

모바일프로그래밍기초

- 안드로이드(Android) -

○안드로이드 기능/실습 - 8

- 디바이스 실행 방법
- APK 생성 과정

○안드로이드 기능/실습 - 9

- 기본 이벤트 처리
 - ◆ 클릭, 롱클릭, 포커스, 키, 터치

○안드로이드 기능/실습 - 10

- WebView
 - ◆ 웹컨텐츠표현

□ 안드로이드 기능/실습 - 8

- 디바이스 실행, APK 생성 과정

□ 결과의 디바이스 실행

○개발 결과물을 안드로이드폰에서 직접 실행하기 위해서는 각 제조사에서 제공하는 드라이버를 설치하고 그에 따라 진행

- 특정 소프트웨어를 사용하는 것이 아니면 드라이버가 필요 없는 경우도 많음, 적용 범위에 따라 차이가 있음
- 각 스마트폰 제조사의 매뉴얼 참조

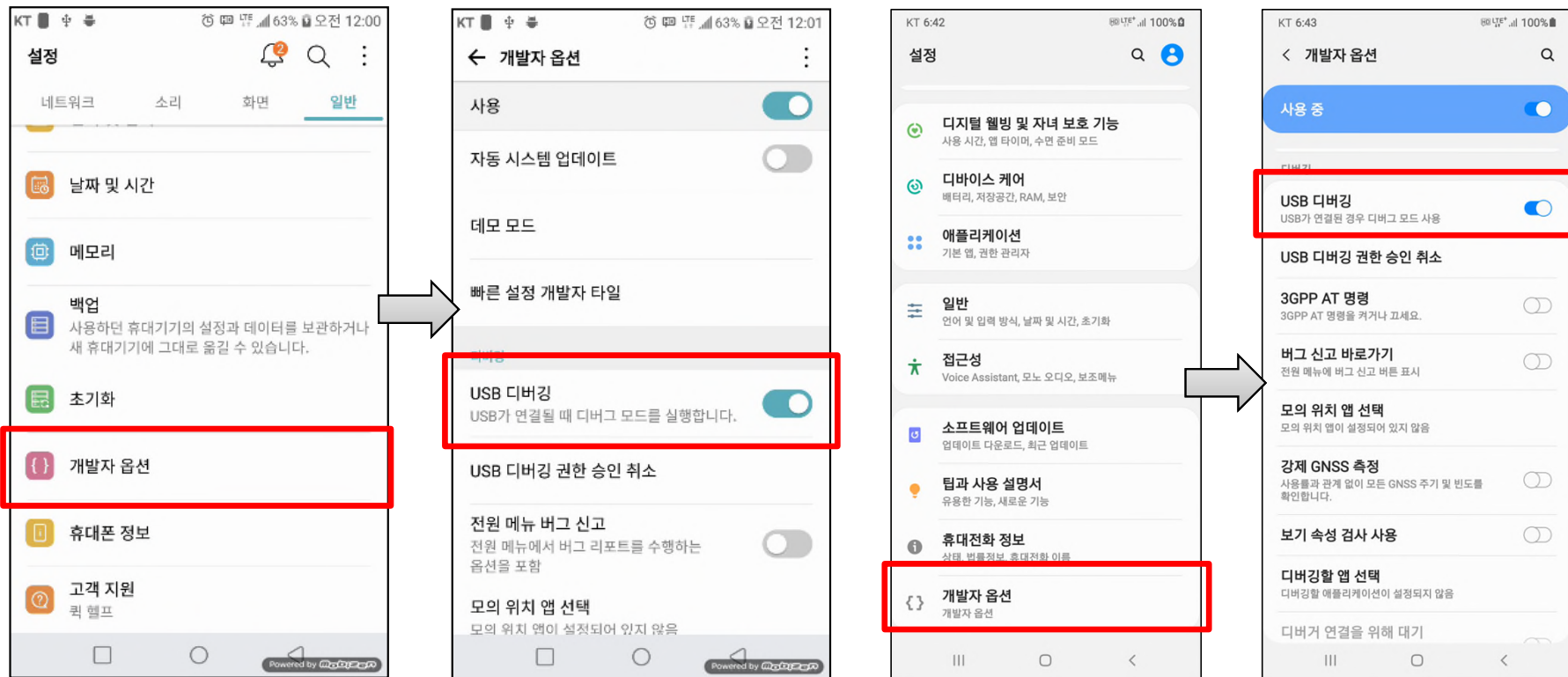
○여기서는 V20(8.0 Oreo), A50(Android10, 11) 등 기준으로 진행

- 다른 제조사, 또는 버전도 메뉴 위치만 다르고 거의 같음

□ 결과의 디바이스 실행

○ 스마트폰으로 직접 디버깅하기 위해서는 USB디버깅이 체크되어 있어야 함

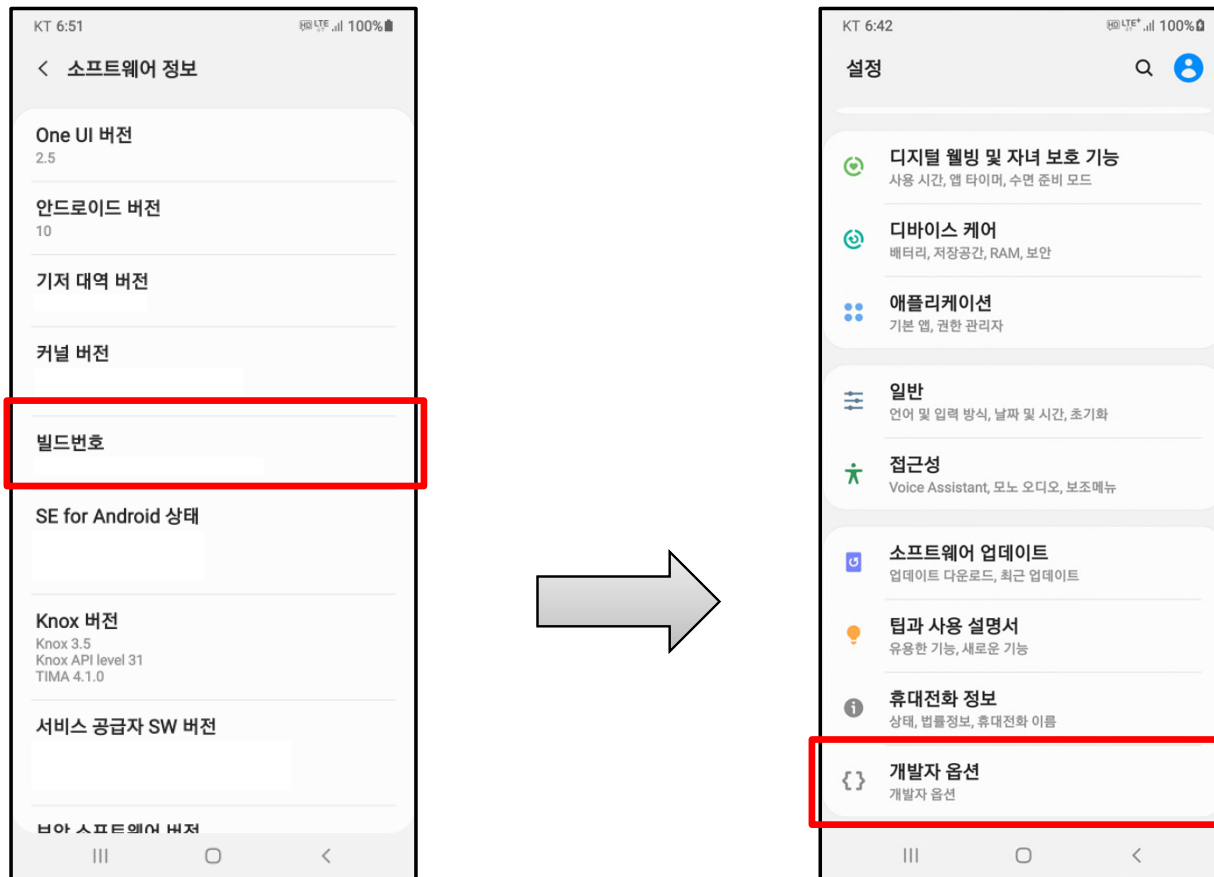
□ 보안 위험성이 높으니 테스트 시에만 체크하고 해제하세요.



