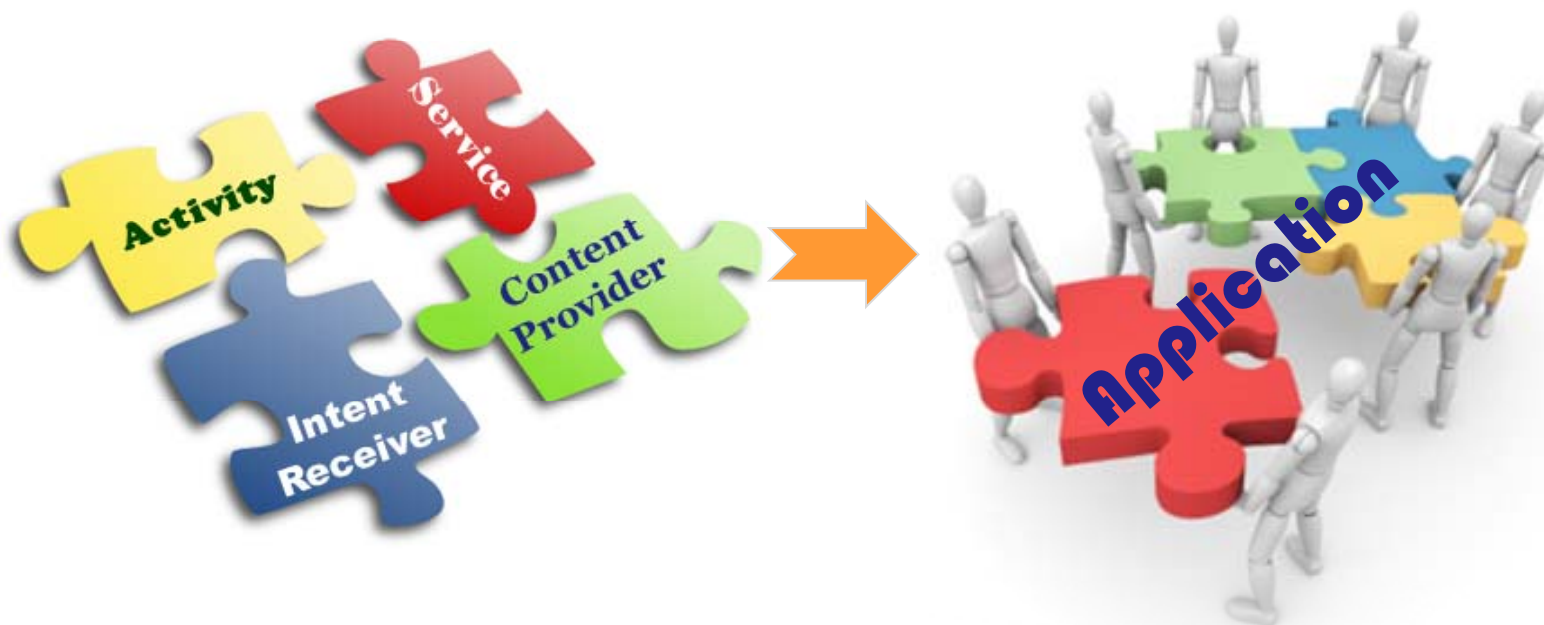




Android UI 구현





학습 목표

교육 목표

- ❖ Android application 구성 요소
- ❖ Activity
- ❖ Layout
- ❖ UI 설계
- ❖ Linear Layout 구현



Navigation voice



Traffic view



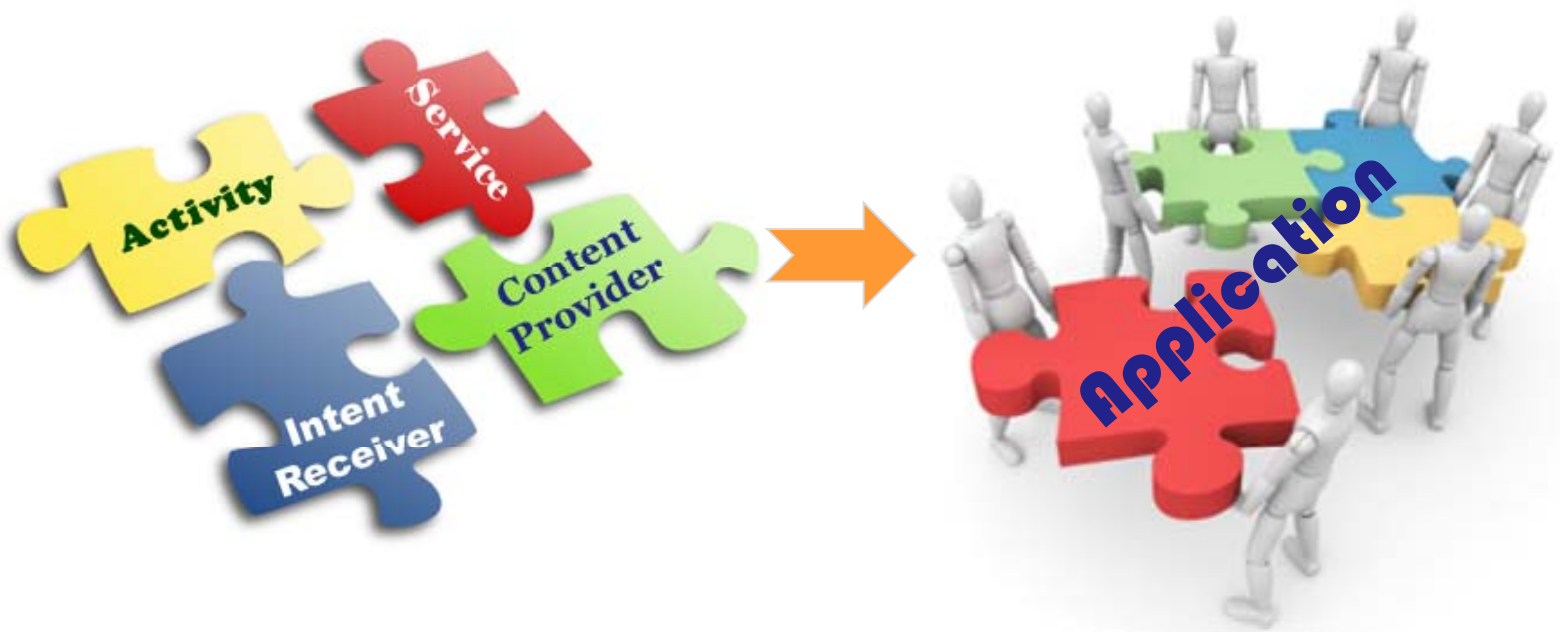
Street View





Android application 구성 요소 (1)

■ Application 구성 요소



AndroidManifest.xml





Android application 구성 요소 (2)

구성 요소	기능
Activity	<ul style="list-style-type: none">• Application의 프리젠테이션 계층• UI component를 화면에 표시• Windows application의 Form과 동일 기능
Intent Receiver	<ul style="list-style-type: none">• 메시지 전달 framework• Event handler 기능
Service	<ul style="list-style-type: none">• UI와 관계없이 오랫동안 존재하면서 실행되는 코드• background로 실행(예: MediaPlayer)
Content Provider	<ul style="list-style-type: none">• Application data 공유• DB(SQLite) 연동





Android application 구성 요소 (3)

❖ AndroidManifest.xml

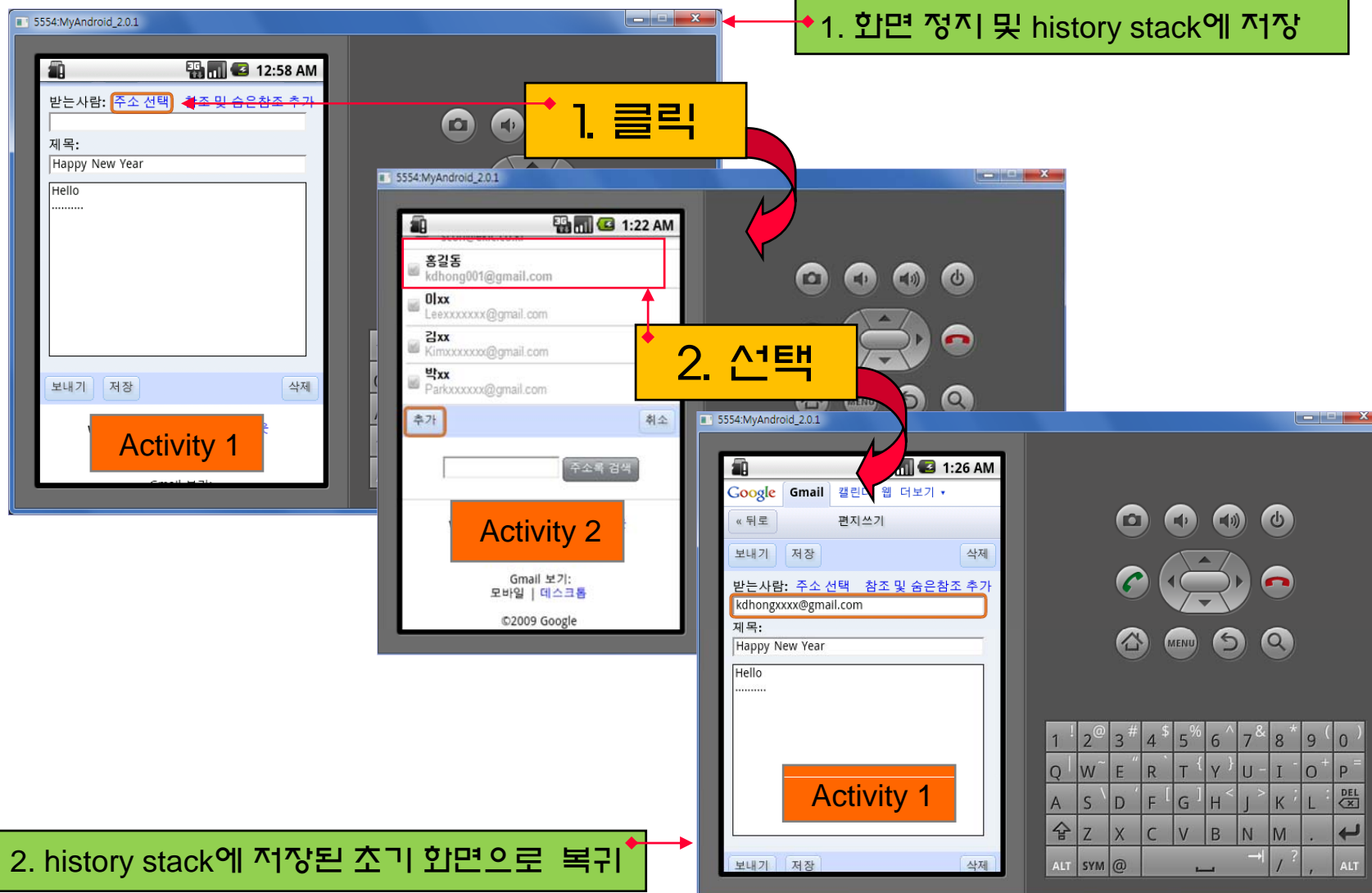
```
*Android_EditText Manifest
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.inhatec.android_EditText"
    android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".Android_EditText"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-sdk android:minSdkVersion="6" />
</manifest>
```





