
4
5
6
7 public class MainActivity extends AppCompatActivity {
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12
13         TextView textView = findViewById(R.id.textView);
14         textView.setText(R.string.app_name);
15     }
16 }
```

**activity\_main.xml (Right Panel):**

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.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     tools:context=".MainActivity">
9
10     <TextView
11         android:id="@+id/textView"
12         android:layout_width="wrap_content"
13         android:layout_height="wrap_content"
14         android:text="@string/app_name"
15         app:layout_constraintBottom_toBottomOf="parent"
16         app:layout_constraintLeft_toLeftOf="parent"
17         app:layout_constraintRight_toRightOf="parent"
18         app:layout_constraintTop_toTopOf="parent" />
19
20 </android.support.constraint.ConstraintLayout>
```

**Annotations and Arrows:**

- A red box highlights the `<string name="app_name">첫번째 앱</string>` line in activity\_main.xml. An arrow points from this box to the `R.string.app_name` reference in MainActivity.java.
- A red box highlights the `setContentView(R.layout.activity_main);` line in MainActivity.java. An arrow points from this box to the `<android.support.constraint.ConstraintLayout` tag in activity\_main.xml.
- A red box highlights the `findViewById(R.id.textView)` line in MainActivity.java. An arrow points from this box to the `android:id="@+id/textView"` attribute in activity\_main.xml.
- A red box highlights the `textView.setText(R.string.app_name);` line in MainActivity.java. An arrow points from this box to the `android:text="@string/app_name"` attribute in activity\_main.xml.