□ 결과의 디바이스 실행

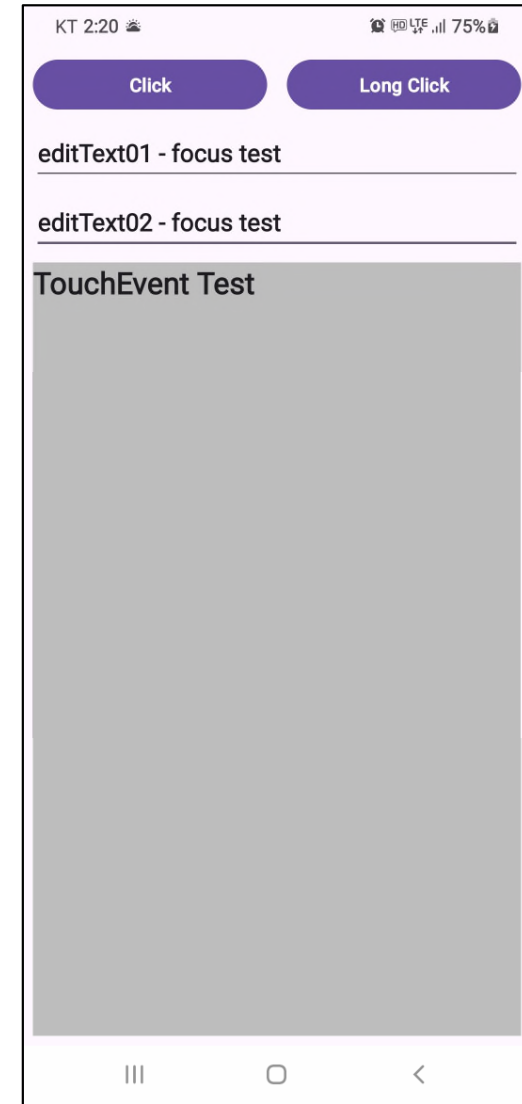
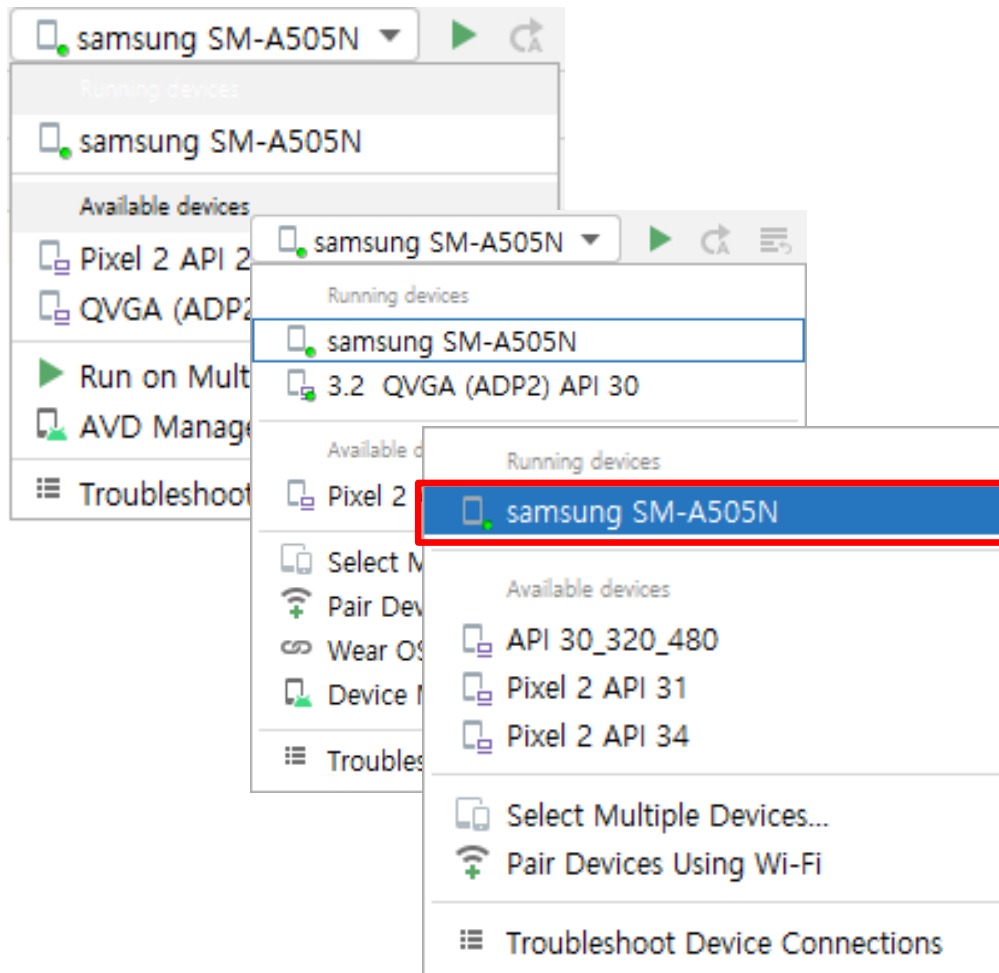
○ 개발자 옵션이 안보이는 경우

- 휴대전화 정보 부분에서 빌드번호 메뉴를 연속으로 클릭
- 빌드번호 메뉴는 소프트웨어 정보 하단, 휴대폰 정보 하단 등, 제조회사에 따라 위치가 다름



□ 결과의 디바이스 실행

○ 실행시 해당 디바이스가 연결되어 정상적으로 인식되는 경우, (A50 등의 연결 모습)



실행 결과

- 안드로이드 응용 프로그램 패키지의 확장자
 - 안드로이드 어플리케이션 설치 파일
 - Android application package
- 개발한 어플리케이션을 내 스마트폰에 설치하기 위한 APK 생성
 - 해당 어플리케이션의 설치 파일 생성 과정
 - 해당 어플리케이션 실행 모습 스크린샷 확인