Activity (1)

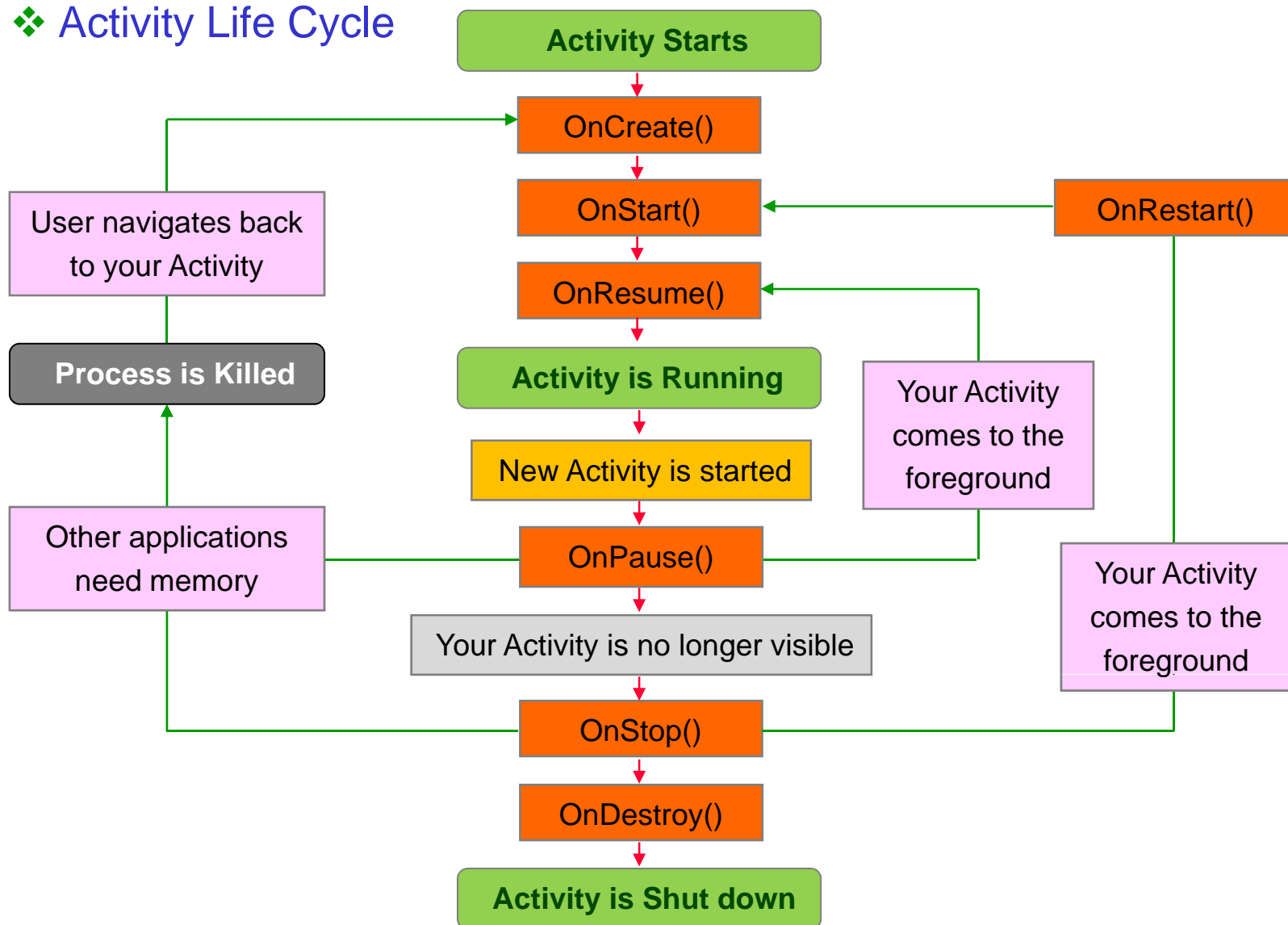
Activity





Activity (2)

❖ Activity Life Cycle





Activity (3)

❖ Activity Method

구성 요소	기능
OnCreate ()	<ul style="list-style-type: none">• Activity가 생성될 때 처음 호출• 리소스 초기화• findViewById(int) 메소드 호출 : Widget 검색• setContentView(int) 메소드 호출 : Widget 화면 출력
OnStart ()	<ul style="list-style-type: none">• Activity 화면 출력시 호출
OnResume ()	<ul style="list-style-type: none">• Activity stack의 Top에 위치• Activity 화면 출력 및 입력 처리시 호출
OnPause ()	<ul style="list-style-type: none">• Activity 1 -> Activity 2 : Activity 1의 내용 저장• 데이터 저장, animation 중지, CPU 점유 작업 중단 등 수행
OnStop ()	<ul style="list-style-type: none">• 더 이상 Activity가 stack의 Top에 위치하지 않으므로 Activity 출력 안됨
OnDestroy ()	<ul style="list-style-type: none">• 시스템에서 Activity 제거





Activity (4)