# 안드로이드 앱 실행

# 앱 빌드 과정

## ■ 작업공간설정

- 안드로이드스튜디오설치
- 프로젝트생성

## ■ 앱작성

- 리소스설정
- 레이아웃편집/매니페스트파일편집
- 액티비티편집

## ■ 빌드와실행

- AVD(AndroidVirtualDevice)
- Android지원기기

## ■ 디버깅및테스트

## ■ 배포



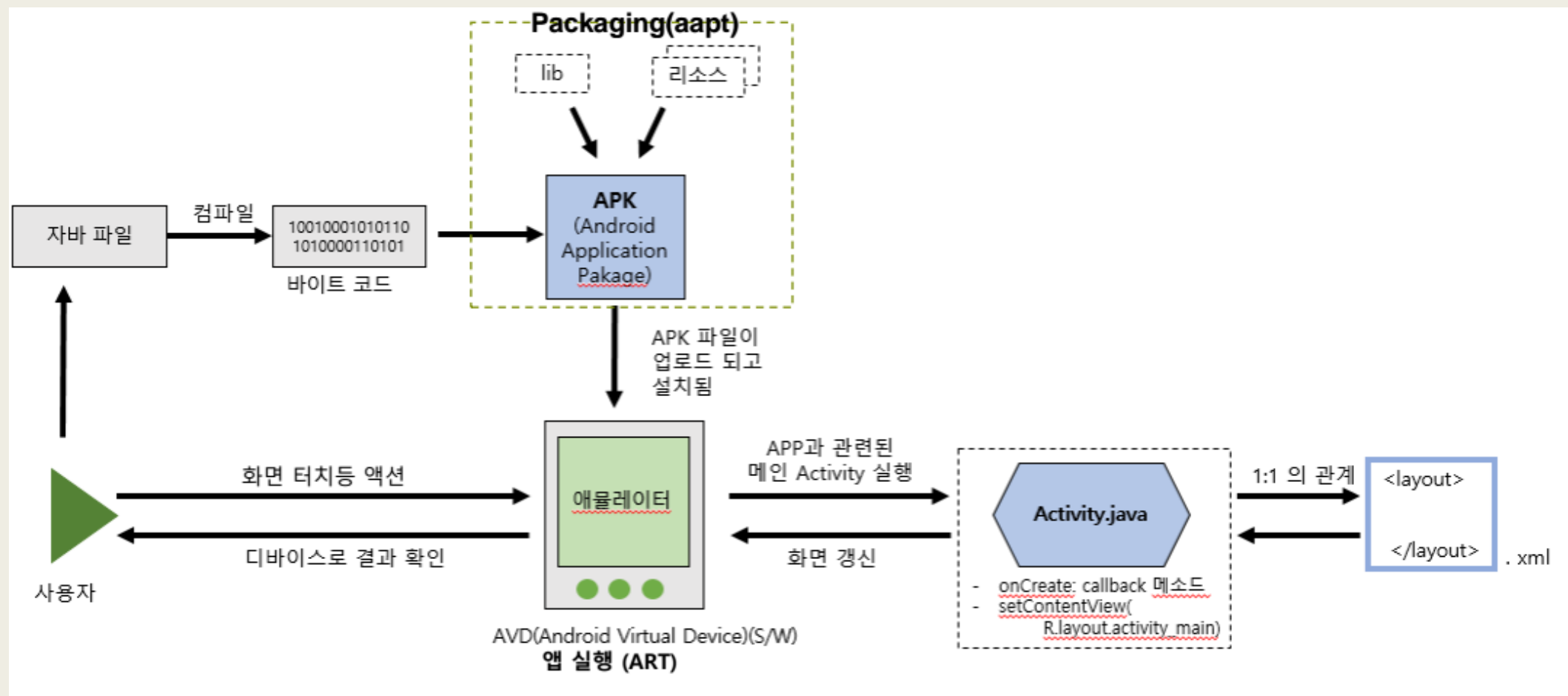
# 안드로이드 앱의 실행 환경



- Email 클라이언트, SMS 프로그램, 달력, 지도, 브라우저 등의 코어 어플리케이션
- 안드로이드 어플리케이션 개발시 필요한 API를 제공
- 안드로이드 프레임 워크에서 필요한 C와 C++ 라이브러리
- 메모리 관리, 프로세스관리, 네트워크 스택 등 하드웨어 지원

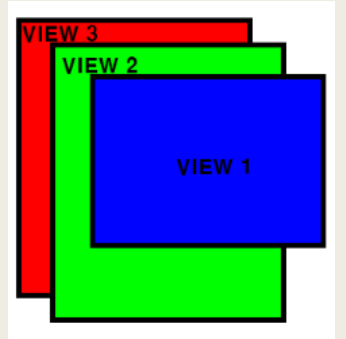
# 앱의 실행 원리

## ■ 앱의 빌드와 실행



**+ 추가 공부**

# FrameLayout?



▼ FrameLayout의 Frame의 '액자'를 의미합니다. '액자'를 FrameLayout과 연관 지어 생각해보다면 FrameLayout의 배치 기능을 이해하는 데 도움이 됩니다. 액자에 여러 장의 같은 크기의 사진을 끼워 넣는다고 가정을 하였을 때 먼저 들어간 사진의 경우는 나중에 들어간 사진에 의해서 가려져서 보이지 않습니다. 마찬가지로 FrameLayout과 연결시켜 생각해보면 FrameLayout에 여러 개의 같은 크기를 가지는 뷰(View)를 배치하였을 때 가장 나중에 추가된 뷰(View)만 전면에 표시가 됩니다.

▼ 가장 마지막에 추가된 뷰(View)를 VIEW1이라고 하면 화면상에서 전면에 나오는 뷰(View)가 VIEW1이 됩니다. 그리고 나머지의 경우는 VIEW1에 가려져서 보이지 않게 됩니다.

▼ 하지만 반드시 전면에 나오는 뷰(View)만 표시되는 건 아닙니다. 앞과 같은 내용은 FrameLayout에 중첩되는 뷰(View)들이 모두 동일한 크기일 때만 해당되는 얘기입니다. 만약 위 그림처럼 가장 마지막에 들어간 뷰(View)의 크기가 먼저 들어간 뷰(View)보다 작으면 먼저 들어간 뷰(View)의 일부도 나타날 수 있습니다.