□ 개발한 애플리케이션 설치

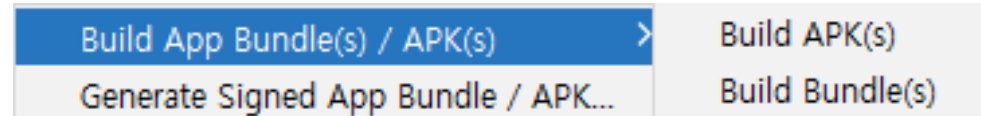
○ 진행 순서

- APK 파일 생성
- 생성한 APK 파일을 스마트폰으로 이동
- 스마트폰에 설치

□ APK 파일 생성 과정

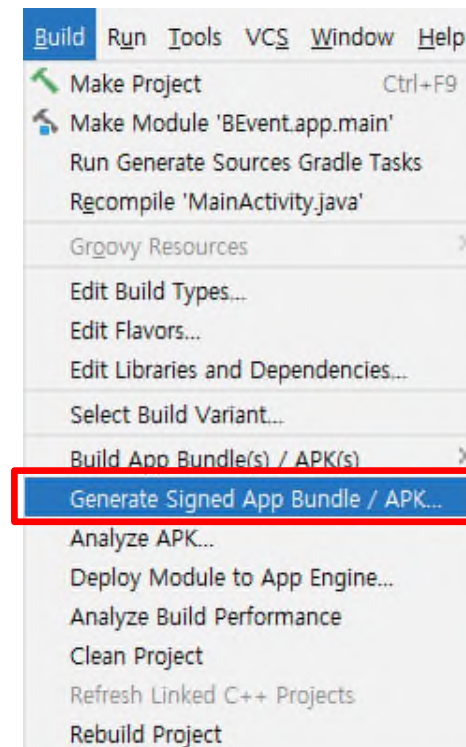
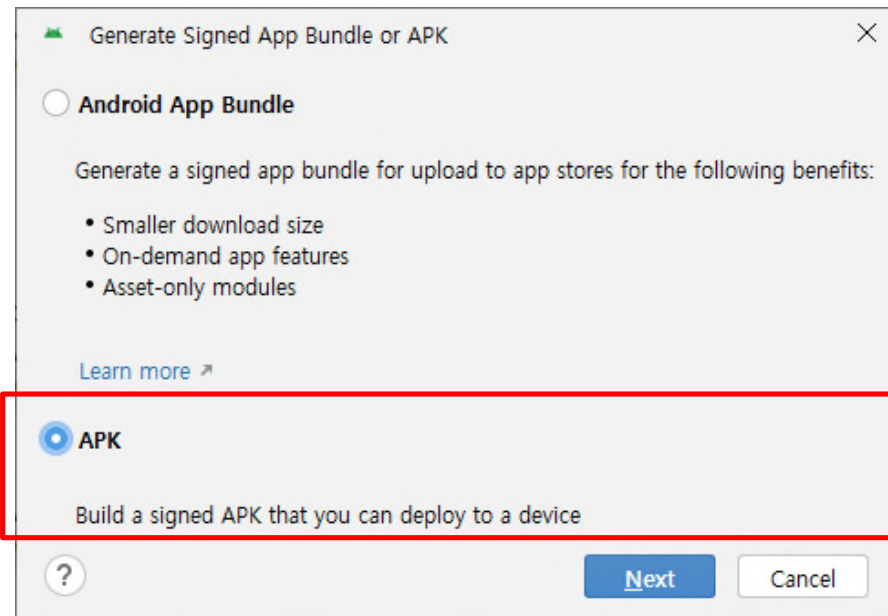
○ APK로 생성하고자 하는 해당 프로젝트에서 [Build] → [Generate Signed Bundle / APK...] 선택

- 참고) 일반적으로 APK 파일을 단순히 생성하는 경우에는 상위에 Build Bundle(s) / APK(s)를 선택해서 생성하면 됨



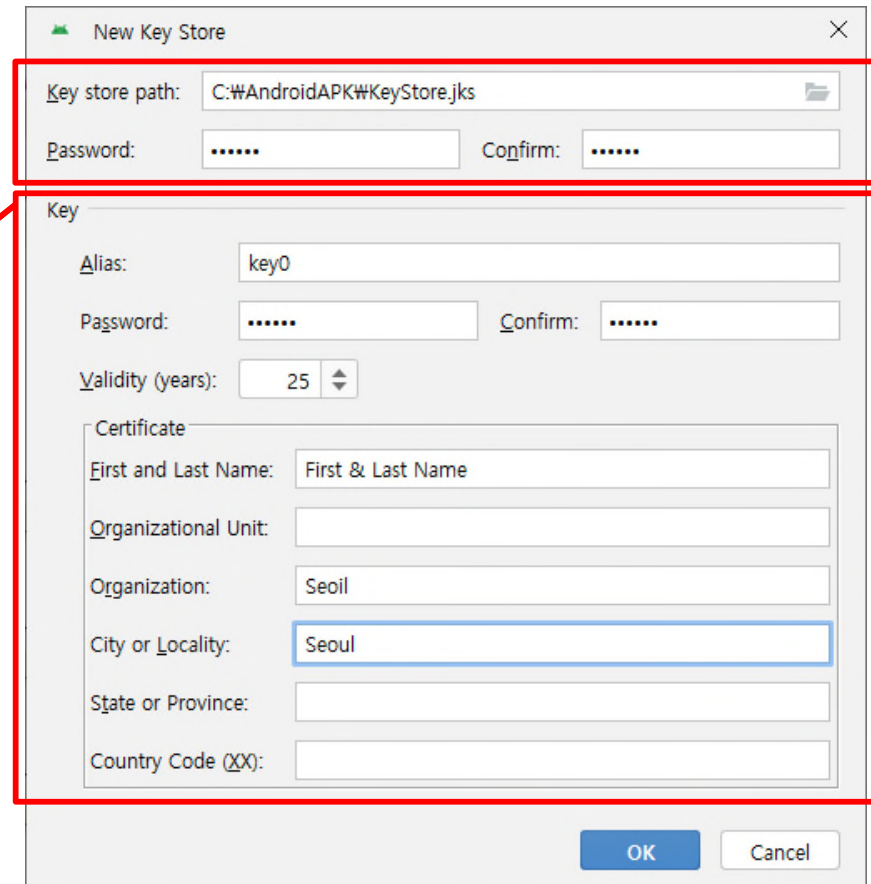
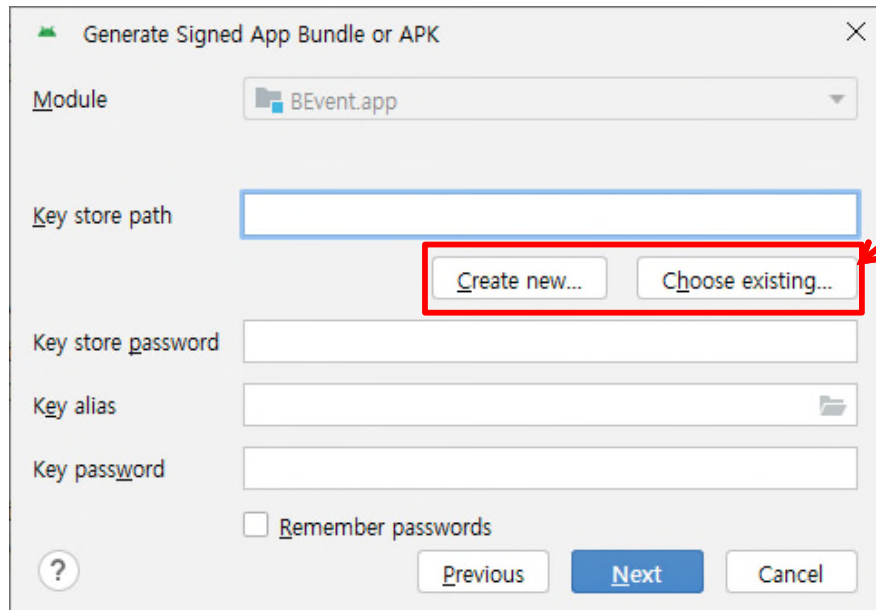
○ [Build] → [Generate Signed Bundle / APK...] 선택

- 스튜디오 버전에 따라 메뉴 이름, 구조가 조금 다름



□ APK 파일 생성 과정

- 키가 있는 경우에는 기존 키로 생성
- 키가 없는 경우, 키를 생성하기 위해서 Create new... 선택
- 저장할 경로와 저장될 이름, 키 생성을 위한 정보 입력 작성