Welcome.java

```
Welcome.java X
package com.inhatc.welcome;

import android.app.Activity;

public class Welcome extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```





Activity (5)

Process Life Cycle

Foreground activity

- 최상위 화면의 activity로 가장 중요한 프로세스
- 메모리 부족 시 마지막으로 제거



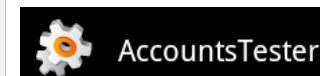
Visible activity

- 화면에 보이는 activity로 foreground activity가 아닌 activity



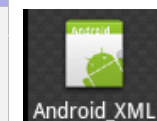
Background activity

- 화면에 보이지 않고 stop 상태의 activity
- 메모리 부족 시 foreground, visible activity보다 먼저 제거
- onSaveInstanceState(Bundle savedInstanceState) method 호출
– savedInstanceState : 중요 데이터 저장
- onCreate() method에서 savedInstanceState parameter를 이용하여 Background activity 제거 전의 상태로 복귀 가능



Empty Process

- Service, Broadcast Receiver 클래스와 같이 activity, application component도 없는 프로세스
- 메모리 부족 시 즉시 제거





Layout (1)

■ Layout

❖ 기능

◆ 화면 구성

❖ 분류

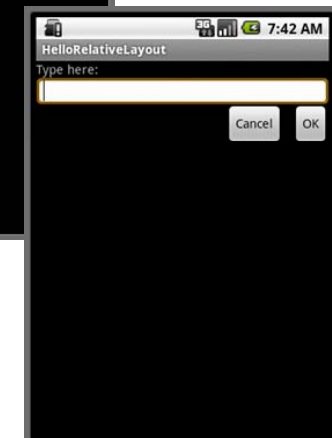
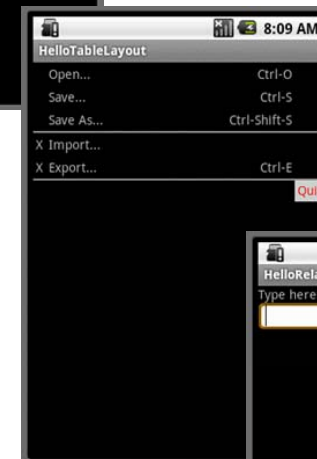
◆ FrameLayout

◆ LinearLayout

◆ TableLayout

◆ AbsoluteLayout

◆ RelativeLayout





Layout (2)

■ View 클래스

❖ 기능

- ◆ 직사각형 형태의 layout과 각종 정보 저장
- ◆ 화면 크기 조절, Layout 구성, draw, focus 변환
- ◆ Scrolling, key 처리
- ◆ Widget의 base class

❖ Widget

- ◆ Text, EditText, Button, RadioButton, Checkbox, ScrollView 등과 같이 화면에 컴포넌트처럼 동작하는 것
- ◆ Widget을 이용하면 UI를 빠르게 구현





Layout (3)

❖ View 크기 지정 단위

- ◆ px (pixels)
- ◆ dip (device independent pixels)
- ◆ sp (scaled pixels – best for text size)
- ◆ pt (points)
- ◆ in (inches)
- ◆ mm (millimeters)





Layout (4)

■ ViewGroup

❖ View 집합체

Class	설 명
FrameLayout	<ul style="list-style-type: none">• 단일 객체를 표현하기 위한 View frame 기능• 2개 이상의 객체를 표현할 때는 마지막으로 그린 객체 화면 출력
LinearLayout	<ul style="list-style-type: none">• 수평/수직 으로 구성 요소 배치• 창의 길이가 화면 길이를 초과하면 scrollbar 자동 생성
TableLayout	<ul style="list-style-type: none">• Table• 행 : 가장 큰 구성 요소 크기에 맞게 설정• 셀 테두리 : hidden
AbsoluteLayout	<ul style="list-style-type: none">• 객체의 위치를 좌표값(x, y)으로 지정
RelativeLayout	<ul style="list-style-type: none">• 하나의 구성 요소를 기준으로 Up/Down, Left/Right 로 상대적 위치 지정
Gallery	<ul style="list-style-type: none">• image를 수평으로 scroll하면서 보여주는 클래스
GridView	<ul style="list-style-type: none">• 눈금 구조를 보여주는 클래스
ListView	<ul style="list-style-type: none">• scrolling 되는 column list를 보여줌
ScrollView	<ul style="list-style-type: none">• 수직으로 구성 요소의 열을 scrolling



Layout (5)

❖ ViewGroup

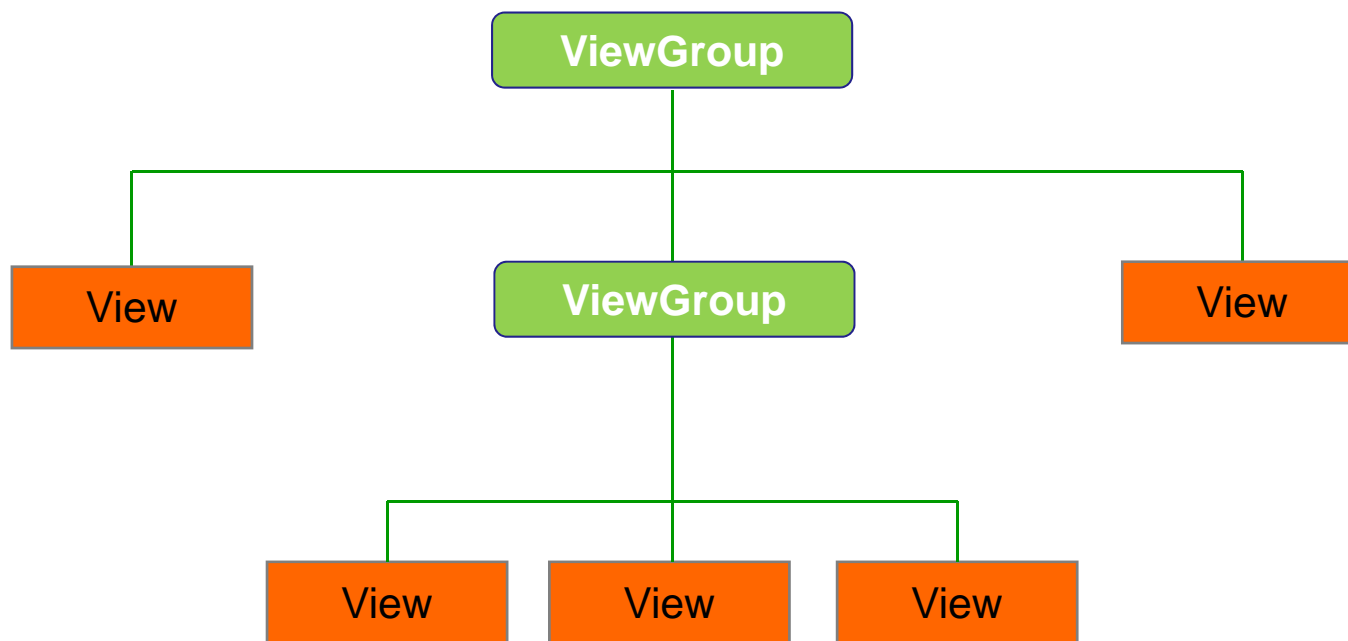
Class	설 명
Spinner	<ul style="list-style-type: none">• 1 line textbox에 list item을 하나씩 출력
SurfaceView	<ul style="list-style-type: none">• 그리기 전용 표면에 직접 접근 가능• 점과 line을 직접 draw 하는 low-level coding에 사용
TabHost	<ul style="list-style-type: none">• click에 반응하는 Tab 목록 제공• Tab 클릭시 application 변경
ViewFlipper	<ul style="list-style-type: none">• 한번에 1 개의 item 출력• slider show와 같이 주기적으로 item 변경 출력
ViewSwitcher	<ul style="list-style-type: none">• ViewFlipper와 동일 기능





Layout (6)

■ Tree 구조의 User Interface



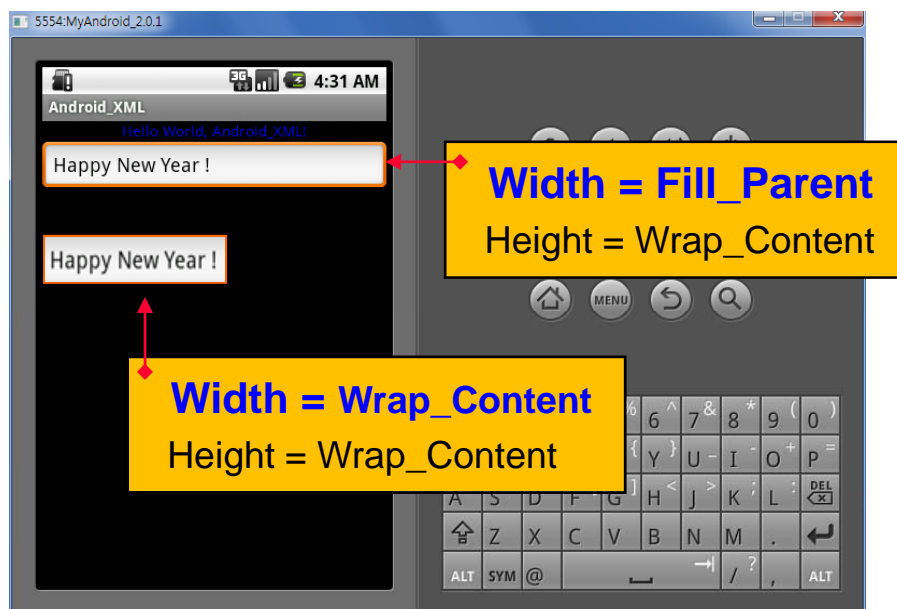


Layout (7)

■ LayoutParams 클래스

❖ 기능

- ◆ View 객체를 그리기 위한 정보를 부모 객체에 전송
- ◆ 각 객체의 Width, Height 설정
 - WRAP_CONTENT : 필요한 최소 크기
 - FILLPARENT : 부모 객체와 여백을 제외한 모든 공간



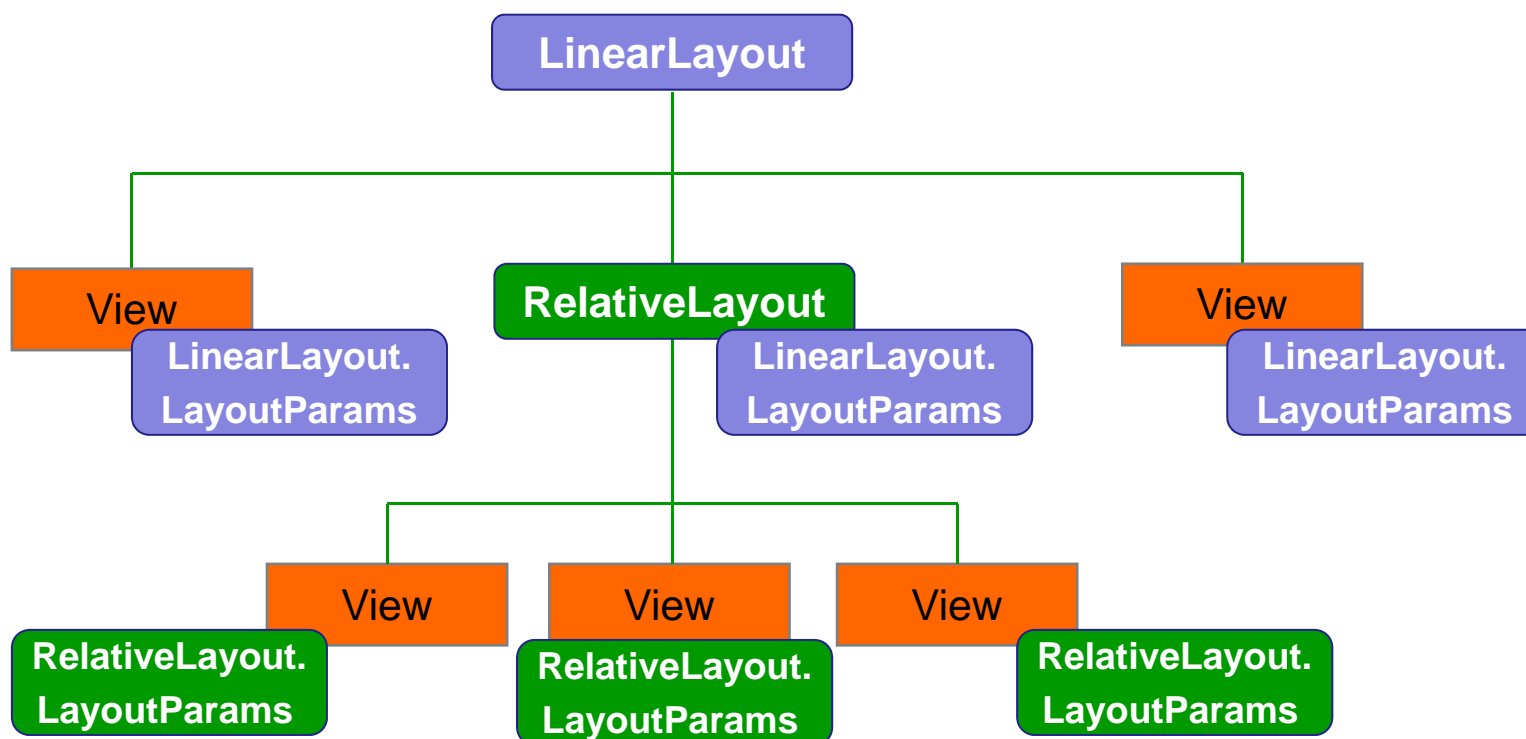


Layout (8)

LayoutParams subclass 설정

❖ 자식 객체

◆ 부모 객체의 속성에 맞는 LayoutParams 지정





UI 설계

■ UI 설계

❖ 방법 1

◆ XML 프로그래밍

- \res\layout\main.xml

```
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello"
/>
```

<EditText

Element

```
    android:text="@+id/EditText01"
    android:id="@+id/EditText01"
    android:layout_height="wrap_content"
    android:layout_width="fill_parent"
    android:editable="true">
```

</EditText>

Attribute

❖ 방법 2

◆ Java 프로그래밍

- \src*.java

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    EditText objET = (EditText)findViewById(R.id.EditText01);
    objET.setText("Merry Christmas !");
    objET.setTextColor(Color.BLUE);
    objET.setGravity(0x01);
    String strData = objET.getText().toString();
    objET.setText(strData + " Happy New Year !");
}
```

