

# Android study

한창훈  
류영승  
조호열  
권대영  
장정운  
이호정



날짜

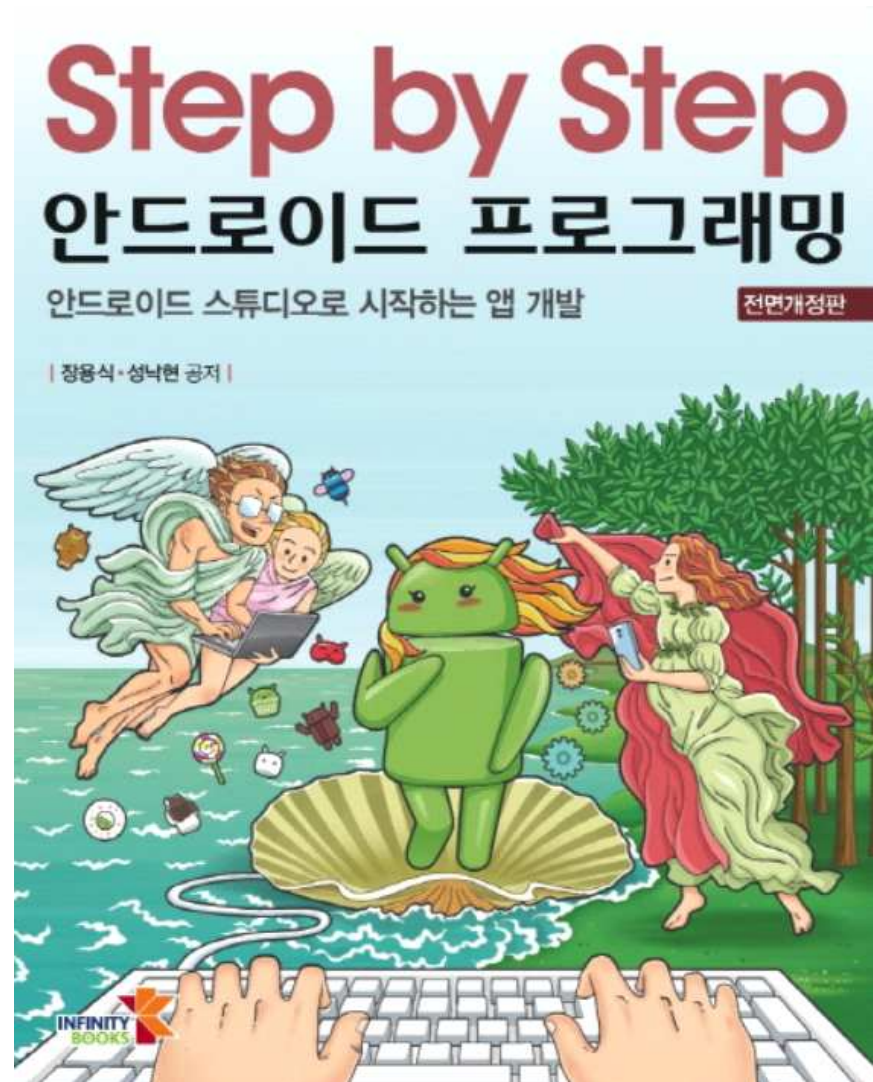
주차

주간  
학습 내용

3/20

1

## 교재 선정



3/20

1

# Android Studio 설치

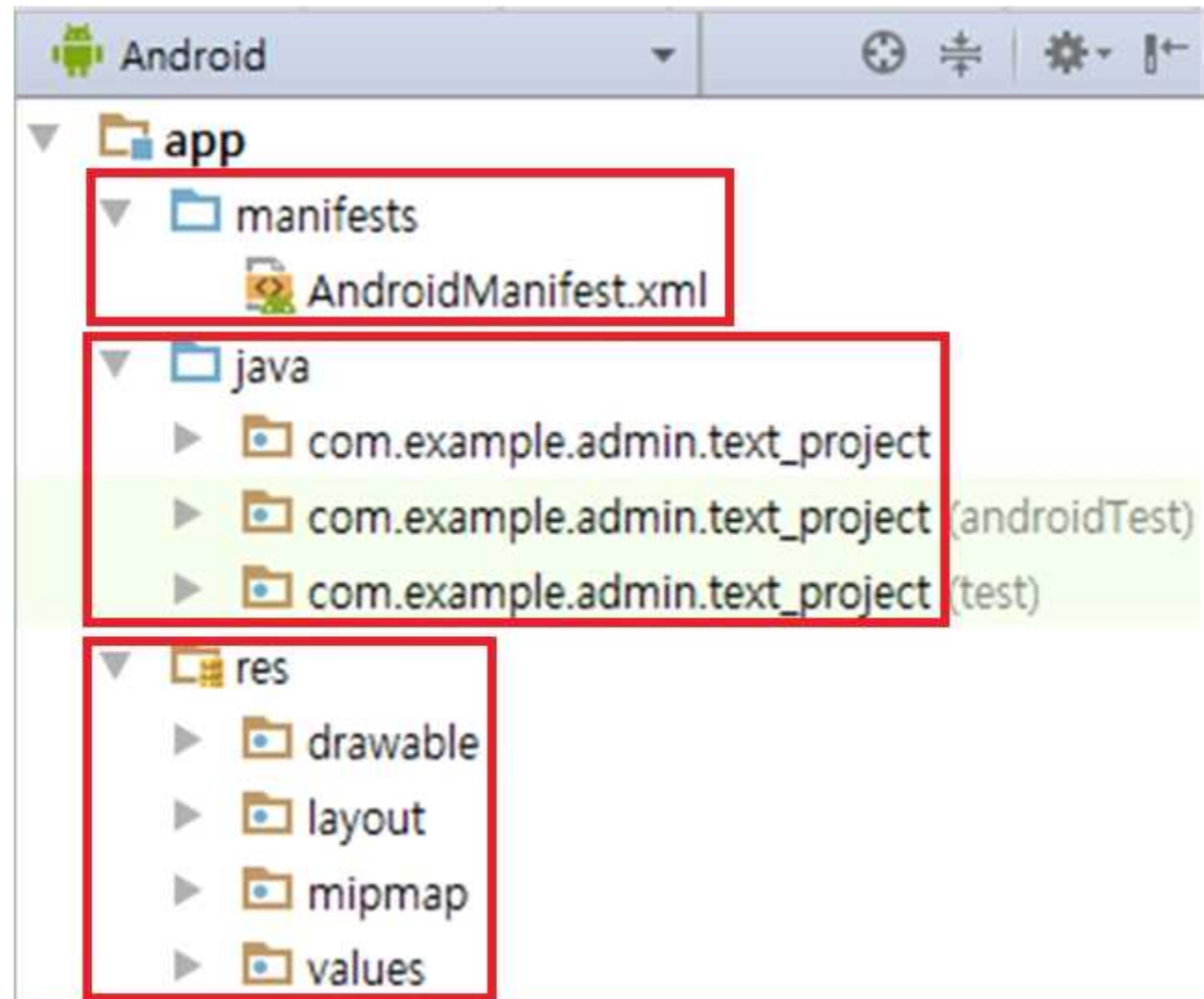


3 / 27

2

# Android Studio

## 프로젝트 파일 구조



3 / 27

2

## (1) manifests 모듈



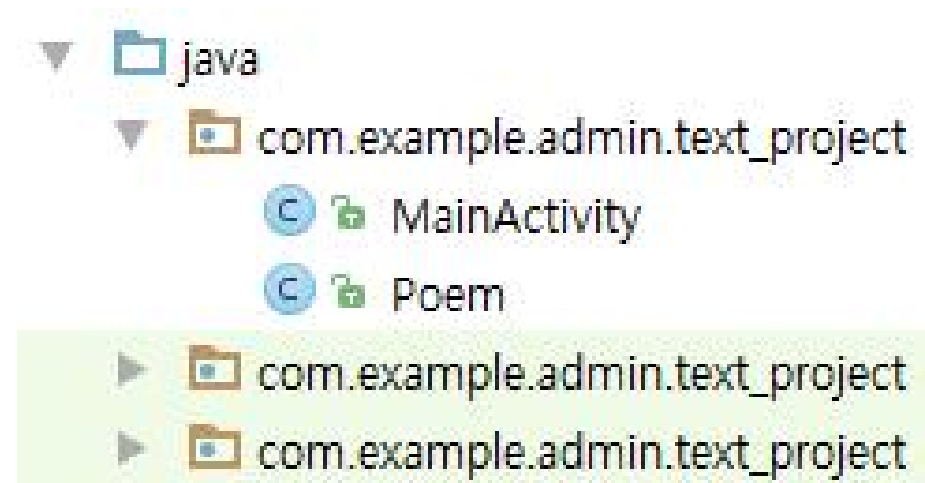