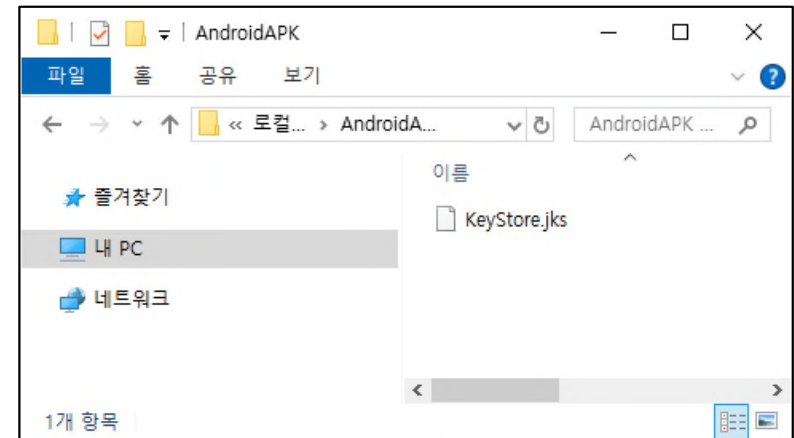
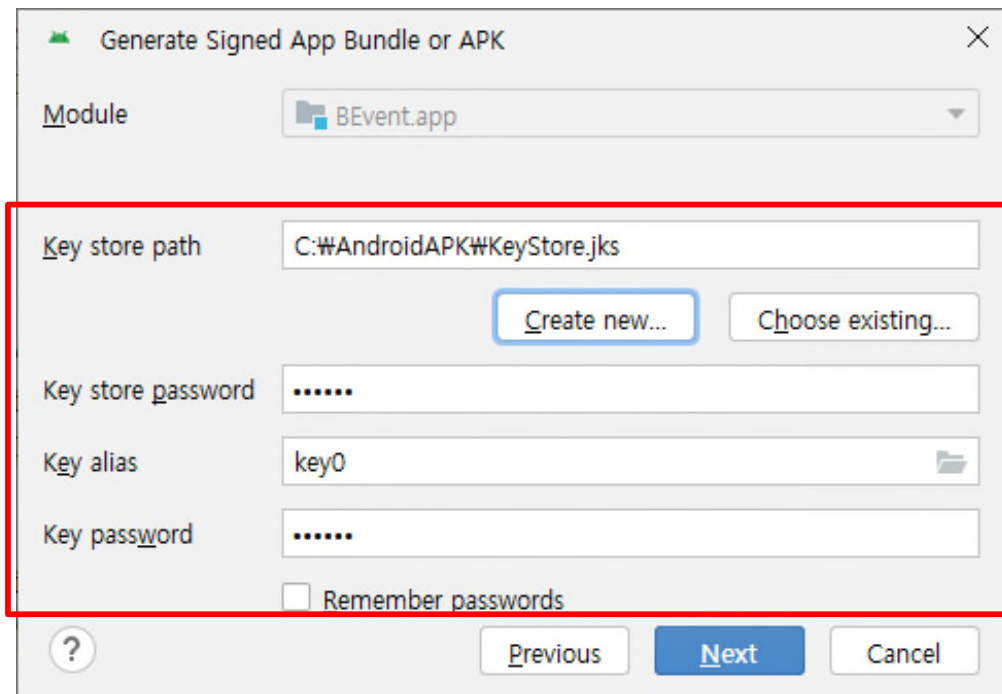
□ APK 파일 생성 과정

○키가 생성되면서 APK 생성을 위한 화면으로 돌아옴

- ❑ 저장된 키를 사용하기 위해서는 생성할 때 입력한 Key store와 Key alias에 대한 패스워드를 입력해야 함

○생성된 KeyStore

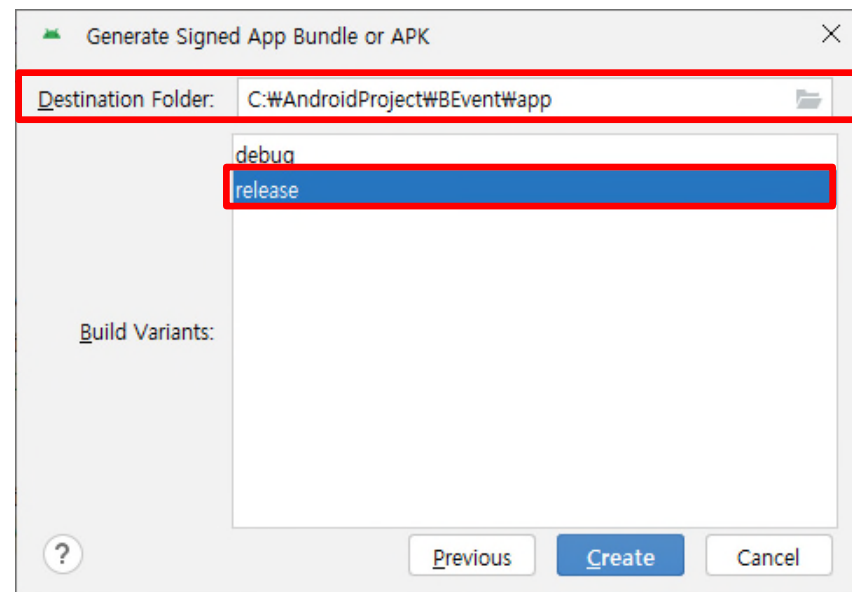
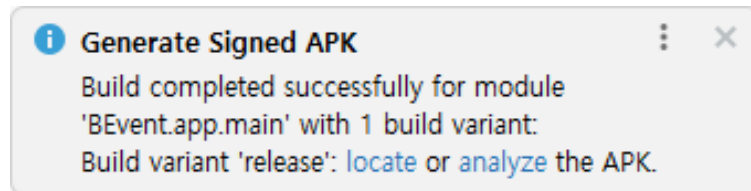
- ❑ 지정된 경로에 생성된 Key 확인



□ APK 파일 생성 과정

○ APK 저장 경로 선택

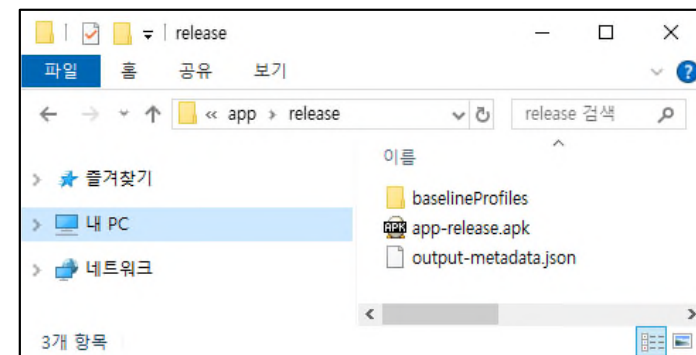
- 생성되는 과정에 약간의 시간 소요
- 생성이 완료되면 Generate signed APK 메시지가 등장



○ 생성된 결과

- 선택한 저장 경로에 생성된 APK

생성한 APK 파일은 필요에 따라서 원하는 이름으로 변경



□ 안드로이드 기능/실습 - 9

- 기본 이벤트 처리

□ 안드로이드 기능 / 실습 - 9

○ 목적

- 사용자 간의 상호작용을 위한 간단한 이벤트 처리 과정 학습 (한 화면에 여러 이벤트가 존재하는 다중 이벤트 처리 학습)

○ 기본 이벤트 종류에 따른 처리 과정 학습

- 클릭
 - ❖ OnClickListener
 - Void onClick(View v)
- 포커스 변경
 - ❖ onFocusChangeListener
 - Void onFocusChange (View v, boolean hasFocus)
- 롱클릭
 - ❖ OnLongClickListener
 - Boolean onLongClick(View v)
- 키 (특정 위치를 사용하는 경우)
 - ❖ OnKeyListener
 - Boolean onKey(View v, int keyCode, KeyEvent event)
 - ❖ (특정 위치를 사용하지 않는 경우)
 - onKeyDown(int keycode, KeyEvent event)
- 터치
 - ❖ onTouchListener
 - Boolean onTouch(View v, MotionEvent event)