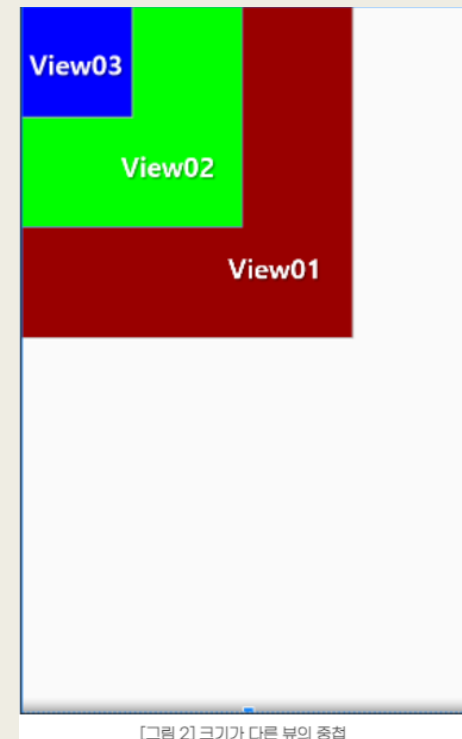
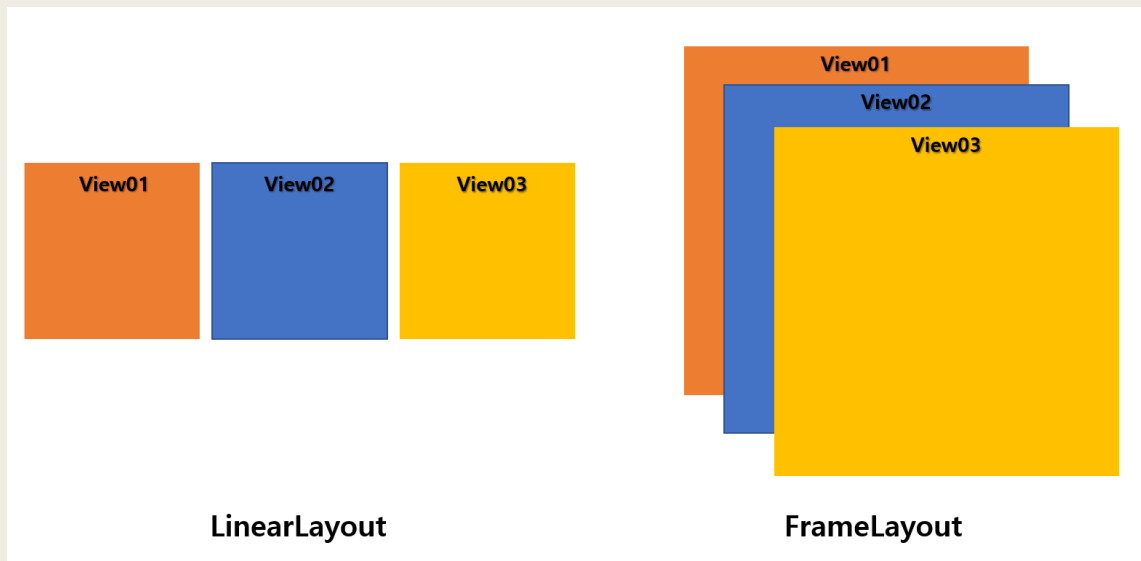
# FrameLayout vs LinearLayout

▼ **프레임 레이아웃 (FrameLayout)**은 여러 개의 뷰(View) 위젯들을 **중첩**하고, 그 중 **하나를 전면**에 표시할 때 사용하는 레이아웃 입니다. 이렇게 여러 뷰를 겹치게 배치할 수 있는 FrameLayout의 특징이, 뷰 위젯을 겹치지 않고 순서대로 화면에 배치하는 선형 레이아웃(LinearLayout) 과 가장 큰 차이점입니다. 여러 뷰를 중첩하여 배치가 가능하기 때문에 좀 더 다양하게 화면을 구성할 수 있습니다. 액자 속 사진을 마음대로 빼고 넣고 하듯이, 경우에 따라 보여주고 싶은 화면을 자유자재로 스위칭 할 수 있도록 하는 것이 FrameLayout 사용 목적입니다.

▼ FrameLayout 을 사용하는 방법은 LinearLayout을 사용하는 것처럼 레이아웃 안에 버튼이나 텍스트 뷰 등의 자식 뷰 위젯들을 선언해주면 됩니다.

▼ 여기서 주의할 점은, FrameLayout 화면에서 가장 전면, 즉 **최 상단**에 보여지는 뷰는 xml 코드에서 **가장 마지막에 해당하는 뷰가 배치**됩니다. LinearLayout 같은 경우는 xml 에서 코드를 작성한 순서대로 (vertical | horizontal) 화면에 배치가 되는데, FrameLayout은 이와 반대로 **xml 코드를 작성한 순서 반대로** 화면에 배치가 되어 xml의 가장 마지막으로 뷰가 최 상단에 나타나는 것이죠. 1→2→3 순서로 작성했다면, 실제로 화면에 나타나는 것은 3→2→1 순 입니다.