## 모듈에 관한 기본정보 관리

app 라벨, 아이콘,  
엑티비티 라벨,  
처음 실행 될 엑티비티명

3 / 27

2

## (2) java 모듈



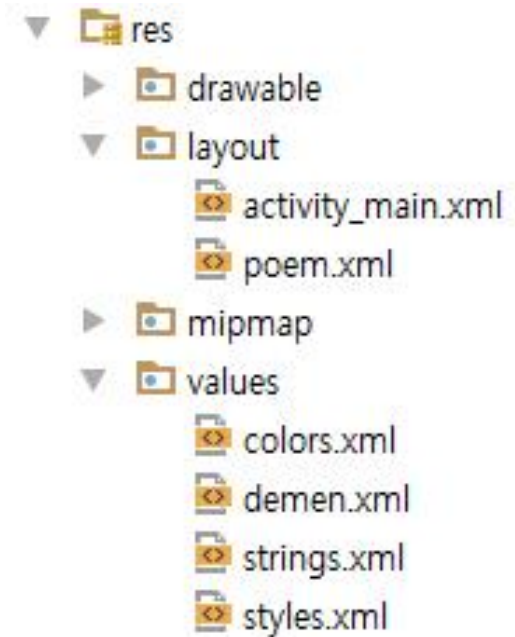
모듈을 위한 자바 소스 파일 관리

화면 출력,  
사용자의 이벤트 처리

3 / 27

2

## (3)res 모듈



모듈에 사용되는 리소스 파일 관리

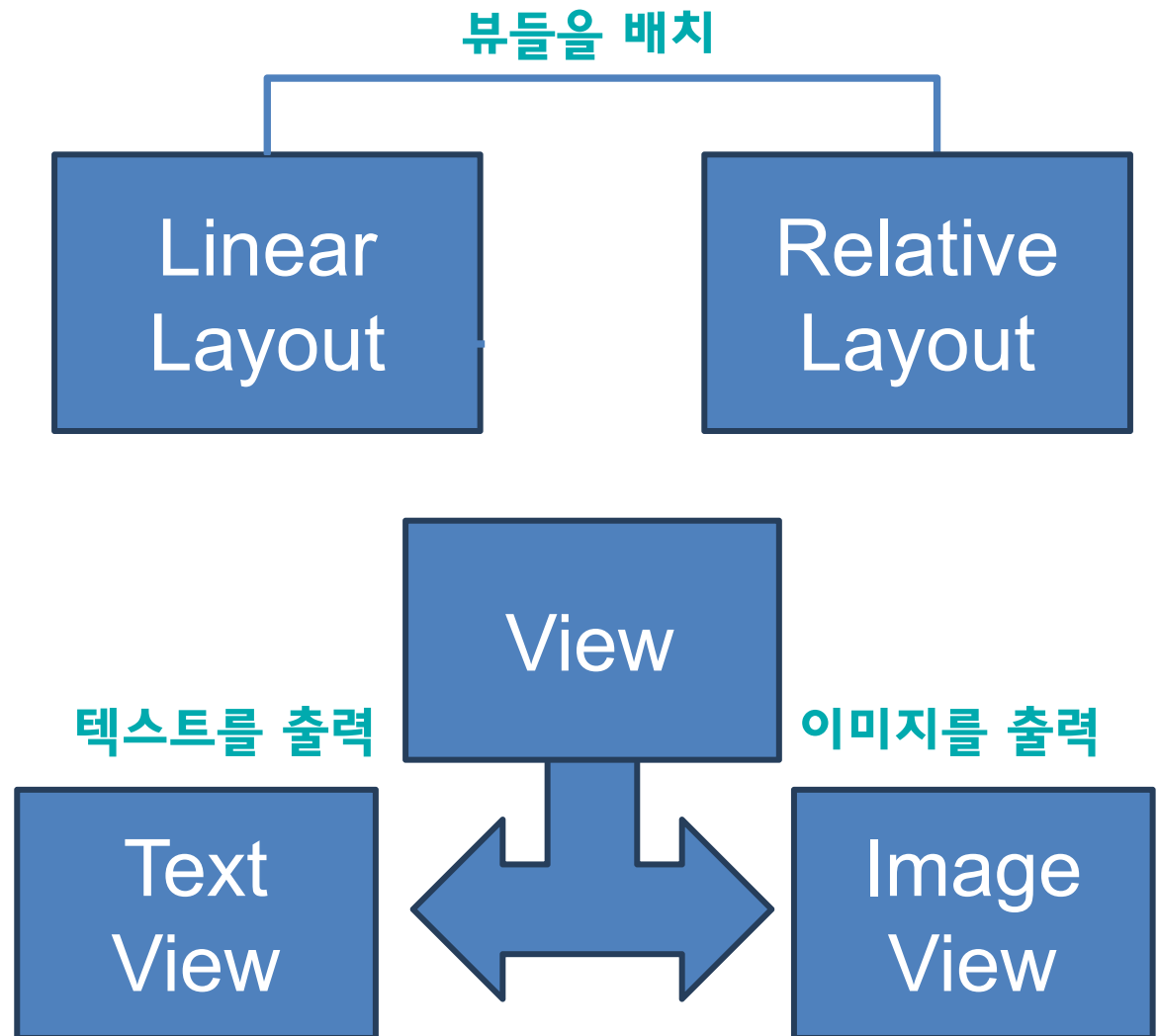
화면 레이아웃 설계,  
텍스트와 이미지 리소스



4/3

3

## XML클래스 구조



4/3

3

# 안드로이드 레이아웃 구성

## (1) RelativeLayout

상대적 위치 지정.  
위치를 지정하지 않으면  
뷰가 0.0에 계속 쌓이는 레이아웃

## (2) LinearLayout

가로(horizontal)또는 세로(vertical)로  
순차적으로 나열되는 레이아웃

4/3

3

## 클래스 속성

`layout_width & layout_height`

(1) `match_parent`

부모 뷰의 크기에 맞춤

(2) `wrap_content`

생성한 뷰의 내용물 크기에 맞춤

(3) `dp`

직접 뷰의 크기값을 설정

4/3

3

## 클래스 속성

### (1) `android:id`

뷰 고유의 식별자(이름)를 설정해준다.

### (2) `android:src`

`Drawable` 폴더에 있는 이미지 파일을 불러온다.

### (3) `android:gravity`

뷰 안의 `content`의 위치 설정  
(bottom, top, right, left, center)

4 / 3

3

## 클래스 속성

### (4) `android:layout_gravity`

현재 뷰를 상위 뷰 안의 위치 설정

### (5) `android:layout_weight`

현재 뷰의 가중치를 설정

(값이 클수록 많은 공간 할당)