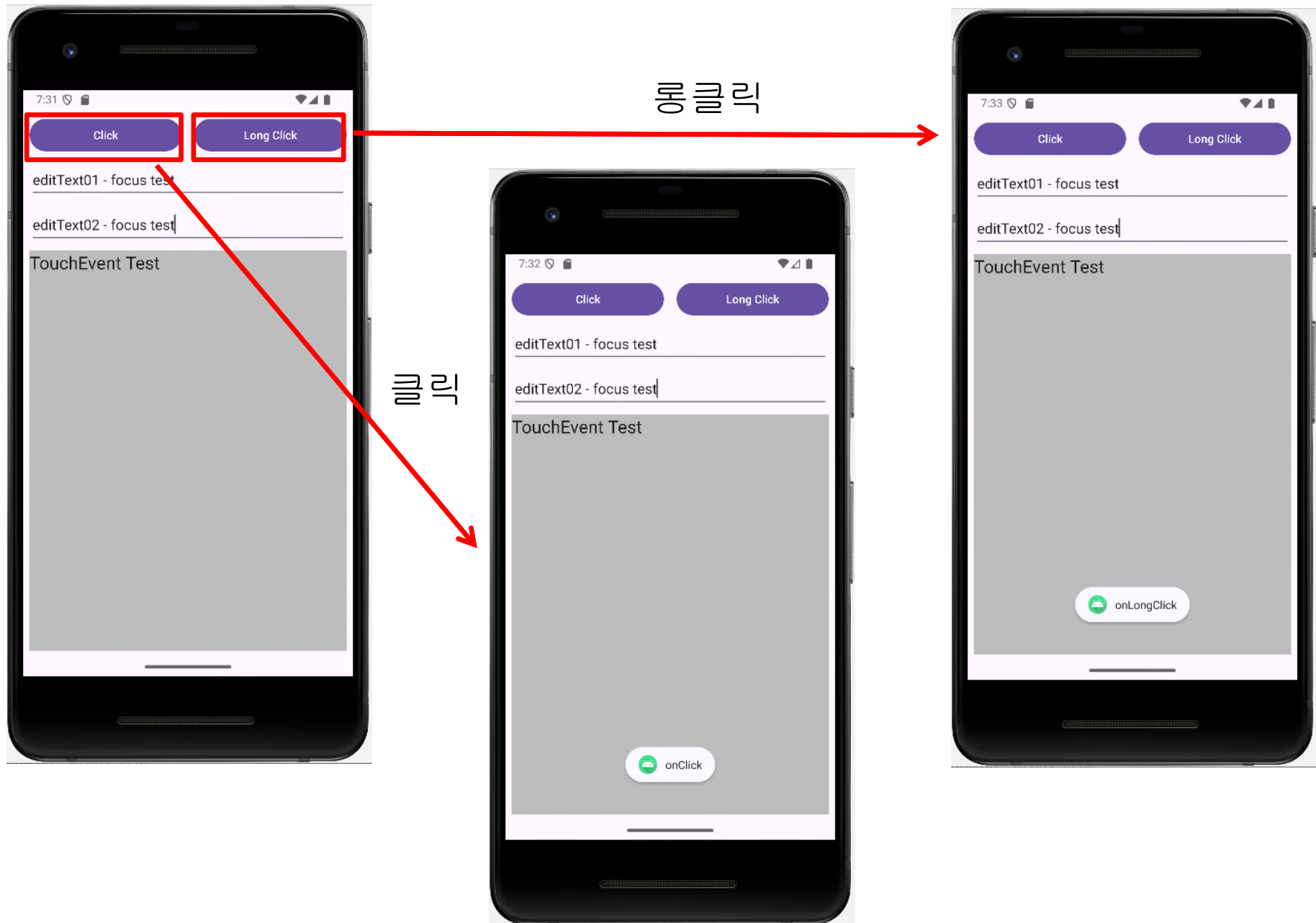
□ 안드로이드 기능 / 실습 - 9

○ KEY 코드

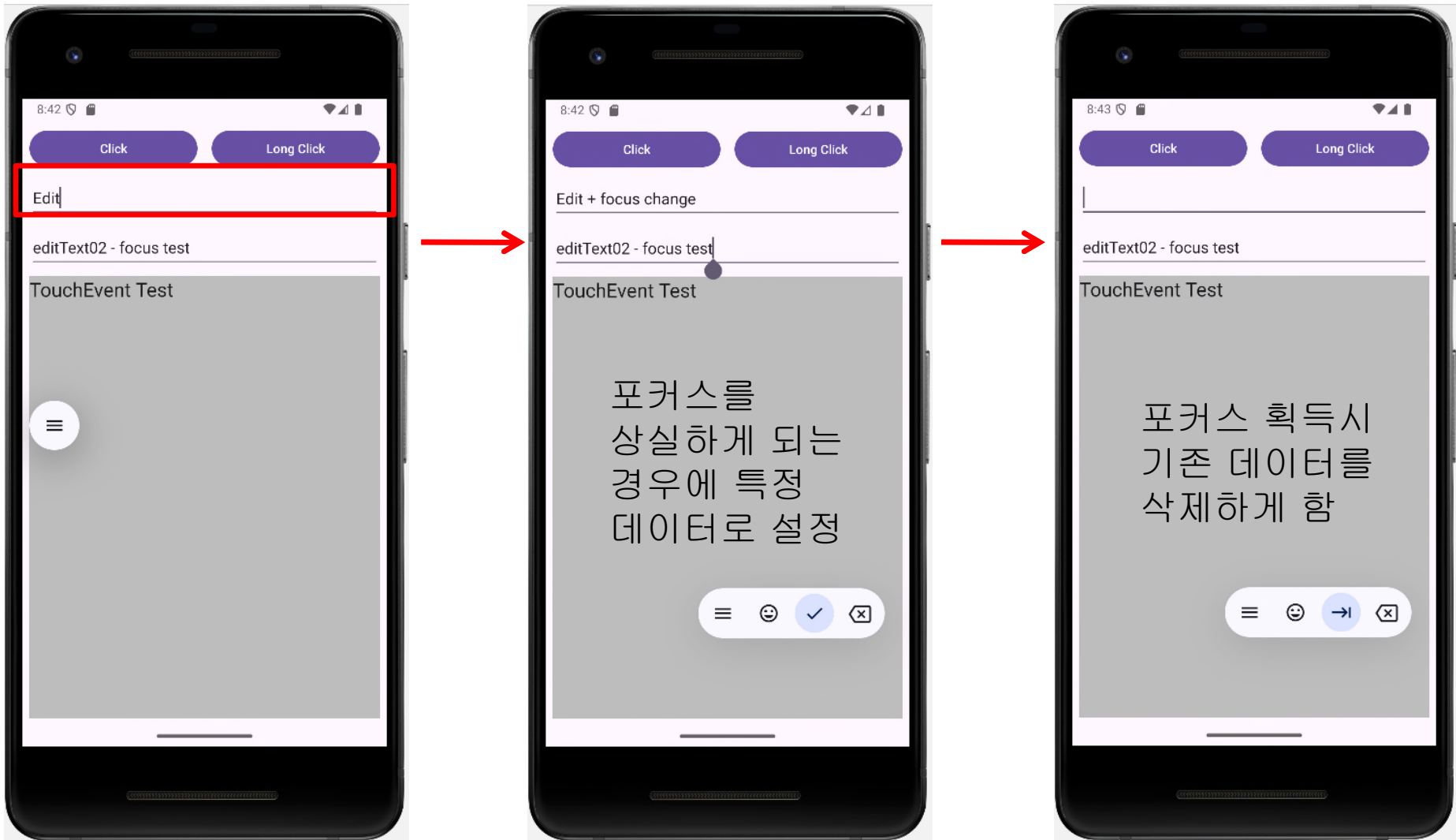
□ KEYCODE_DPAD_LEFT	왼쪽 화살표
□ KEYCODE_DPAD_RIGHT	오른쪽 화살표
□ KEYCODE_DPAD_UP	위쪽 화살표
□ KEYCODE_DPAD_DOWN	아래쪽 화살표
□ KEYCODE_DPAD_CENTER	중앙 버튼
□ KEYCODE_CALL	통화 버튼
□ KEYCODE_ENDCALL	통화 종료 버튼
□ KEYCODE_HOME	홈 버튼
□ KEYCODE_BACK	뒤로 가기 버튼
□ KEYCODE_VOLUME_UP	볼륨 업 버튼
□ KEYCODE_VOLUME_DOWN	볼륨 다운 버튼
□ KEYCODE_0 ~ KEYCODE_9	숫자 0 ~ 9
□ KEYCODE_A ~ KEYCODE_Z	알파벳 A ~ Z