FrameLayout은 **아래에서 위로 쌓이는 구조**이기 때문에, 아래의 [그림2] 처럼 크기가 다른 뷰를 추가할 때 먼저 넣은 뷰의 일부가 추가된 뷰의 크기만큼 가려집니다. 이런 식으로 FrameLayout 내에서 첫번째로 선언한 View01 은 가장 하단에 배치되고, 마지막으로 선언한 View03은 가장 상단에 배치됩니다.



[그림 2] 크기가 다른 뷰의 중첩

# EditText?

- 안드로이드 에디트 텍스트(EditText)는 TextView로부터 파생된 클래스로 TextView는 단순히 Text를 보여주는 역할을 한다면 EditText는 Text를 입력 및 수정까지 가능한 뷰(View) 위젯입니다.

## [ EditText 속성 - hint / textColorHint ]

- **hint** 속성은 EditText 배경에 속성값으로 지정한 문자열을 나타나게 합니다. Text 영역에 어떠한 내용을 입력하라는 간단한 안내 문구나 텍스트 입력 예시를 표시할 때 사용합니다. **textColorHint** 속성은 hint 속성에 의해 지정된 문자열의 색상을 지정할 때 사용합니다.
- App을 최초 실행하면 hint 속성의 속성값으로 지정했던 문자열이 배경화면에 나타납니다. 텍스트의 색상은 textColorHint에서 지정되었던 색상을 가진 채로 말이죠. 해당 텍스트는 EditText를 클릭하고 텍스트를 입력하는 순간 사라집니다.

## [ EditText 속성 - inputType ]

- "inputType" 속성은 EditText의 텍스트가 의미하는 데이터의 유형에 따라, 텍스트 입력기(IME)를 통해 입력할 수 있는 문자를 제한하거나, 입력 방법을 지정 또는 텍스트 입력기(IME)와 관련된 부가 기능을 설정할 때 사용하는 속성입니다. 여기서 말하는 "데이터의 유형"이라 함은 텍스트 문자열, 숫자, 전화번호 등과 같이, EditText에 표시된 데이터의 형식을 뜻합니다. 그리고 "입력 방법"은 데이터 유형에 따라 텍스트 입력기(IME)의 키보드 자판이 어떤 모양(QWERTY, 숫자 전용)으로 표시될 것인지를 의미하는 것입니다.



# AppcompatButton?

- ◆ AppCompatActivity 라이브러리는 모든 Android 버전에서 앱에 최신 테마를 적용하는데 도움이 됩니다.
- ◆ 레이아웃 파일을 확장하게 되면 AppCompatActivity 라이브러리가 일부 구성요소를 AppCompatActivity 버전(AppCompatActivity, AppCompatActivityTextView 등)으로 전환합니다.
- ◆ AppCompatActivityButton 은 Button의 하위 클래스 이지만 일부 테마 지원이 내장되어 있습니다.
- ◆ 즉, 일반 Button을 생성하는 경우 Background 및 테마 등이 적용되지 않는 경우가 많은데 이런 경우 AppCompatActivityButton으로 생성하면 해결할 수도 있다고 합니다.

# AppcompatButton vs Button

※ 안드로이드 개발 시 Xml 에서 Button 형식을 2가지를 접할 수 있습니다.  
[ androidx.appcompat.widget.AppCompatButton vs Button ]  
그렇다면 2개의 차이는 무엇일까요?

◆ AppCompatView와 일반 View의 차이점은 동적 background Tint를 허용합니다.

<차이점>

1. ViewCompat의 메서드인 backgroundTint를 통해서 동적으로 button의 background 색상을 변경 가능합니다.
2. R.att.backgroundTint와 R.attr.backgroundTintMode를 사용하여 background 색상을 변경 가능합니다.
3. R.attr.fontFamily를 사용하여 fontFamily를 설정 가능합니다.
4. AppCompatButton에서는 background에 drawable 리소스 적용 가능합니다.

```
<androidx.appcompat.widget.AppCompatButton
    android:text="AppCompatButton"
    android:id="@+id/appcompatbutton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

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


# 작업 물(로그인 & 회원가입 화면 구성)

# 로그인 & 회원가입 화면

뒤로 가기(취소)

11:50 P01\_LoginActivity

# Focus




아이디  
비밀번호

로그인

**회원가입**

12:22 P01\_LoginActivity

# Focus



**back**

닉네임  
아이디  
비밀번호  
비밀번호 확인

비밀번호 다시 입력

회원가입

# 비밀번호 (일치 / 불일치)

12:22 P01\_LoginActivity

back Focus On

닉네임  
닉네임

아이디  
아이디

비밀번호  
....

비밀번호 확인  
.... 일치

회원가입

12:22 P01\_LoginActivity

back Focus On

닉네임  
닉네임

아이디  
아이디

비밀번호  
.....

비밀번호 확인  
..... 확인

비밀번호가 다릅니다.

토스트 창

# 로그인 화면

## (activity\_login.xml)

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

<FrameLayout 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"

    tools:context=".LoginActivity">

    <ImageView

        android:id="@+id/imageView"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:layout_marginLeft="30dp"

        android:layout_marginTop="50dp"

        android:layout_marginRight="30dp"

        app:srcCompat="@drawable/logo" />
```