4/3

3

# TextView 클래스 속성

(1) `android:text`

현재 뷰에 텍스트 출력

(2) `android:textSize`

텍스트의 사이즈 설정

(3) `android:textStyle`

텍스트의 글꼴 설정

(4) `android:textColor`

텍스트의 색상 설정

4/24

4

## 예제 4~5

- (1) 회전하는 이미지 출력
- (2) 터치 위치를 따라오는 기능
- (3) 터치시 진동 기능

4/24

4

예제 4~5 결과



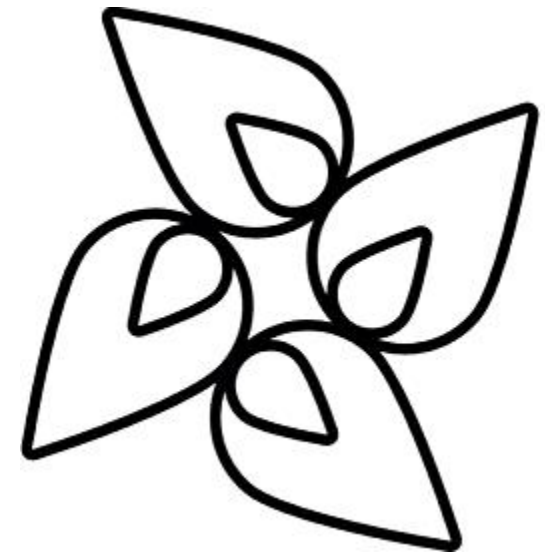
오 리얼?



4/24

4

## (1) 회전하는 이미지 출력



4/24

4

## (1) 회전하는 이미지 출력



```
<ImageView  
    android:id="@+id/pinwheel"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_centerInParent="true"  
    android:src="@drawable/pinwheel" />
```

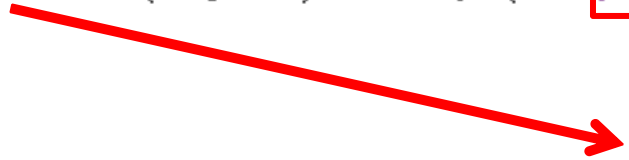
4/24

4

## (1) 회전하는 이미지 출력



```
iv_pinwheel = (ImageView)findViewById(R.id.pinwheel);
```



```
ObjectAnimator object = ObjectAnimator.ofFloat(iv_pinwheel, "rotation", 360);  
  
object.setInterpolator(new LinearInterpolator());  
object.setDuration(2000);  
object.setRepeatCount(ValueAnimator.INFINITE);  
object.start();
```

# 4/24

# 4

## (2) 터치 위치를 따라오는 기능

```
case MotionEvent.ACTION_MOVE:
    int touch_x = (int) event.getX();
    int touch_y = (int) event.getY();

    ObjectAnimator pinwheel_X = ObjectAnimator.ofFloat
        (iv_pinwheel, "translationX", previous_x, touch_x);
    pinwheel_X.start();

    ObjectAnimator pinwheel_Y = ObjectAnimator.ofFloat
        (iv_pinwheel, "translationY", previous_y, touch_y);
    pinwheel_Y.start();

    previous_x = touch_x;
    previous_y = touch_y;
    break;

case MotionEvent.ACTION_UP:
    break;
```

# 4/24

# 4

## (3) 터치시 진동 기능



```
protected void onCreate(Bundle savedInstanceState) {  
    ■Vibe = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE);
```

### 권한 받아오기

```
switch (event.getAction()) {  
    case MotionEvent.ACTION_DOWN:  
        ■Vibe.vibrate(50);
```

### 진동 설정

4/24

4

## 예제 6 나만의 시집 만들기

### 시목록

#### 별 헤는 밤

윤동주

계절이 지나가는 하늘에는 가을로 가득 차 있습니다..

#### 가지 않은 길

로버트 프로스트

단풍 든 숲 속에 두 갈래 길이 있었습니다. 한 몸이 ...