○ 키 코드 가운데 숫자, 알파벳 등의 입력은 안드로이드 버전 상승에 따라 인식이 제한됨

□ 실행 모습 - Click, LongClick Event

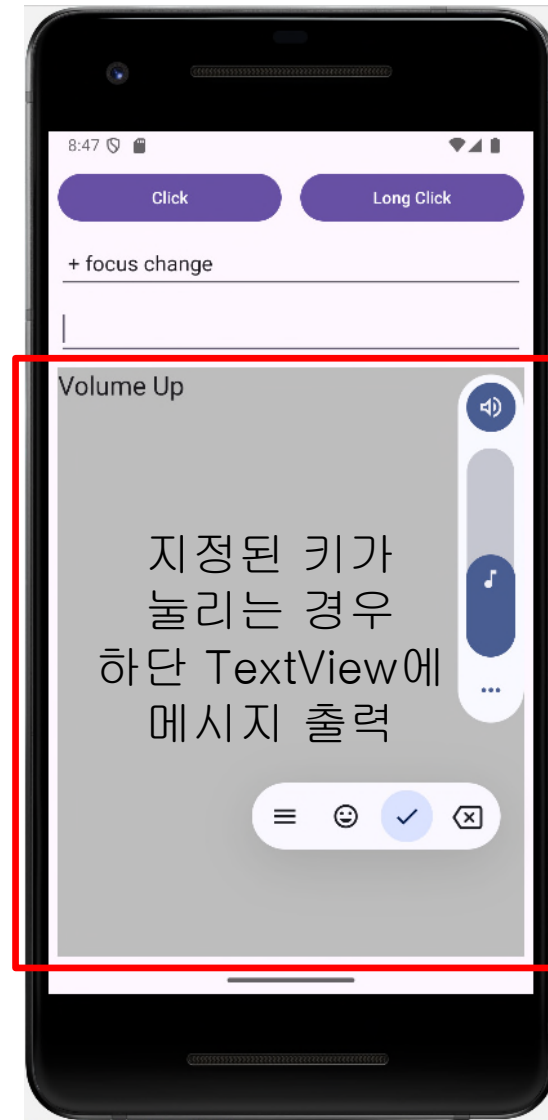
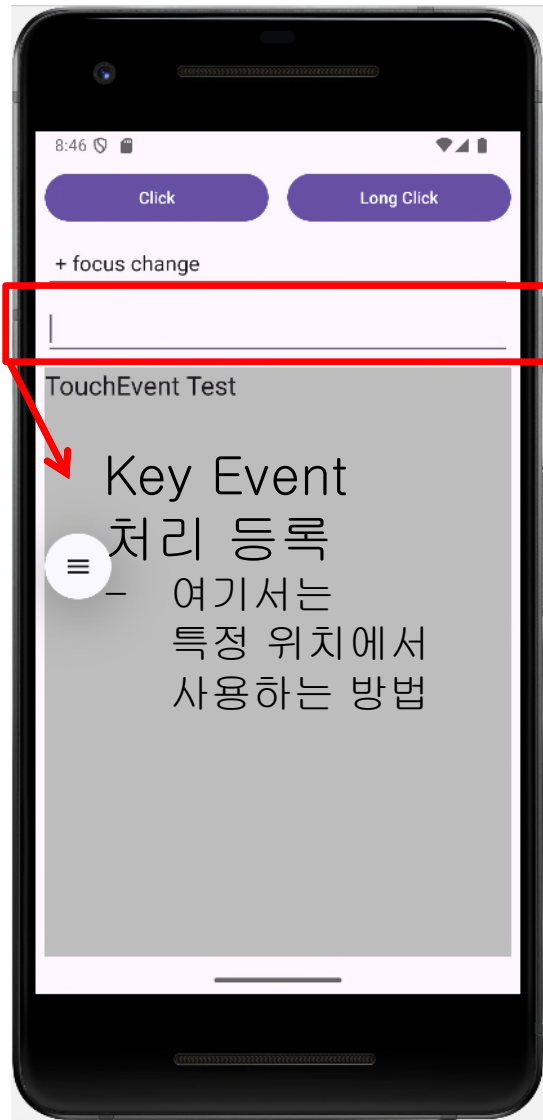


□ Focus Change

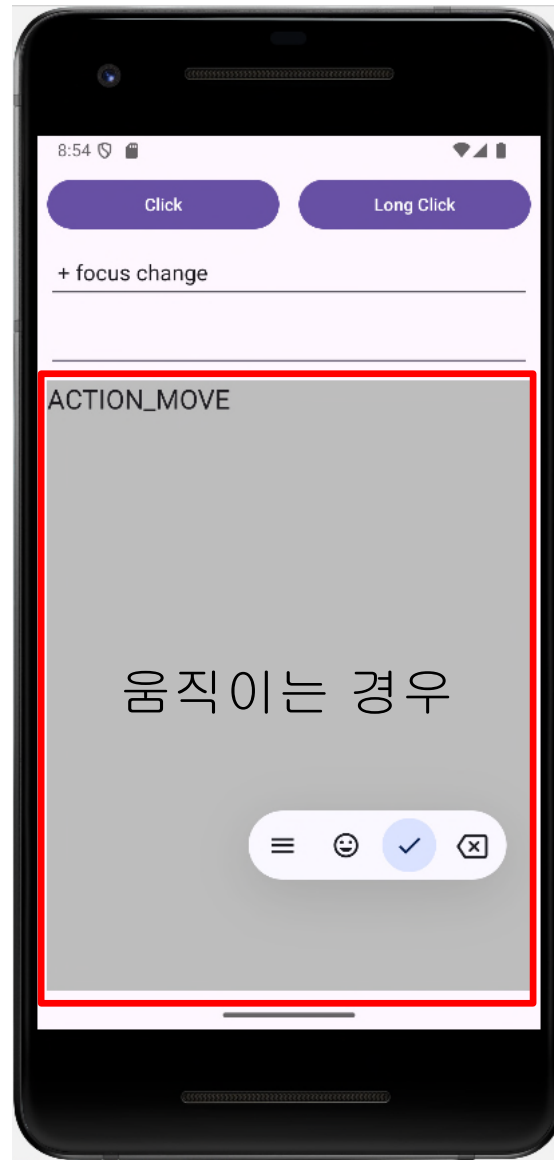


포커스 이동 시
(필요에 따라 데이터/기능에 대한 추가/삭제 등...)