# 로그인 화면 (activity\_login.xml)

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="249dp"

android:layout\_marginTop="280dp"

android:layout\_marginLeft="40dp"

android:layout\_marginRight="30dp"

android:orientation="vertical">

<EditText

android:id="@+id/editID"

android:layout\_width="wrap\_content"

android:layout\_height="40dp"

android:layout\_gravity="center"

android:layout\_marginBottom="10dp"

android:ems="10"

android:hint="아이디"

android:inputType="textPersonName"

android:text="" />

<EditText

android:id="@+id/ediPassword"

android:layout\_width="wrap\_content"

android:layout\_height="40dp"

android:layout\_gravity="center"

android:ems="10"

android:hint="비밀번호"

android:inputType="textPassword"

/>

# 로그인 화면

## (activity\_login.xml)

```
<androidx.appcompat.widget.AppCompatButton
    android:id="@+id/loginbutton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="20dp"

    android:text="로그인"

    android:textColor="@color/black"

    android:textSize="16sp"

    android:textStyle="bold" />

<androidx.appcompat.widget.AppCompatButton

    android:id="@+id/signin"        android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="10dp"

    android:text="회원가입"

    android:textColor="@color/black"

    android:textSize="13dp"

    android:textStyle="bold" />

</LinearLayout>

</FrameLayout>
```



# 로그인 화면 (LoginActivity.java)

```
package com.example.p01_loginactivity;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.widget.TextView;

public class LoginActivity extends AppCompatActivity {

    TextView sign;

    @Override

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

        setContentView(R.layout.activity_login);

        //회원가입 버튼

        sign = findViewById(R.id.signin);

        //회원가입 버튼 클릭 시, 회원가입 페이지로 이동

        sign.setOnClickListener(v -> {

            Intent intent = new Intent(this, SignupActivity.class);

            startActivity(intent);

        });

    }

}
```

# 회원가입 화면 (activity\_signup.xml)

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

```
<FrameLayout
```

```
    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"
```

```
    tools:context=".SignupActivity">
```

```
    <ImageView
```

```
        android:id="@+id/imageView2"  
        android:layout_width="match_parent"
```

```
        android:layout_height="75dp"
```

```
        android:layout_marginTop="10dp"
```

```
        app:srcCompat="@drawable/logo" />
```

# 회원가입 화면 (activity\_signup.xml)

```
<LinearLayout
```

```
    android:layout_width="match_parent"  
    android:layout_height="match_parent"
```

```
    android:orientation="vertical">
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="30dp"  
    android:layout_marginBottom="10dp"
```

```
    android:orientation="horizontal">
```

```
<TextView
```

```
    android:id="@+id/back"
```

```
    android:layout_width="148dp"  
    android:layout_height="match_parent"
```

```
    android:layout_weight="1.7"
```

```
    android:text="back"
```

```
    android:textColor="@color/black"
```

```
    android:textSize="20dp" />
```

```
</LinearLayout>
```

# 회원가입 화면 (activity\_signup.xml)

```
<LinearLayout
    android:layout_width="match_parent"        android:layout_height="wrap_content"
    android:layout_marginTop="10dp"

    android:layout_marginLeft="30dp"        android:layout_marginRight="30dp"

    android:orientation="vertical" >

    <TextView
        android:id="@+id/signNametext"        android:layout_width="match_parent"
        android:layout_height="wrap_content"        android:layout_marginBottom="5dp"

        android:text="닉 네 임"

        android:textColor="@color/black"

        android:textSize="20dp"

        android:textStyle="bold"        />

    <EditText
        android:id="@+id/signName"        android:layout_width="match_parent"
        android:layout_height="40dp"        android:layout_marginBottom="20dp"

        android:ems="10"

        android:hint=" 닉 네 임"

        android:inputType="textPersonName"

        android:text="" />
```