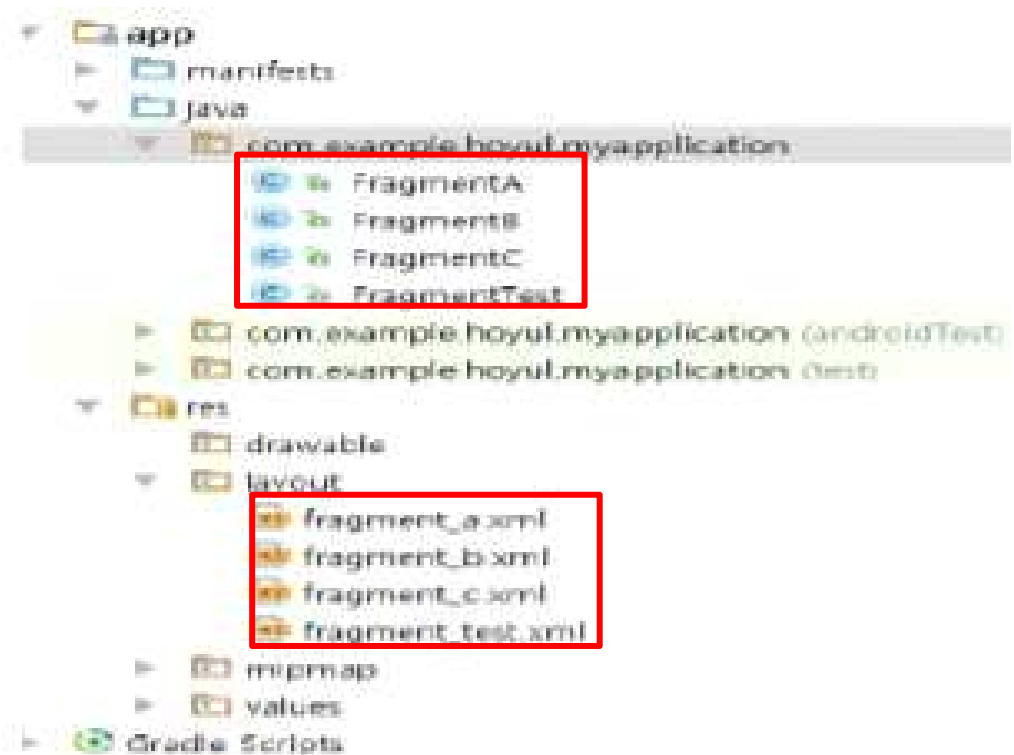


5 / 1

5

## Fragment를 이용하여 activity 화면 분할해보기

구성도



5 / 1

5

## 버튼 만들기

fragment\_test.xml

<Button

android:id="@+id/btn\_a"

android:layout\_width="0dp"

android:layout\_height="80dp"

android:layout\_weight="1"

android:text="화면A"/>

<Button

android:id="@+id/btn\_b"

android:layout\_width="0dp"

android:layout\_height="80dp"

android:layout\_weight="1"

android:text="화면B"/>

<Button

android:id="@+id/btn\_c"

android:layout\_width="0dp"

android:layout\_height="80dp"

android:layout\_weight="1"

android:text="화면C"/>



5 / 1

5

## 버튼 만들기



5 / 1

5

## Fragment를 이용하여 화면 분할

```
<fragment
    android:name="com.example.hoyul.myapplication.FragmentA"
    android:id="@+id/frag_a"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="10dp"/>
```

5 / 1

5

## Fragment를 이용하여 화면 분할



5 / 1

5

# Fragment를 이용하여 화면 분할

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    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:orientation="horizontal"
        android:gravity="center_horizontal">
        <Button
            android:id="@+id/btn_a"
            android:layout_width="0dp"
            android:layout_height="80dp"
            android:layout_weight="1"
            android:text="화면A"/>
        <Button
            android:id="@+id/btn_b"
            android:layout_width="0dp"
            android:layout_height="80dp"
            android:layout_weight="1"
            android:text="화면B"/>
        <Button
            android:id="@+id/btn_c"
            android:layout_width="0dp"
            android:layout_height="80dp"
            android:layout_weight="1"
            android:text="화면C"/>
    </LinearLayout>
    <fragment
        android:name="com.example.hoyul.myapplication.FragmentA"
        android:id="@+id/frag_a"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="10dp"/>
</LinearLayout>
```

fragment\_test.xml

5 / 1

5

## 분할한 activity에 들어갈 화면 구성하기

fragment\_a.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#ffdd00">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="노란색"/>
</RelativeLayout>
```

5 / 1

5

## 분할한 activity에 들어갈 화면 구성하기

fragment\_b.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#f1f2f2" >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="별관색"/>
</RelativeLayout>
```

5 / 1

5

## 분할한 activity에 들어갈 화면 구성하기

fragment\_c.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#22741C" >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="조독색"/>
</RelativeLayout>
```