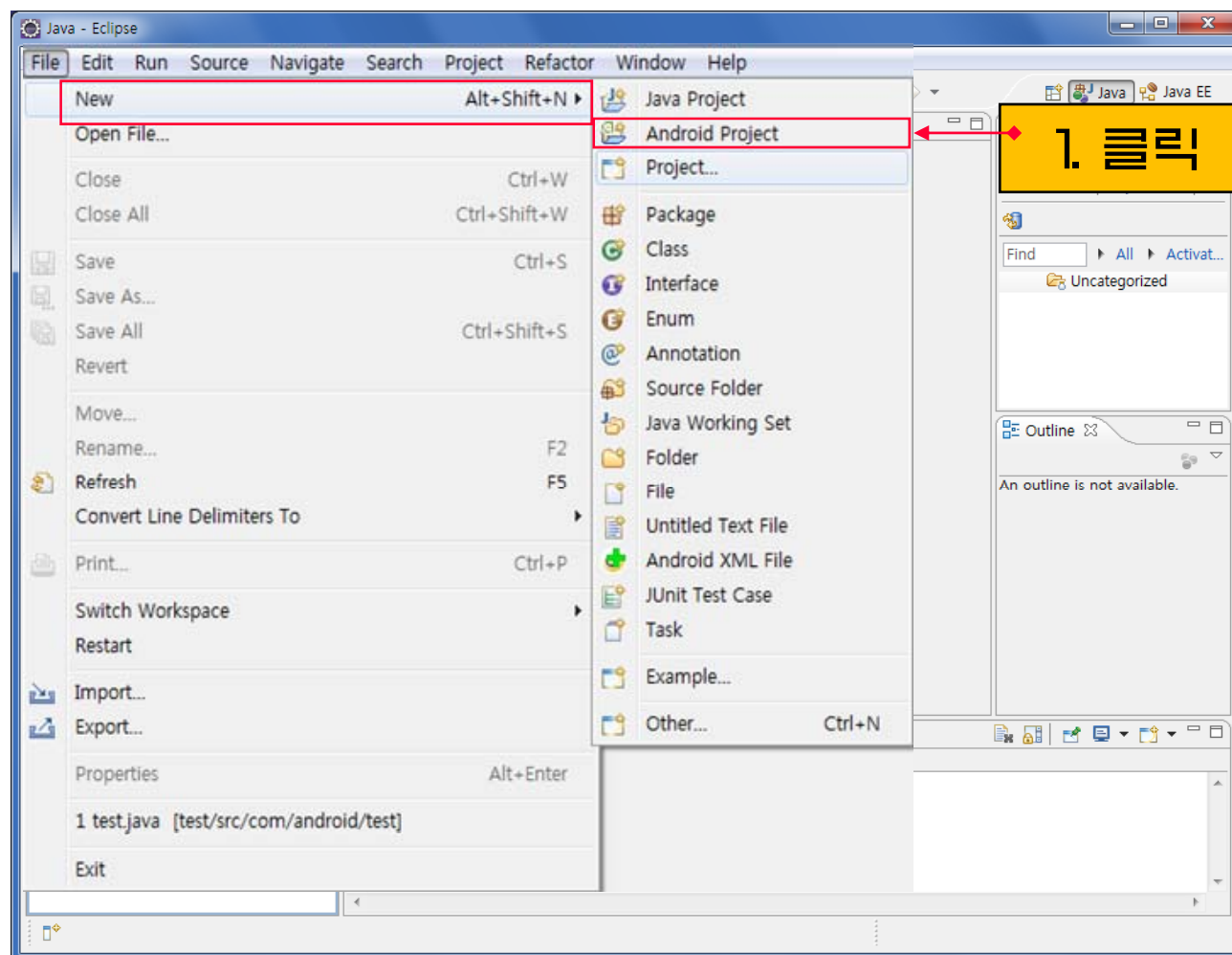




Linear Layout 구현 (1)

■ Android 프로젝트 생성

❖ 프로젝트 명 : Android_LinearLayout





Linear Layout 구현 (2)

New Android Project

Creates a new Android Project resource.

Project name:

Contents

☒ Create new project in workspace

☐ Create project from existing source

☒ Use default location

Location:

☐ Create project from existing sample

Samples:

Build Target

Target Name	Vendor	Platform	API ...
<input type="checkbox"/> Android 1.1	Android Open Source Project	1.1	2
<input type="checkbox"/> Android 1.5	Android Open Source Project	1.5	3
<input type="checkbox"/> Android 1.6	Android Open Source Project	1.6	4
<input type="checkbox"/> Android 2.0	Android Open Source Project	2.0	5
<input checked="" type="checkbox"/> Android 2.0.1	Android Open Source Project	2.0.1	6
<input type="checkbox"/> Google APIs	Google Inc.	1.5	3
<input type="checkbox"/> Google APIs	Google Inc.	1.6	4

Standard Android platform 2.0.1

Properties

Application name:

Package name:

☒ Create Activity:

Min SDK Version:

2. Android_LinearLayout 입력

3. 클릭

4. Android_LinearLayout 입력

5. com.inhatec.Android_LinearLayout 입력

6. Android_LinearLayout 입력

7. 클릭





Linear Layout 구현 (3)

■ UI 설계

❖ LinearLayout 01

◆ TextView01

◆ TextView02

◆ TextView03

❖ LinearLayout 02

◆ TextView04

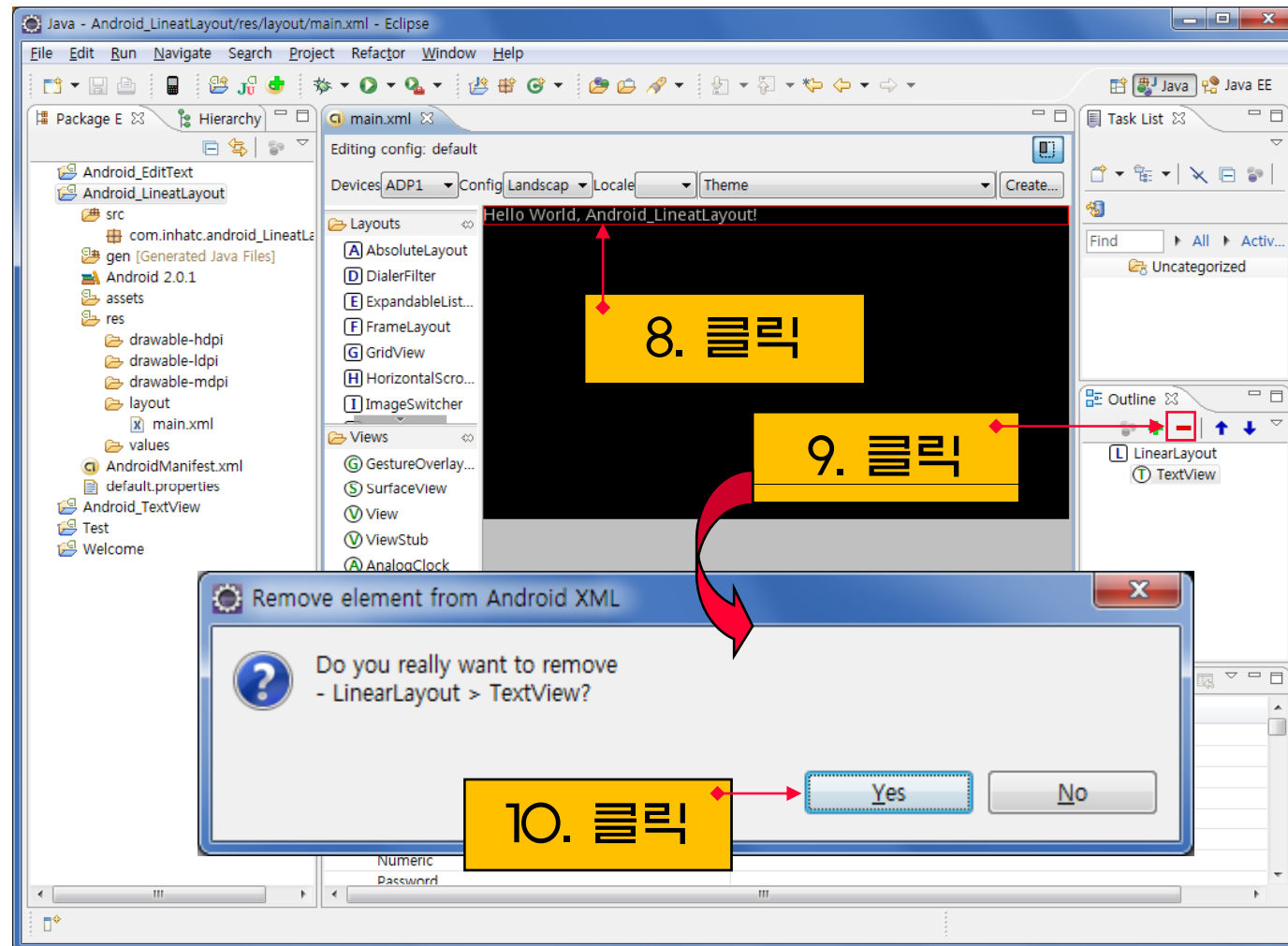




Linear Layout 구현 (4)