# 회원가입 화면 (activity\_signup.xml)

<TextView

```
android:id="@+id/signIDtext"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_marginBottom="5dp"
```

```
android:text="아이디"
```

```
android:textColor="@color/black"
```

```
android:textSize="20dp"
```

```
android:textStyle="bold" />
```

<EditText

```
android:id="@+id/signID"
```

```
android:layout_width="match_parent"  
android:layout_height="40dp"  
android:layout_marginBottom="20dp"
```

```
android:ems="10"
```

```
android:hint="아이디"
```

```
android:inputType="textPersonName"
```

```
android:text=""
```

/>

# 회원가입 화면 (activity\_signup.xml)

```
<LinearLayout
```

```
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="horizontal">
```

```
</LinearLayout>
```

```
<TextView
```

```
    android:id="@+id/signPWtext"                android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginBottom="5dp"
```

```
    android:text="비밀번호"
```

```
    android:textColor="@color/black"
```

```
    android:textSize="20dp"
```

```
    android:textStyle="bold" />
```

```
<EditText
```

```
    android:id="@+id/signPW"
```

```
    android:layout_width="match_parent"                android:layout_height="40dp"  
    android:layout_marginBottom="20dp"
```

```
    android:ems="10"
```

```
    android:hint="비밀번호"
```

```
    android:inputType="textPassword"
```

```
    android:text="" />
```

# 회원가입 화면 (activity\_signup.xml)

<TextView

```
android:id="@+id/signPW2text"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_marginBottom="5dp"
```

```
android:text="비밀번호 확인"
```

```
android:textColor="@color/black"
```

```
android:textSize="20dp"
```

```
android:textStyle="bold" />
```

<LinearLayout

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

<EditText

```
android:id="@+id/signPW2"  
android:layout_width="wrap_content"  
android:layout_height="40dp"  
android:layout_marginBottom="20dp"
```

```
android:ems="10"
```

```
android:hint="비밀번호 다시 입력"  
android:inputType="textPassword"
```

```
android:text=""
```

/>

# 회원가입 화면 (activity\_signup.xml)

```
<androidx.appcompat.widget.AppCompatButton
android:id="@+id/pwcheckboxbutton"
android:layout_width="wrap_content"
android:layout_height="40dp"                android:layout_marginLeft="10dp"

android:text="확인"

android:textColor="@color/white"

android:textSize="16sp"

android:textStyle="bold" />

</LinearLayout>

<androidx.appcompat.widget.AppCompatButton
android:id="@+id/signupbutton"
android:layout_width="wrap_content"                android:layout_height="40dp"

android:layout_gravity="center"

android:layout_marginTop="20dp"

android:text="회 원 가 입"

android:textColor="@color/white"

android:textSize="16sp"

android:textStyle="bold" />

</LinearLayout>

</LinearLayout>

</FrameLayout>
```



# 회원가입 화면 (SignupActivity.java)

```
package com.example.p01_loginactivity;

import android.app.Activity;import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;import android.os.Bundle;

import android.view.View;import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;


public class SignupActivity extends AppCompatActivity {

    TextView back;

    EditText name,id,pw,pw2;

    Button pwcheck, submit;

    @Override    protected void onCreate(Bundle savedInstanceState) {

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


        //뒤로 가기 버튼

        back = findViewById(R.id.back);

        back.setOnClickListener(v -> onBackPressed() );
```

# 회원가입 화면 (SignupActivity.java)

```
//가입 항목
```

```
name = findViewById(R.id.signName);
```

```
id=findViewById(R.id.signID);
```

```
pw=findViewById(R.id.signPW);
```

```
pw2=findViewById(R.id.signPW2);
```

```
//비밀번호 확인 버튼
```

```
pwcheck = findViewById(R.id.pwcheckbutton);    pwcheck.setOnClickListener(v -> {
```

```
if(pw.getText().toString().equals(pw2.getText().toString())){
```

```
pwcheck.setText("일 치");
```

```
}else{
```

```
Toast.makeText(SignupActivity.this, "비밀번호가 다릅니다.", Toast.LENGTH_LONG).show();
```

```
}
```

```
});
```

```
//회원가입 완료 버튼
```

```
submit = findViewById(R.id.signupbutton);
```

```
submit.setOnClickListener(v -> {
```

```
Intent intent = new Intent(this, LoginActivity.class);
```

```
startActivity(intent);  });
```

```
} }
```