5 / 1

5

## 각 버튼에 각 화면 매칭 시키기

### FragmentTest.java

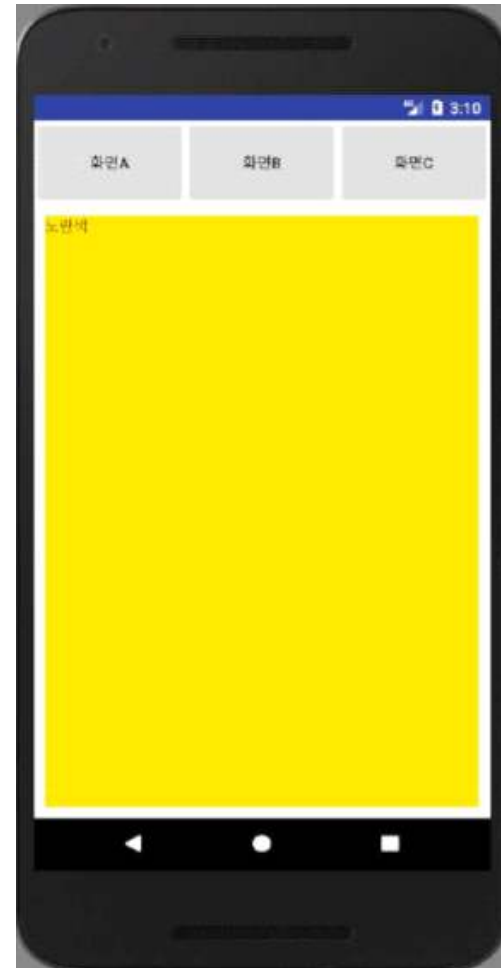
```
btn_a.setOnClickListener(new OnClickListener() {  
    public void onClick(View v) {  
        trans = manager.beginTransaction();  
        trans.replace(R.id.frg_a, fa);  
        trans.commit();  
    }  
});  
btn_b.setOnClickListener(new OnClickListener() {  
    public void onClick(View v) {  
        trans = manager.beginTransaction();  
        trans.replace(R.id.frg_a, fb);  
        trans.commit();  
    }  
});  
btn_c.setOnClickListener(new OnClickListener() {  
    public void onClick(View v) {  
        trans = manager.beginTransaction();  
        trans.replace(R.id.frg_a, fc);  
        trans.commit();  
    }  
});
```



5 / 1

5

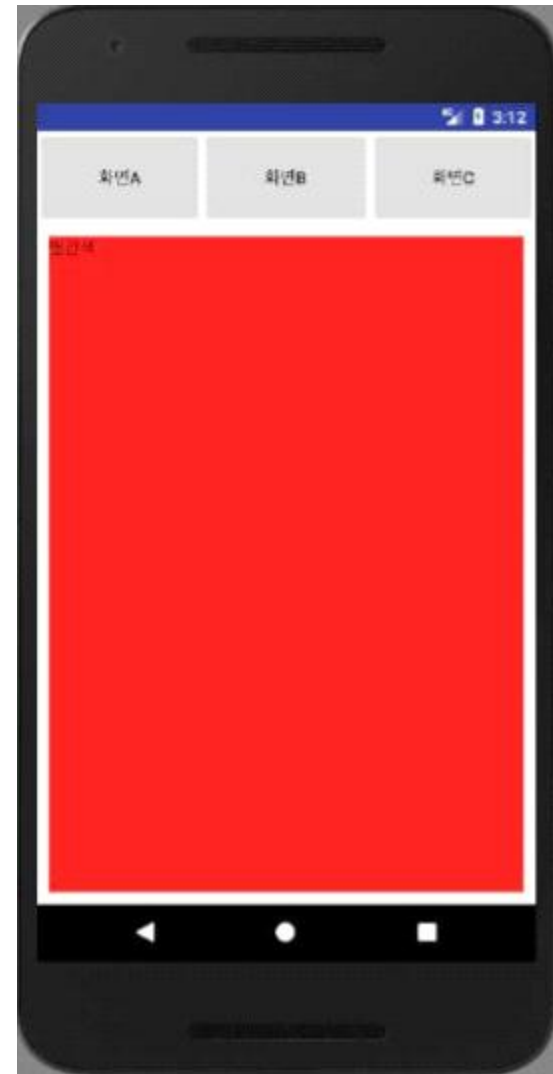
앱 실행시 가장 먼저 나오는 화면  
(기본 설정을 `fragment_a`로 설정)