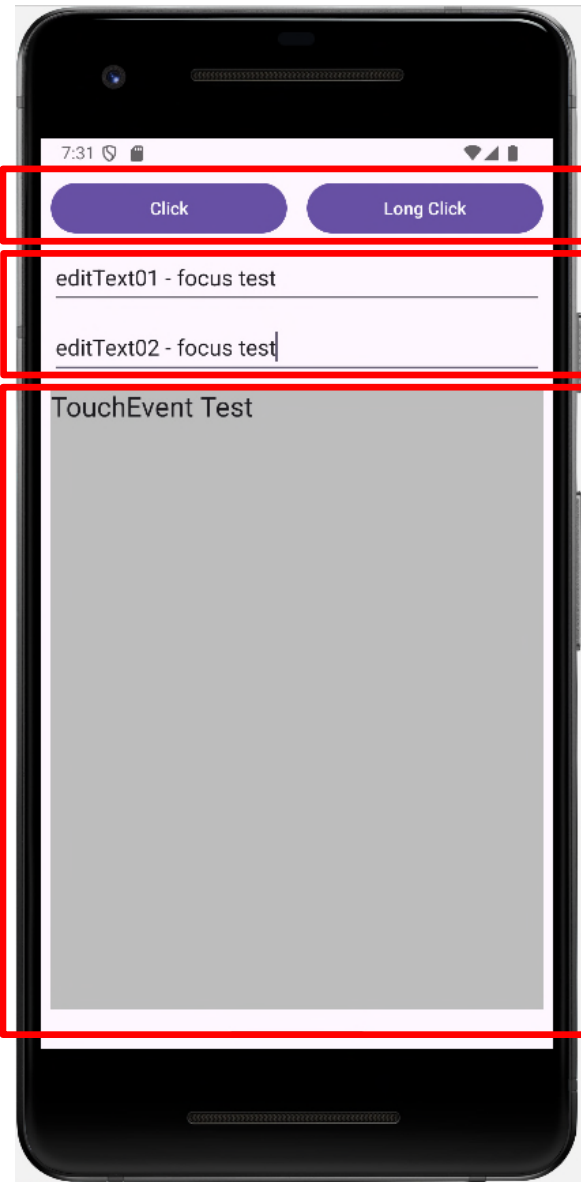
○ 특정 위치에서 키 이벤트 처리



□ Touch



□ 레이아웃 구성



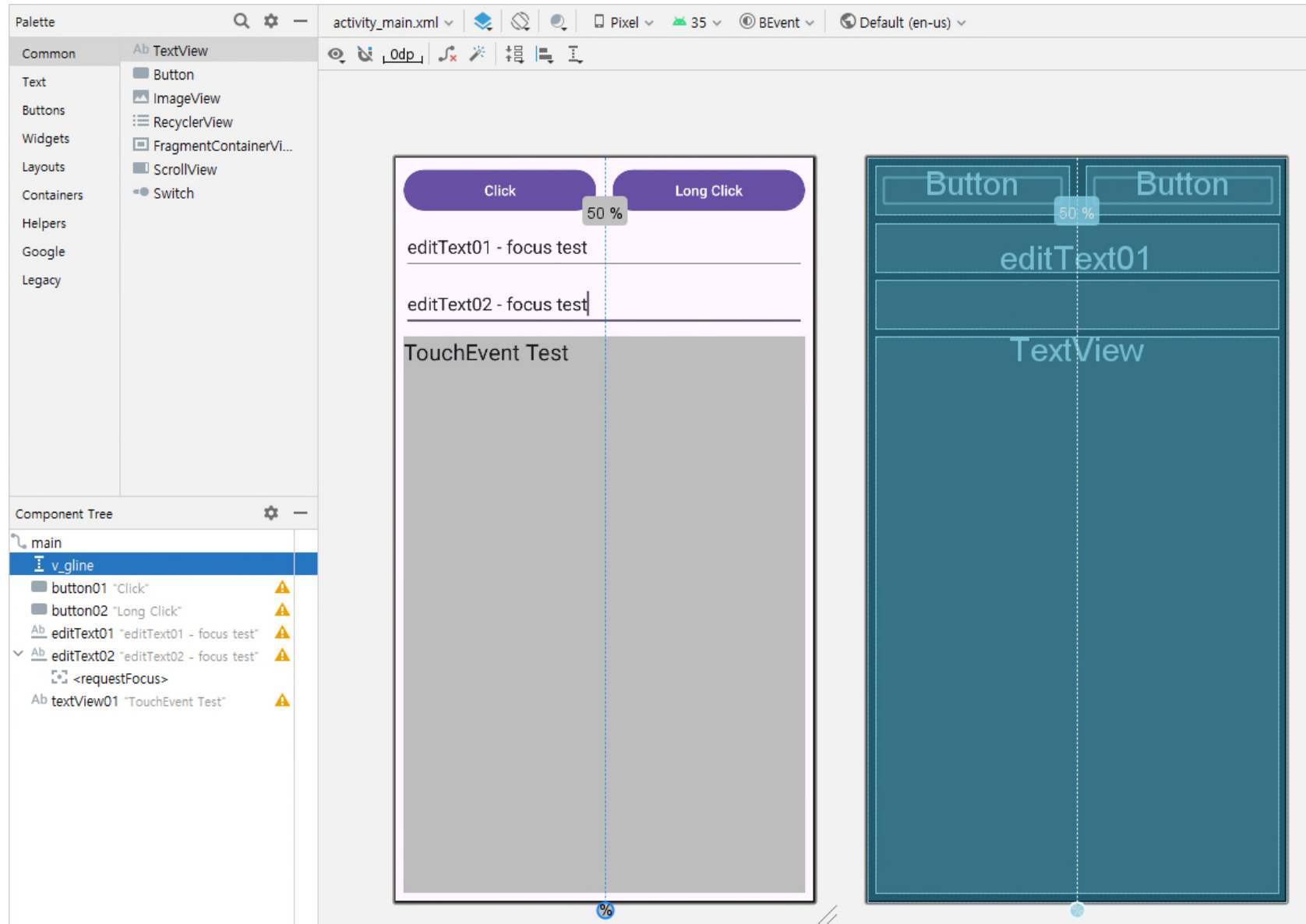
클릭과 롱클릭 처리를 위한 버튼

포커스 변경 테스트를 위한
에디트텍스트 2개
상단은 포커스 등록
하단은 특정 위치에서의 키 테스트 등록

터치 이벤트 확인용 텍스트뷰,
터치는 텍스트뷰
안에서만 발생

□ 레이아웃 작성

○ ConstraintLayout 구성 과정에서 필요한 비율 지정 방법 확인



□ 레이아웃 작성

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
```

기준 위치 설정, 비율 지정 등이
필요할 때 Guideline 사용

```
<androidx.constraintlayout.widget.Guideline
    android:id="@+id/v_gline"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintGuide_percent="0.50" />
```

비율로 지정하는 경우 : layout_constraintGuide_percent

시작점을 기준으로 거리를 지정하는 경우 : layout_constraintGuide_begin

끝점을 기준으로 거리를 지정하는 경우 : layout_constraintGuide_end

□ 레이아웃 작성

```
} <Button
    android:id="@+id/button01"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:text="Click"
    app:layout_constraintEnd_toStartOf="@+id/v_gline"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

}

} <Button
    android:id="@+id/button02"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:text="Long Click"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="@+id/v_gline"
    app:layout_constraintTop_toTopOf="parent" />

}
```


□ 레이아웃 작성

```
<EditText
    android:id="@+id/editText01"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:ems="10"
    android:inputType="text"
    android:minHeight="48dp"
    android:text="editText01 - focus test"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button01" />
```

```
<EditText
    android:id="@+id/editText02"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:ems="10"
    android:inputType="text"
    android:minHeight="48dp"
    android:text="editText02 - focus test"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editText01">
```

```
<requestFocus />
```

```
</EditText>
```

레이아웃 상의 초기 포커스
변경이 필요한 경우,
포커스를 가지는 구성요소 안에
<requestFocus /> 추가

□ 레이아웃 작성

```
<TextView
    android:id="@+id/textView01"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:background="#FFBDBDBD"
    android:text="TouchEvent Test"
    android:textAppearance="@style/TextAppearance.AppCompat.Large"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editText02" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

□ 코드 분석/설명

```
package com.practice.ex.bevent;

import android.os.Bundle;
import android.view.KeyEvent;
import android.view.MotionEvent;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {
```

아래와 같이 필요한 리스너를 모두 기술

```
implements View.OnClickListener,
            View.OnLongClickListener,
            View.OnFocusChangeListener,
            View.OnKeyListener,
            View.OnTouchListener {
```

```
Button btn01, btn02; 2 usages
EditText edit01, edit02; 4 usages
TextView text01; 7 usages
```

□ 코드 분석/설명

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable( $this$enableEdgeToEdge: this);
    setContentView(R.layout.activity_main);
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {...});

    btn01 = (Button) findViewById(R.id.button01);
    btn01.setOnClickListener(this);

    btn02 = (Button) findViewById(R.id.button02);
    btn02.setOnLongClickListener(this);

    edit01 = (EditText) findViewById(R.id.editText01);
    edit01.setOnFocusChangeListener(this);

    edit02 = (EditText) findViewById(R.id.editText02);
    edit02.setOnKeyListener(this);

    text01 = (TextView) findViewById(R.id.textView01);
    text01.setOnTouchListener(this);
}
```

레이아웃과 연결한 후에
각각의 해당되는 리스너에 등록

클릭리스너에 등록

롱클릭리스너에 등록

포커스리스너에 등록

키리스너에 등록

터치리스너에 등록

□ 코드 분석/설명

@Override

public void onClick(View v) {

Toast.makeText(context: this, text: "onClick", Toast.LENGTH_LONG).show();

}

Click이 발생했을 때

@Override

public boolean onLongClick(View v) {

Toast.makeText(context: this, text: "onLongClick", Toast.LENGTH_LONG).show();

return false;

}

Long Click이 발생했을 때

토스트 알림 출력

출력할 메시지 작성, 메시지 길이

- Toast.LENGTH_SHORT
- Toast.LENGTH_LONG

□ 코드 분석/설명

```
@Override
public void onFocusChange(View v, boolean hasFocus) { 포커스 변경이 발생했을 때 호출
    String s = "";
    if(hasFocus) { 포커스를 얻은 경우
        ((EditText) v).setText("");
    } else {
        s = edit01.getText().toString();
        edit01.setText(s + " + focus change");
    }
}
```

포커스를 잃은 경우

키 이벤트 발생시