Linear Layouts

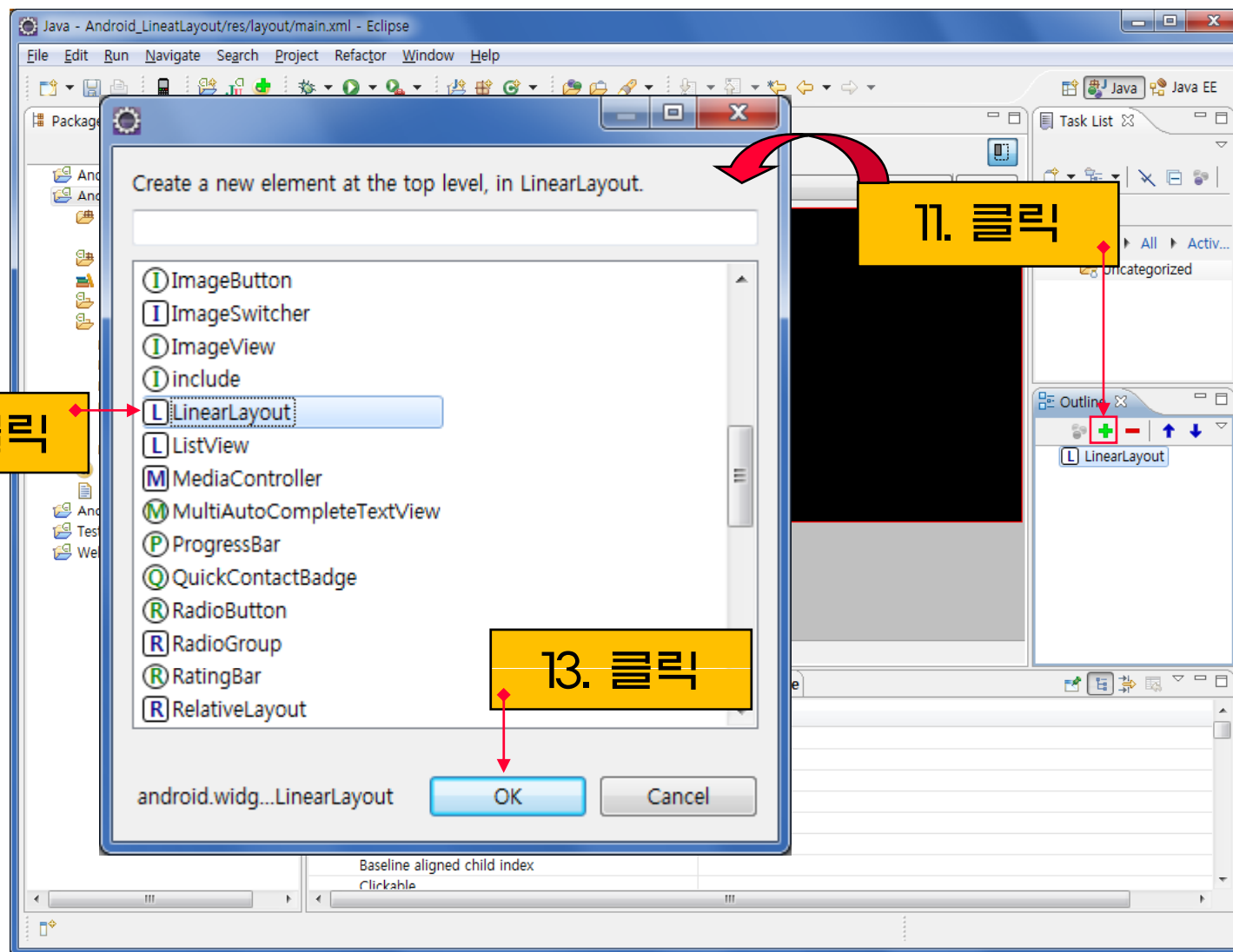
❖ TextView 삭제





Linear Layout 구현 (5)

❖ LinearLayout 추가





Linear Layout 구현 (6)

❖ LinearLayout 01 속성 지정

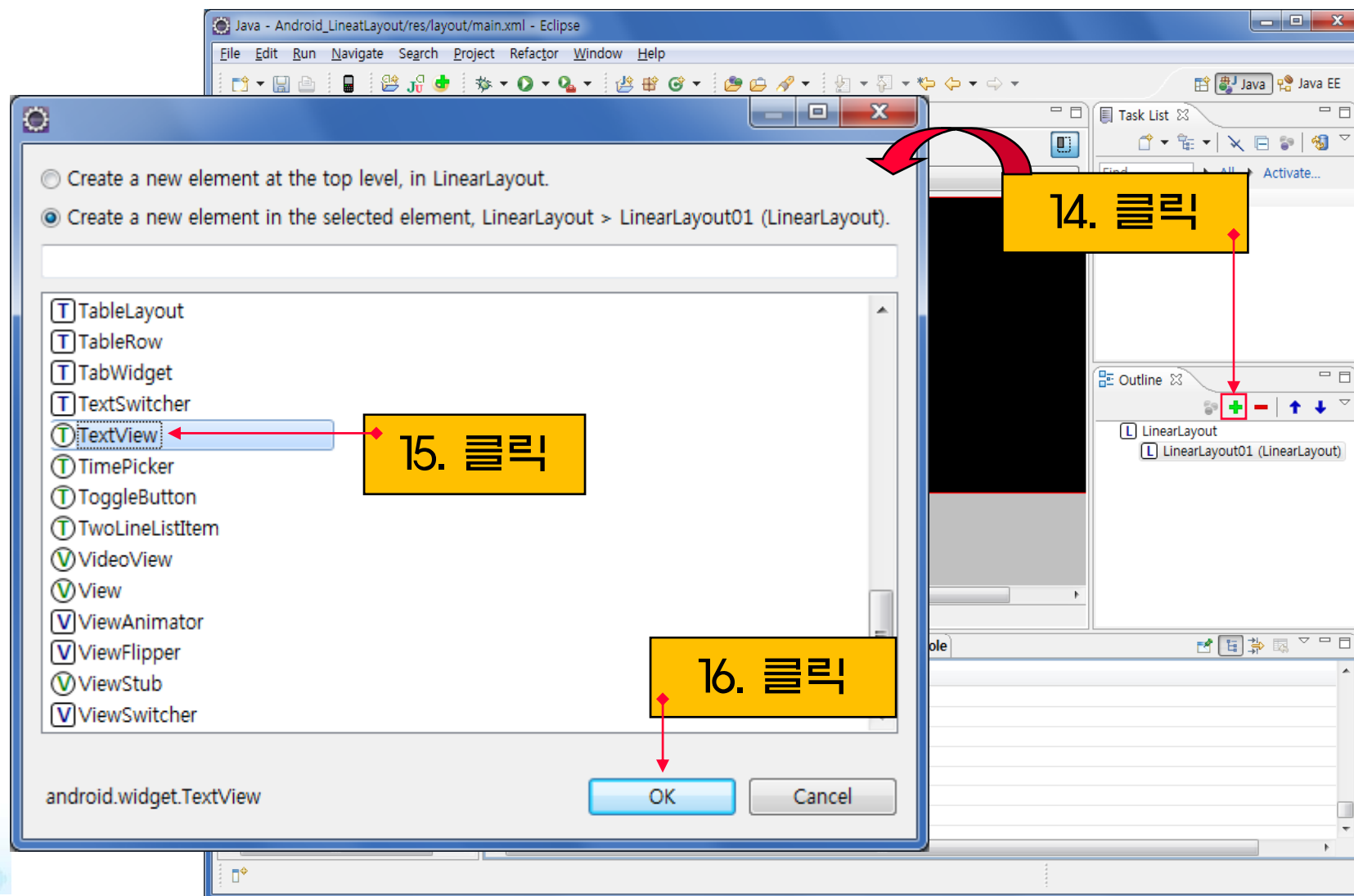
컨트롤	속성 지정
LinearLayout01	<ul style="list-style-type: none">• id="@+id/LinearLayout01"• layout_width="fill_parent"• layout_height="fill_parent"• layout_weight="1"• orientation="horizontal"





Linear Layout 구현 (7)

❖ TextView 추가





Linear Layout 구현 (8)

❖ TextView 01 ~ 03 속성 지정

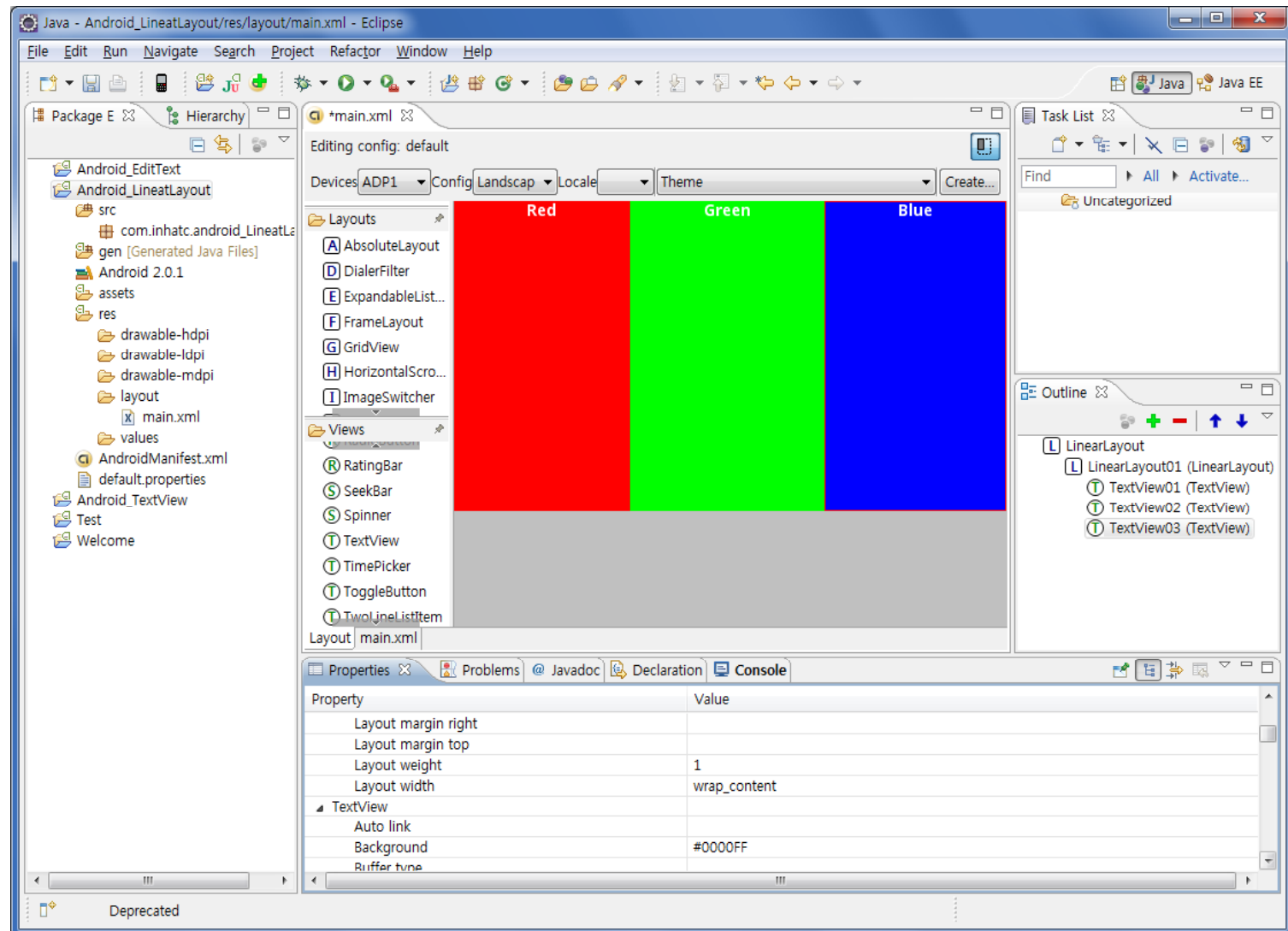
컨트롤	속성 지정
TextView01	<ul style="list-style-type: none">• id="@+id/TextView01"• layout_width="wrap_content"• layout_height="fill_parent"• text="Red"• gravity="center_horizontal"• layout_weight="1"• background="#FF0000"• textColor="#FFFFFF"• textStyle="bold"
TextView02	<ul style="list-style-type: none">• id="@+id/TextView02"• layout_width="wrap_content"• layout_height="fill_parent"• text="Green"• gravity="center_horizontal"• layout_weight="1"• background="#00FF00"• textColor="#FFFFFF"• textStyle="bold"
TextView03	<ul style="list-style-type: none">• id="@+id/TextView03"• layout_width="wrap_content"• layout_height="fill_parent"• text="Blue"• gravity="center_horizontal"• layout_weight="1"• background="#0000FF"• textColor="#FFFFFF"• textStyle="bold"