5 / 1

5

화면B 버튼을 누르면 나오는 화면



5 / 1

5

화면C 버튼을 누르면 나오는 화면



5 / 1

5

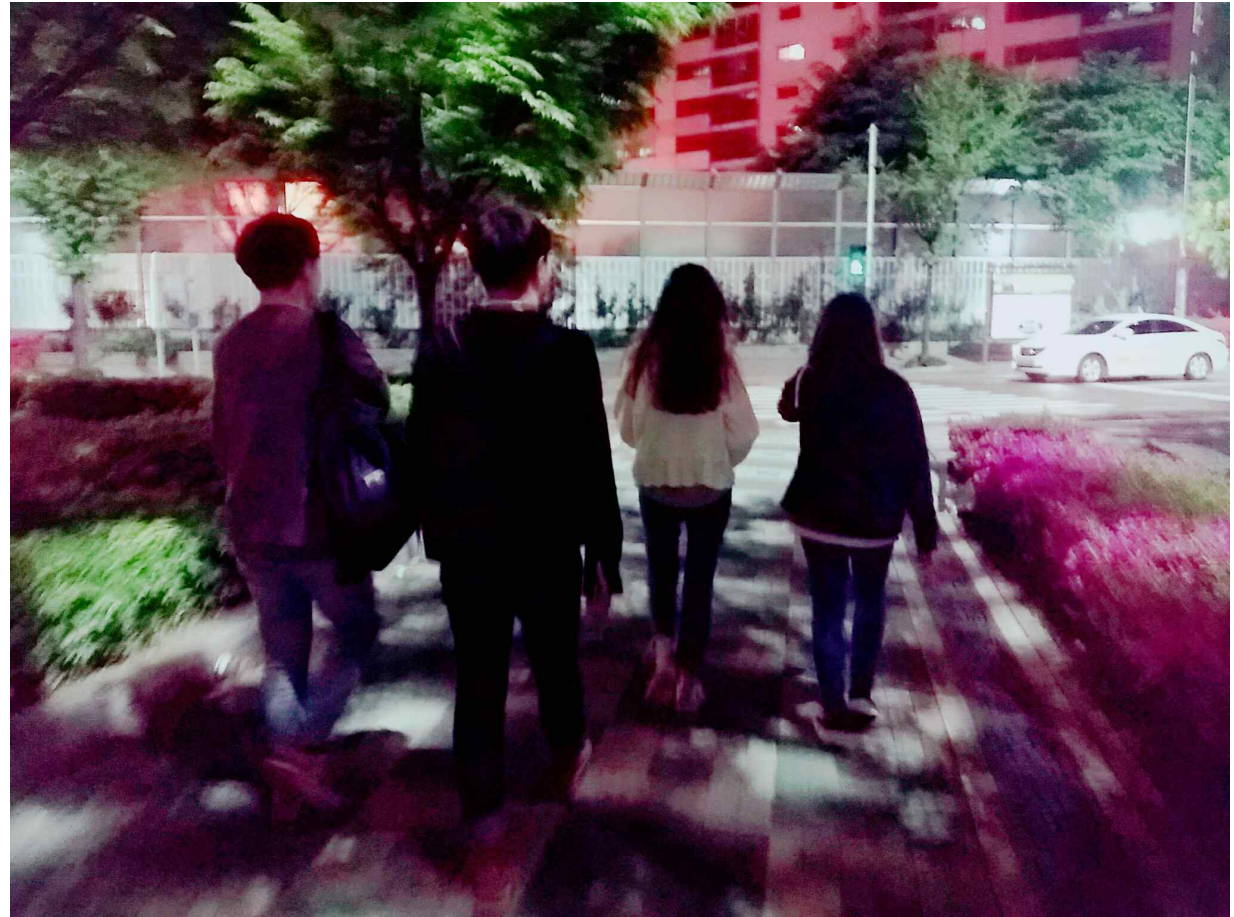
## 단합차 술터디



5 / 1

5

## 단합차 술터디



**앞으로**

6

**교재 진도 마무리**

**서버와 DB 연동을 통해**

**회원가입 & 로그인 기능**

**Application 개발이 목표!**

Q & A