```
@Override
public boolean onKeyDown(View v, int keyCode, KeyEvent event) {
    if(keyCode == KeyEvent.KEYCODE_BACK) {
        Toast.makeText(context: this, text: "KEYCODE_BACK", Toast.LENGTH_LONG).show();
    }
    if(keyCode == KeyEvent.KEYCODE_VOLUME_UP) {
        text01.setText("Volume Up");
    }
    if(keyCode == KeyEvent.KEYCODE_VOLUME_DOWN) {
        text01.setText("Volume Down");
    }
    return false;
}
```

Back 키 및 볼륨 키 등록 테스트
* Back키의 경우 기능 관련 추후 변동사항이 있음

□ 코드 분석/설명

```
@Override
public boolean onTouch(View v, MotionEvent event) {
    if(event.getAction() == MotionEvent.ACTION_DOWN) {
        text01.setText("ACTION_DOWN");
        return true;
    }
    if(event.getAction() == MotionEvent.ACTION_MOVE) {
        text01.setText("ACTION_MOVE");
        return true;
    }
    if(event.getAction() == MotionEvent.ACTION_UP) {
        text01.setText("ACTION_UP");
        return true;
    }
    return false;
}
```

터치 이벤트 발생시

ACTION_DOWN
- 터치 할 때(누를 때)

ACTION_MOVE
- 터치한 상태로 움직일 때

ACTION_UP
- 떴는 경우

□ 안드로이드 기능/실습 - 10

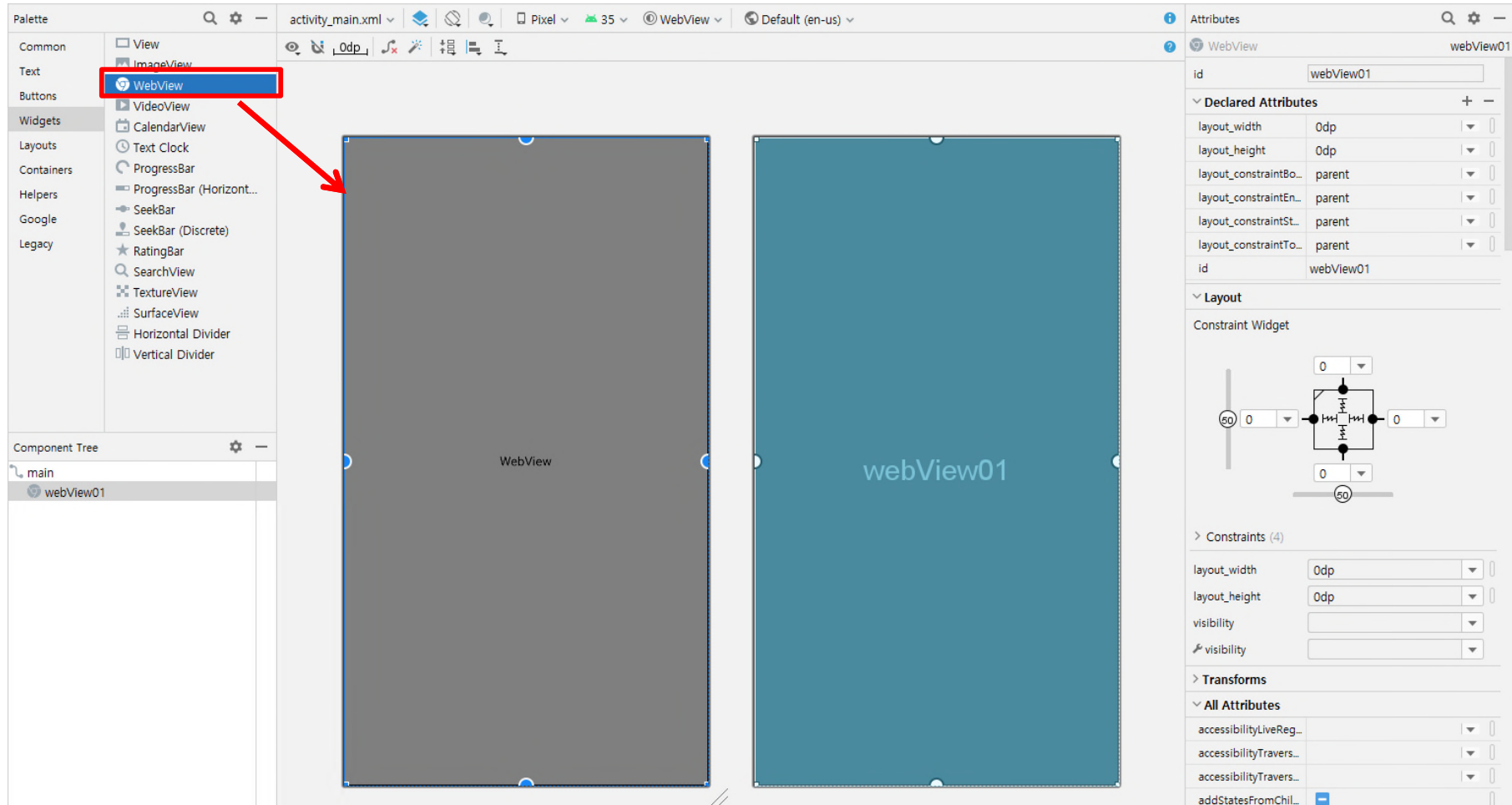
네트워크 - WEBVIEW

□ Network - WebView

- 안드로이드에서는 네트워킹과 관련하여 java.net, WebKit과 같은 네트워크 기술을 제공
- 웹에서 가져온 자료를 화면에 보여주기 위해서 WebView라는 위젯을 이용
- WebView
 - WebKit 엔진 이용
 - 웹페이지를 화면에 디스플레이하는 뷰
 - 사용자가 자신의 웹브라우저를 만드는 기반
 - 액티비티 안의 온라인 콘텐츠 표시 가능
 - 줌인, 줌아웃, 텍스트 탐색 등의 기능 지원



기능을 단순 확인하기 위함이라 webView 1개로 전체 화면을 구성



□ 레이아웃

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <WebView
        android:id="@+id/webView01"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

□ AndroidManifest.xml

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

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

인터넷 접근 권한 허용 추가

```
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="WebView"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:usesCleartextTraffic="true"
        android:theme="@style/Theme.WebView"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

ERR_CLEARTEXT_NOT_PERMITTED

해결 관련 추가 코드

-> API28 (Pie) 이상을 대상으로 하는 경우,
http://로 시작되는 주소를 호출시
에러가 발생

□ 코드 분석

```
package com.practice.ex.webview;

import android.os.Bundle;
import android.view.KeyEvent;
import android.webkit.WebView;
import android.webkit.WebViewClient;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {

    WebView web01; 9 usages

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {...});
    }
}
```

□ 코드 분석

```
web01 = (WebView) findViewById(R.id.webView01); 레이아웃 객체 연결
web01.setWebViewClient(new WebViewClient()); 웹뷰의 기본 기능을
// 필요에 따라 설정 추가 변경하기 위해
// 캐시 제거가 필요한 경우 - 구현 위치는 상황에 따라 조절 WebViewClient 클래스 사용
web01.getSettings().setJavaScriptEnabled(true); 자바스크립트 허용 여부
web01.getSettings().setBuiltInZoomControls(true); 확대 기능 사용 여부
web01.clearCache( includeDiskFiles: true); 캐시 및 히스토리 제거
// 필요에 따라 기타 추가적 설정 진행
web01.clearHistory();
web01.loadUrl("https://www.seoil.ac.kr/"); 주어진 해당 URL을
// } 이용하여 웹페이지 표시
```

□ 코드 분석

```
@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if((keyCode == KeyEvent.KEYCODE_BACK) && web01.canGoBack()) {
        web01.goBack();
        return true;
    }
    return super.onKeyDown(keyCode, event);
}
```

Back키를 눌렀을 때 이전 페이지로 돌아갈 수 있으면 이전 페이지로 돌아감

웹뷰를 보이기 위한 코드로 웹뷰를 구현하는 과정을 보여주기 위한 매우 간략히 구현한 경우이므로 좀 더 **세밀한 기능 구현을 위해서는 상당량의 추가 구현이 필요함**