Linear Layout 구현 (9)

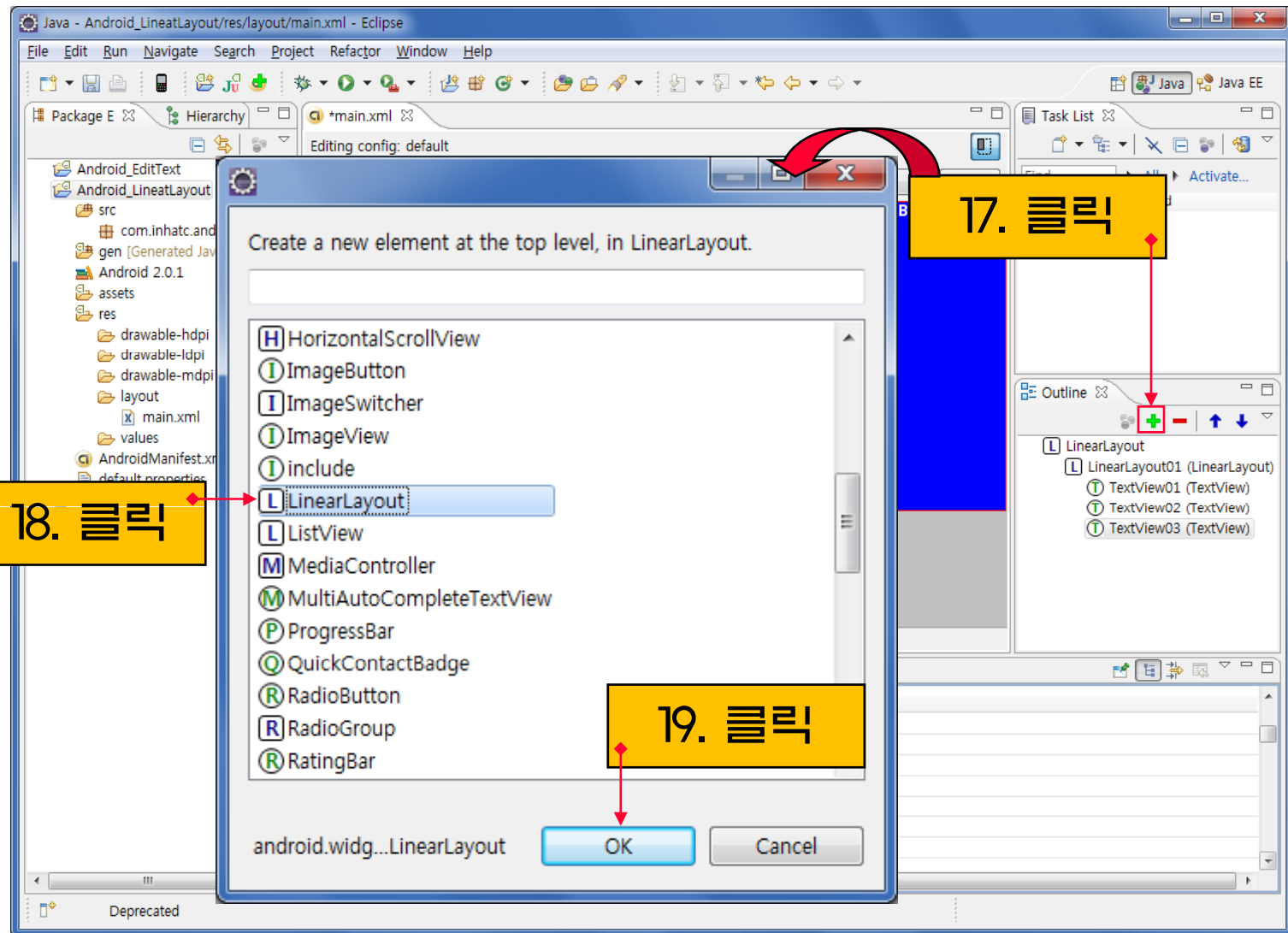
❖ TextView 속성 지정 결과





Linear Layout 구현 (10)

❖ LinearLayout 추가





Linear Layout 구현 (11)

❖ LinearLayout02 속성 지정

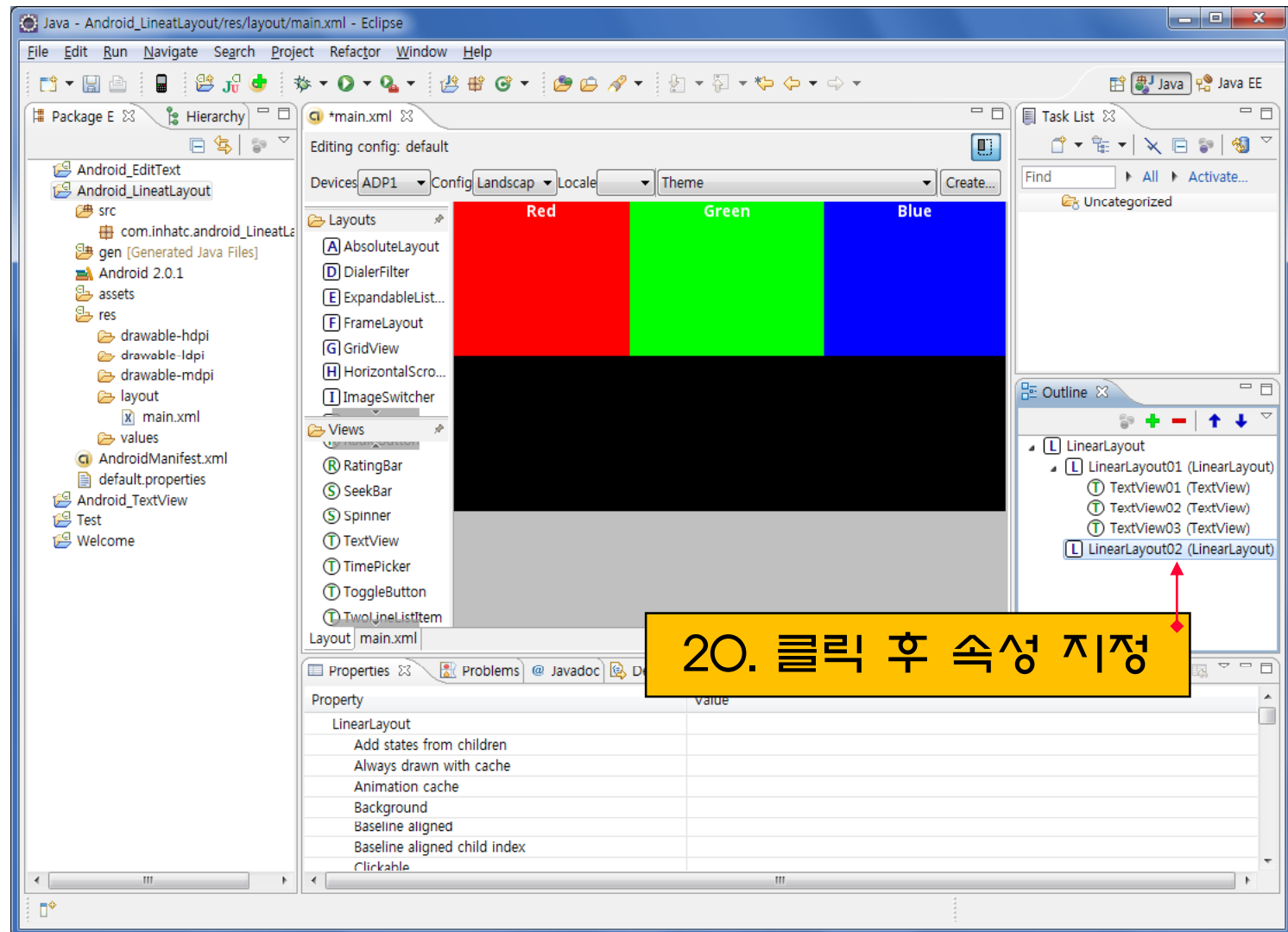
컨트롤	속성 지정
LinearLayout02	<ul style="list-style-type: none">• id="@+id/LinearLayout02"• layout_width="fill_parent"• layout_height="fill_parent"• layout_weight="1"• orientation="vertical"





Linear Layout 구현 (12)

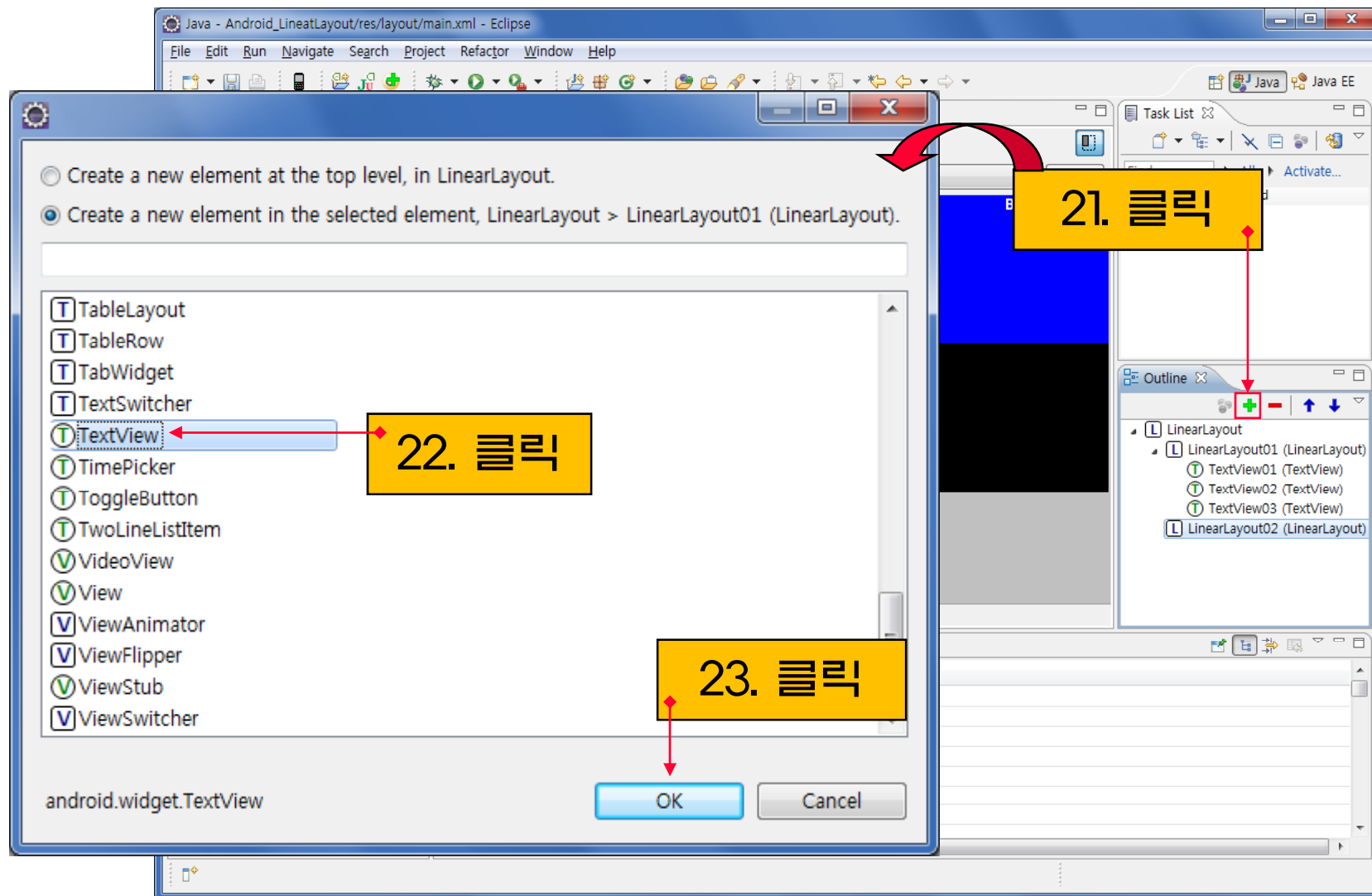
❖ LinearLayout02 속성 지정 결과





Linear Layout 구현 (13)

❖ TextView 추가





Linear Layout 구현 (14)

❖ TextView 04 속성 지정

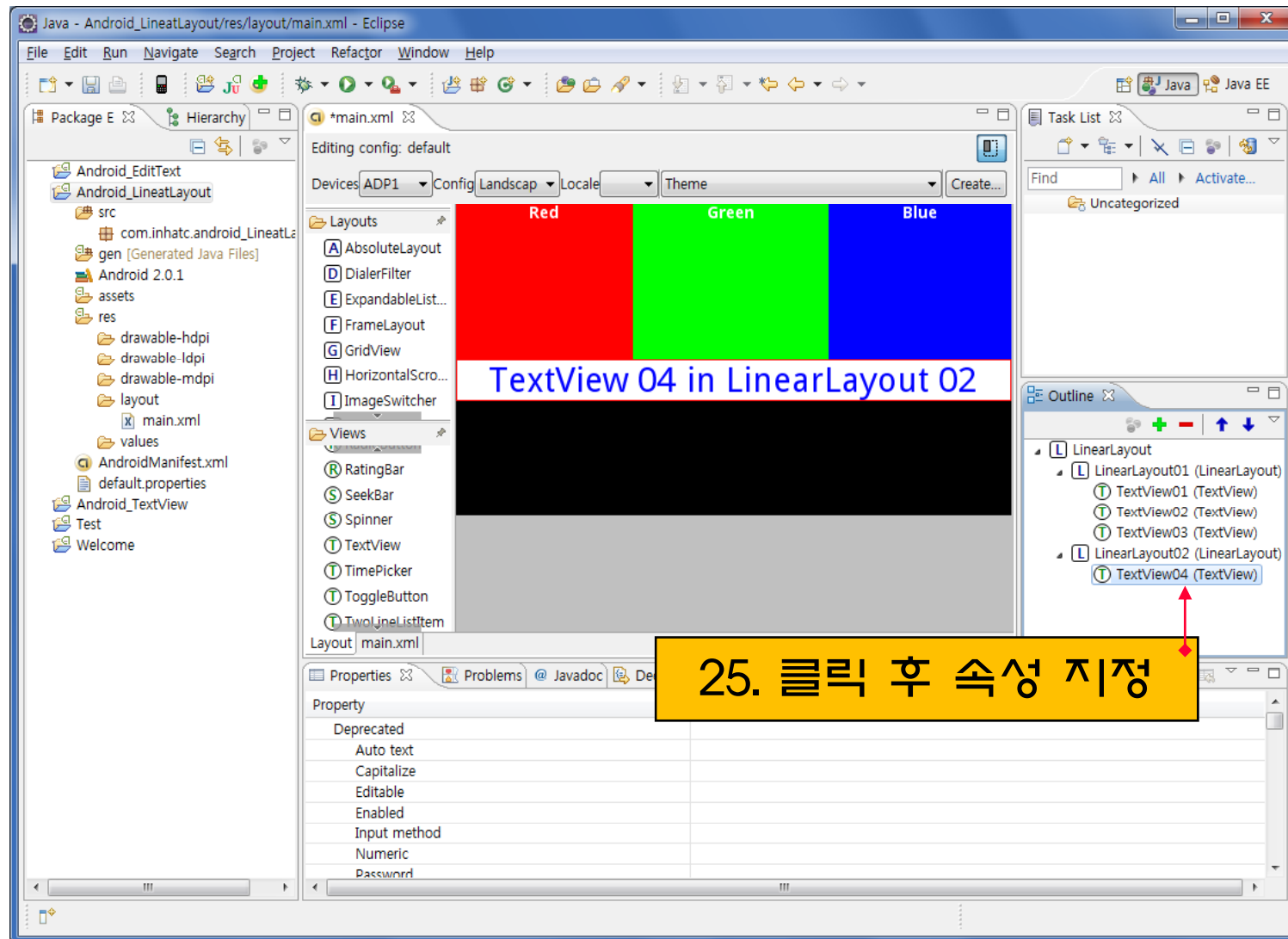
컨트롤	속성 지정
TextView04	<ul style="list-style-type: none">• id="@+id/TextView04"• layout_width="fill_parent"• layout_height="wrap_content"• typeface="normal"• text="TextView 04 in LinearLayout 02"• gravity="center_horizontal"• layout_weight="1"• background="#FFFFFF"• textSize="12pt"• textColor="#0000FF"• textStyle="bold"





Linear Layout 구현 (15)

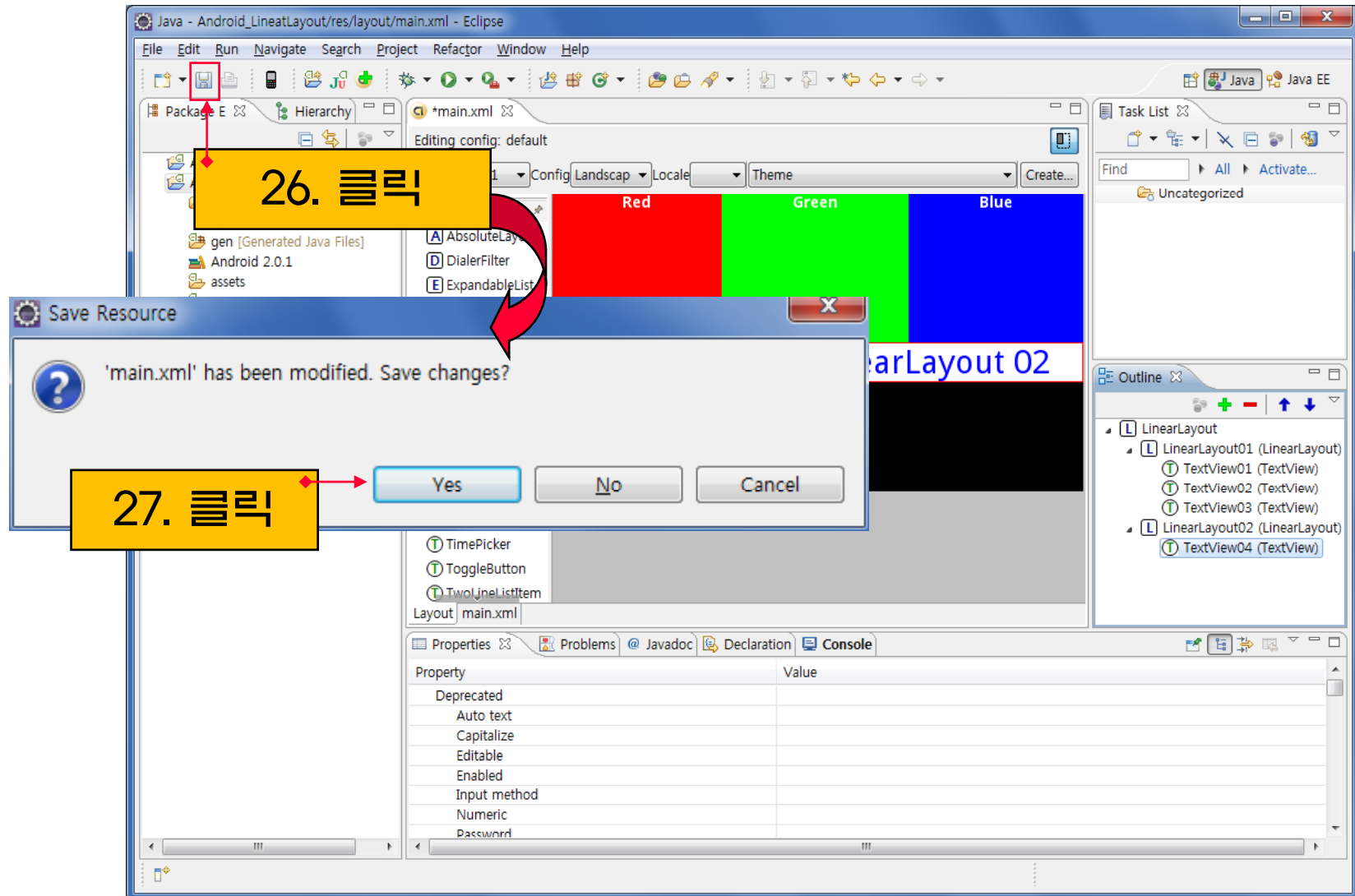
❖ TextView04 속성 지정 결과





Linear Layout 구현 (16)

❖ Main.xml 저장





Linear Layout 구현 (17)

❖ Main.xml

```
*main.xml x
    android:layout_weight="1"
    android:gravity="center_horizontal">
</TextView>
</LinearLayout>

<LinearLayout
    android:id="@+id/LinearLayout02"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_weight="1"
    android:layout_height="fill_parent">
    <TextView
        android:id="@+id/TextView04"
        android:layout_height="wrap_content"
        android:layout_width="fill_parent"
        android:typeface="normal"
        android:text="TextView 04 in LinearLayout 02"
        android:background="#FFFFFF"
        android:textSize="12pt"
        android:textColor="#0000FF"
        android:gravity="center_horizontal">
    </TextView>
</LinearLayout>
</LinearLayout>
```

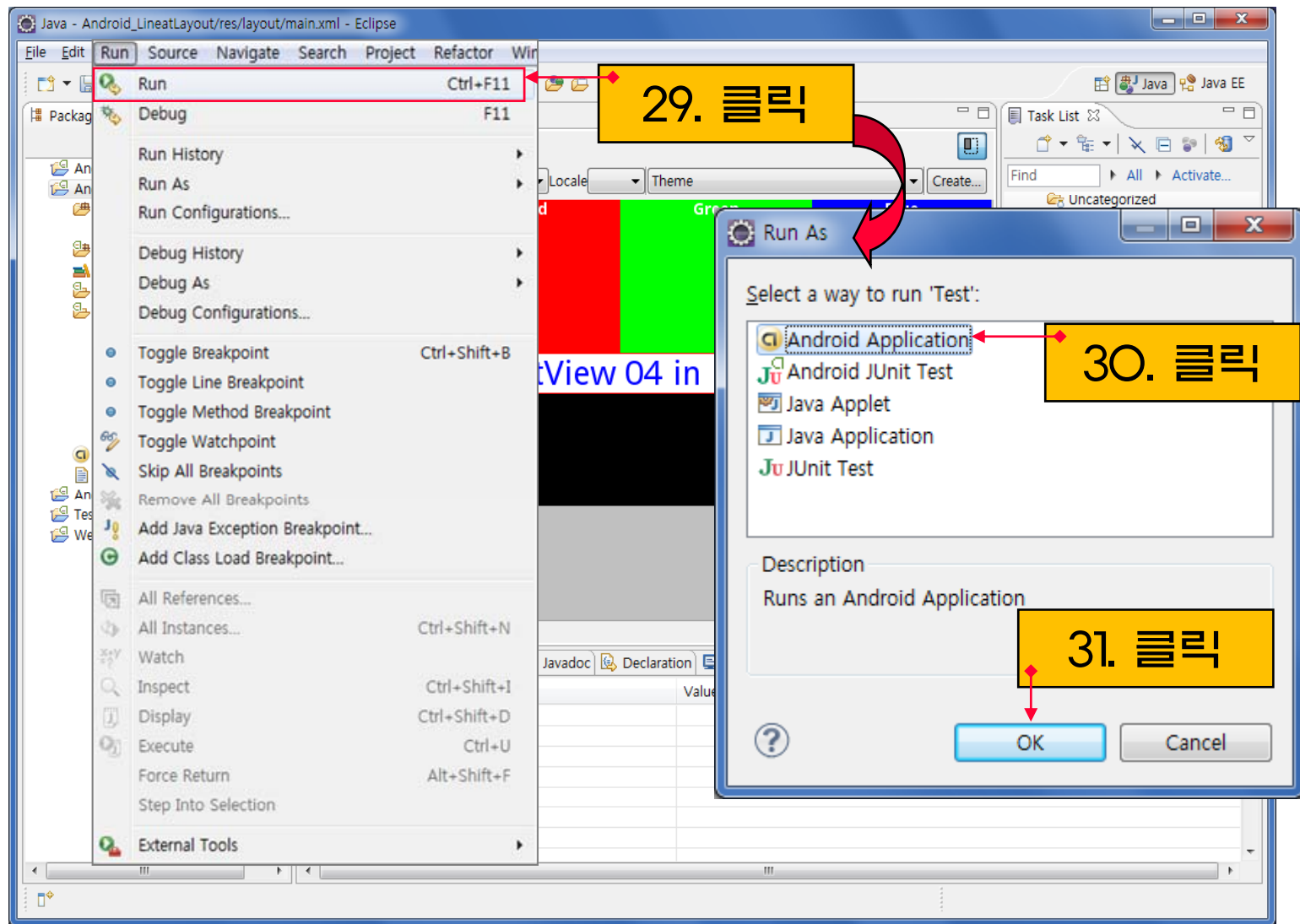
28. XML code 추가 확인





Linear Layout 구현 (18)

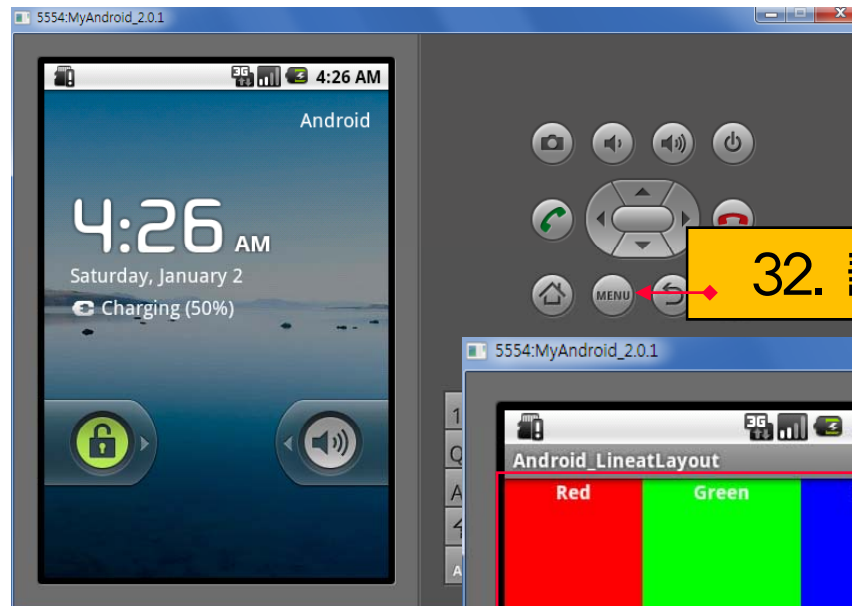
❖ Android 프로젝트 실행





Linear Layout 구현 (19)

실행 결과



32. 클릭



33. 결과 확인





Linear Layout 구현 (20)

❖ Home에서 Android 애플리케이션 실행하기





실습 1 : XML로 Linear Layout 구현 (1)

■ Android_LinearLayout 프로젝트 (실습 시간 : 20분)

❖ main.xml 파일을 프로그래밍하여 아래 그림과 같이 구현하시오.

◆ 단계 1 : LinearLayout01의 Blue TextView 오른쪽에 Grey TextView 추가

◆ 단계 2 : LinearLayout02의 TextView04 아래에 TextView06 추가

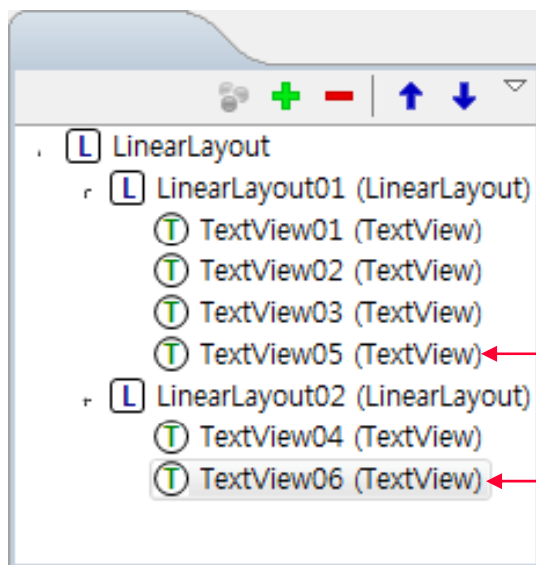




실습 1 : XML로 Linear Layout 구현 (2)

■ Solution

❖ Main.xml



```
<TextView
    android:id="@+id/TextView05"
    android:layout_width="wrap_content"
    android:text="Grey"
    android:background="#AAAAAA"
    android:textColor="#FFFFFF"
    android:textStyle="bold"
    android:layout_height="fill_parent"
    android:layout_weight="1"
    android:gravity="center_horizontal">
</TextView>
```

```
<TextView
    android:id="@+id/TextView06"
    android:layout_height="wrap_content"
    android:layout_width="fill_parent"
    android:typeface="normal"
    android:text="TextView 06 in LinearLayout 02"
    android:background="#FFFFFF"
    android:textSize="10pt"
    android:textColor="#FF0000"
    android:gravity="center_horizontal">
</TextView>
```





실습 1 : XML로 Linear Layout 구현 (3)

```
*main.xml
<TextView
    android:gravity="center_horizontal">
</TextView>
<TextView
    android:id="@+id/TextView05"
    android:layout_width="wrap_content"
    android:text="Grey"
    android:background="#AAAAAA"
    android:textColor="#FFFFFF"
    android:textStyle="bold"
    android:layout_height="fill_parent"
    android:layout_weight="1"
    android:gravity="center_horizontal">
</TextView>
</LinearLayout>

<LinearLayout
    android:id="@+id/LinearLayout02"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView
        android:id="@+id/TextView04"
        android:layout_height="wrap_content"
        android:layout_width="fill_parent"
        android:layout_weight="1"
        android:text="TextView 04 in LinearLayout 02"
        android:background="#FFFFFF"
        android:textSize="12pt"
        android:textColor="#0000FF"
        android:gravity="center_horizontal">
    </TextView>
    <TextView
        android:id="@+id/TextView06"
        android:layout_height="wrap_content"
        android:layout_width="fill_parent"
        android:layout_weight="1"
        android:text="TextView 06 in LinearLayout 02"
        android:background="#FFFFFF"
        android:textSize="10pt"
        android:textColor="#FF0000"
        android:gravity="center_horizontal">
    </TextView>
</LinearLayout>
</LinearLayout>
```

TextView05 XML Code

TextView06 XML Code





학습 요약

- Android application 구성 요소
- Activity
- Layout
- UI 설계
- Linear Layout 구현



open handset alliance